MODÉLISATION ET LOGIQUE PROPOSITIONNELLE CLASSIQUE

Yakoub Salhi

UNIVERSITÉ D'ARTOIS

Modélisation et logique propositionnelle classique

Habilitation à diriger des recherches

(Spécialité Informatique)

Université d'Artois
par

Yakoub Salhi

27 novembre 2019

Composition du jury :

Rapporteurs :	Bruno Crémilleux	Professeur à l'Université de Caen Normandie
	Frédéric Saubion	Professeur à l'Université d'Angers
	Torsten Schaub	Professeur à l'Université de Potsdam
Examinateurs :	Salima Benbernou	Professeur à l'Université Paris Descartes
	Souhila Kaci	Professeur à l'Université de Montpellier
	Jean-Marc Petit	Professeur à l'Institut National des Sciences Appliquées de Lyon
	Lakhdar Sais	Professeur à l'Université d'Artois
Directeur :	Jean-François Condotta	Professeur à l'Université d'Artois

CENTRE DE RECHERCHE EN INFORMATIQUE DE LENS – CNRS UMR 8188
Université d'Artois, rue Jean Souvraz, S.P. 18 F-62307, Lens Cedex France
Secrétariat : Tél. : +33 (0)3 21 79 17 23 – Fax : +33 (0)3 21 79 17 70
http://www.cril.univ-artois.fr

Table des matières

Première partie

Synthèse des travaux de recherche

Chapitre 1

Introduction

1.1 Préambule

Mes travaux de recherche se situent dans le cadre de l'intelligence artificielle. Ils concernent essentiellement l'utilisation des logiques formelles, ainsi que d'autres formalismes symboliques qui s'y apparentent, pour répondre à différentes problématiques, et cela, autour de deux principaux thèmes : la fouille de données par des approches déclaratives et la représentation des connaissances. Mes travaux s'inscrivent ainsi de manière transversale sur les deux axes du Centre de Recherche en Informatique de Lens (CRIL), à savoir « Algorithmes pour l'inférence et contraintes » et « Représentation des connaissances et raisonnements ». En effet, depuis ma thèse de doctorat sur la théorie de la démonstration en logiques modales, mes contributions relèvent notamment de la modélisation en logique propositionnelle de différents problèmes de fouille de données, du raisonnement en présence de l'incohérence, de la théorie de l'argumentation et du raisonnement qualitatif.

J'ai contribué à l'émergence de la thématique de fouille de données au CRIL à l'occasion de mon recrutement en tant que post-doctorant en octobre 2011, dans le cadre du projet ANR DAG (*Approches déclaratives pour l'énumération de motifs intéressants*). Travailler au sein de cette thématique a constitué une réelle ouverture par rapport à ma thèse de doctorat. En outre, ma présence au CRIL a eu pour effet de susciter mon intérêt pour des problématiques se rapportant au domaine de la représentation des connaissances et raisonnements. J'ai pu dans ce domaine tirer grandement profit des compétences en lien avec les logiques formelles acquises durant la réalisation de ma thèse de doctorat.

Mes travaux sur les systèmes de preuve en logiques formelles se sont également poursuivis, mais à un degré moindre. Toutefois, j'ai initié dans ce contexte une collaboration avec la Régie Autonome des Transports Parisiens (RATP) dans le cadre de l'amélioration de son atelier de qualification logicielle. J'étais ainsi le responsable scientifique d'un projet industriel qui a permis le développement d'outils de preuve formelle utilisés dans le projet d'automatisation de lignes de la RATP, de même que dans des projets réalisés pour des clients externes.

Le contenu de ce mémoire est principalement dévolu à mes contributions relatives à

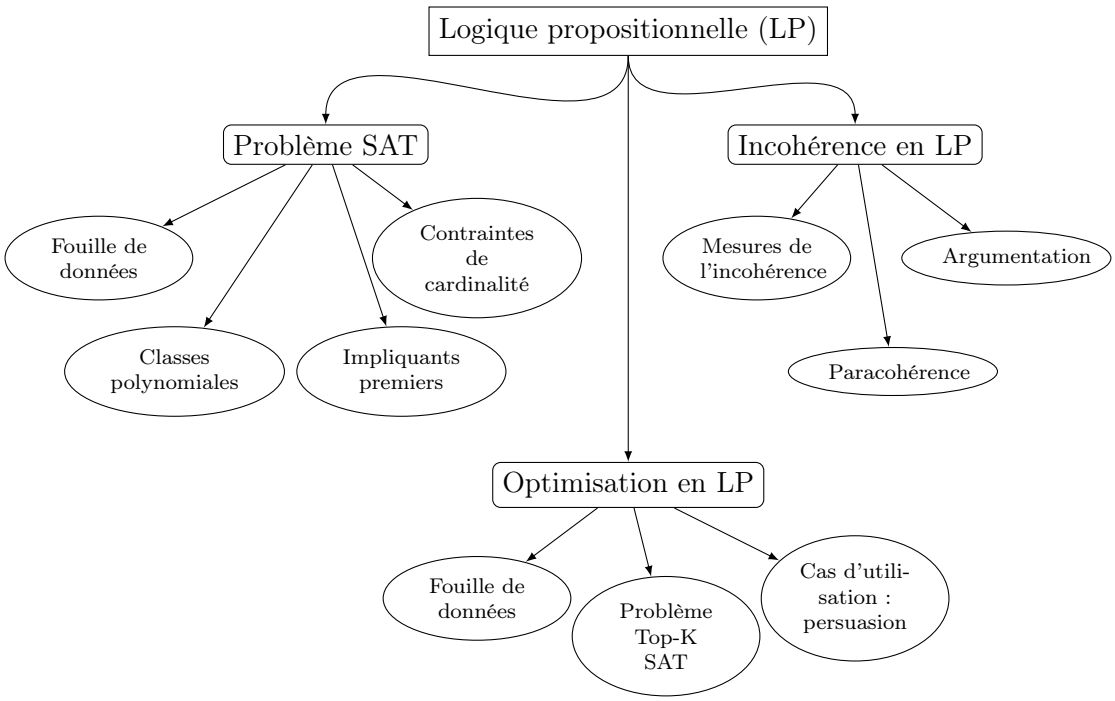


FIGURE 1.1 – Mes travaux relatifs à la logique propositionnelle

l'utilisation de la logique propositionnelle classique comme outil de modélisation, et ce, du fait de leur place prépondérante dans l'ensemble de mes travaux et également pour des raisons de cohésion. Néanmoins, afin de ne pas éluder totalement les apports de mes travaux non détaillés dans ce mémoire, en particulier ceux en relation avec le raisonnement qualitatif et la théorie de l'argumentation, je présente ces derniers de manière succincte dans cette introduction.

1.2 Modélisation en logique propositionnelle

La logique propositionnelle classique est centrale dans l'étude de nombreux problèmes en informatique en général, et en intelligence artificielle en particulier. Dans une grande partie de nos travaux, nous nous sommes focalisés sur l'utilisation de cette logique comme outil de modélisation en suivant trois principaux angles de vue. Le premier angle concerne l'emploi du problème de la cohérence en logique propositionnelle. Le deuxième se rapporte aux problèmes d'optimisation dérivés de la logique propositionnelle. Quant au troisième angle, il est lié à l'utilisation de cette logique en présence de l'incohérence.

Cohérence en logique propositionnelle. Le problème de la cohérence en logique propositionnelle, appelé SAT, possède plusieurs qualités importantes motivant son utilisation en tant qu'outil de modélisation. Il s'agit en effet d'un problème simple à appréhender, ce qui facilite entre autres la compréhension des modèles, mais il est aussi pourvu d'une force d'expressivité permettant à la fois le traitement de problèmes complexes et la définition de modèles compacts. En outre, les progrès considérables depuis plusieurs années

des outils de résolution modernes pour SAT constituent un argument supplémentaire en faveur de son utilisation dans des approches déclaratives.

Optimisation en logique propositionnelle. Motivées par des considérations pratiques, différentes variantes d'optimisation ayant comme base SAT ont fait leur apparition dans la littérature. Elles emploient des fonctions objectives pouvant être en rapport avec les valeurs de vérité associées aux variables dans un modèle, comme dans le problème MinCostSAT, ou avec les clauses falsifiées par une interprétation, comme dans le problème MaxSAT (voir [BHvW09] pour une vue générale). Les problèmes d'optimisation fondés sur SAT trouvent plusieurs applications dans divers domaines, tels que l'ordonnancement [DMW19], la bio-informatique [LM06] et l'analyse des programmes [SMV+07].

Incohérence en logique propositionnelle. Au nombre des cadres où la logique propositionnelle peut être d'une grande aide, il y a ceux nécessitant de raisonner en présence d'informations contradictoires, comme en particulier les croyances et les préférences d'un agent. Dans ce contexte, plusieurs approches ont été introduites afin de raisonner sous l'incohérence de manière pertinente, dont en particulier les mesures de l'incohérence [HK10], les relations de conséquence logique paracohérentes [TBMP13], la théorie de l'argumentation [BH08] et la révision des croyances [Gär92].

Nous décrivons dans ce qui suit nos contributions en utilisant comme ligne directrice les deux domaines dans lesquels nous avons appliqué la modélisation en logique propositionnelle, à savoir la fouille de données et la représentation des connaissances et raisonnements. La figure 1.1 représente de façon synthétique nos travaux dans le cadre de la logique propositionnelle suivant les trois angles de vue précédents.

1.3 Fouille de données

Cadre général

La fouille de données englobe de nombreuses techniques dont l'objectif est l'extraction de connaissances pertinentes ayant des formes variées, et cela, à partir de grands volumes de données (voir par exemple [Agg15]). Elle s'intéresse principalement à deux types d'approches : celles prédictives, comme la classification, et celles explicatives, comme la génération de règles d'association. Les techniques de fouille possèdent un champ d'application très large du fait de leur grande utilité dans l'analyse de données, surtout à une époque où nous assistons à une évolution importante et grandissante des capacités de collecte d'informations. En effet, l'émergence continuelle de nouvelles applications, comme le développement remarquable des techniques utilisées, font de la fouille de données un domaine de premier plan en informatique.

De par la multiplicité des sphères d'application, les techniques de fouille sont utilisées sur plusieurs types de données : les transactions, les séquences, les graphes, les textes, etc. Ces différents types, ainsi que les diverses tâches en fouille de données associées, ont créé le besoin de cadres génériques. C'est dans ce contexte que s'inscrit l'approche déclarative proposée initialement dans [DGN08], où les auteurs montrent que la programmation par contraintes est un outil générique, approprié à plusieurs égards, pour effectuer différentes tâches liées à l'extraction de motifs ensemblistes. Par la suite, un grand nombre de contributions adhérant à cette approche ont vu le jour. En particu-

lier, parmi les abondants travaux utilisant la modélisation en CSP (Constraint Satisfaction Problem) de problèmes en fouille de données, nous pouvons mentionner ceux dans [KBC10, GND11, Gun15, UBLC15, UBC+17, BLM18]. Notre étude, quant à elle, se focalise sur l'utilisation dans ce cadre du problème de la cohérence en logique propositionnelle SAT.

Contributions

Dans le cadre de nos travaux, nous avons proposé des formulations en SAT pour la réalisation de plusieurs tâches en fouille de données. Ainsi dans [CJSS12, JSS13b], nous avons proposé des formulations pour différents problèmes relatifs à l'énumération de motifs séquentiels. Dans ce contexte, de nouveaux problèmes ont en particulier été introduits dans [JSS13b] généralisant ceux liés à la fouille de motifs séquentiels avec joker, tout en montrant que nos formulations s'adaptent de manière très simple à ces problèmes. De plus, nous avons abordé dans [JSS15] la fouille de motifs ensemblistes par la proposition d'un nouveau cadre pour l'utilisation de SAT reposant sur une approche de décomposition. Toujours par rapport aux motifs ensemblistes, nos travaux dans [OJS+15] ont consisté à proposer une approche d'extraction parallèle fondée sur SAT.

Dans [JSS13c, JSS17], nous avons d'abord introduit un problème correspondant à une variante de SAT, appelé *Top-K SAT*. Ce dernier est défini comme le calcul d'un ensemble de modèles d'une formule propositionnelle considérés comme les meilleurs selon une relation de préférence donnée. Il a ensuite été montré que ce problème s'applique dans le cas de tâches de fouille de motifs ensemblistes et de motifs séquentiels via toujours la modélisation en SAT. Par rapport à l'utilisation des relations de préférence entre motifs, nous avons également proposé un nouveau problème en fouille de motifs ensemblistes dans [JKSS16], où l'extraction de ces derniers est réalisée en tenant compte de différentes formes de préférence. Il est à noter que nos contributions en lien avec l'intégration explicite de préférences sur les motifs en fouille de données reposent largement sur l'utilisation de l'optimisation en logique propositionnelle.

Dans [BJSY16, BJSS17c, BJSS17b], nous avons proposé des formulations en SAT pour l'extraction de différents types de règles d'association. Nous nous sommes en effet intéressés à la fouille de règles d'association suivant la définition répandue qui utilise les notions de support et de confiance, à l'instar de restrictions à des représentations condensées, comme les règles fermées et celles minimales non redondantes, et des variantes comme les règles indirectes. Un fait notable issu de notre étude expérimentale relative aux règles d'association qu'il convient de signaler, est que notre approche permet dans le cas de certains types de règles de surpasser en performance des outils spécialisés.

Au cours de nos travaux en fouille de données, notre attention s'est aussi portée sur l'utilisation de la logique propositionnelle pour la représentation des données [BJSS17a]. Nous avons particulièrement étudié la tâche de regroupement (*clustering* en anglais) sur des données complexes représentées par des formules. L'intérêt derrière l'utilisation de la logique propositionnelle dans ce contexte réside surtout dans le fait qu'il s'agit d'un cadre permettant de représenter des entités hétérogènes de manière compacte : une formule peut avoir un nombre exponentiel de modèles. Effectivement, les formules propositionnelles sont largement utilisées en intelligence artificielle pour la représentation de différentes informations, comme des connaissances, des croyances et des préférences.

L'utilisation d'approches fondées sur la logique propositionnelle pour des problèmes en fouille de données nous a mené à examiner d'autres problèmes connexes. Nous avons entre autres proposé dans [JLSS14] une nouvelle approche pour l'énumération des modèles d'une formule propositionnelle. De plus, nous avons introduit un nouvel encodage des contraintes de cardinalité dans [JSS13a, JSS14, HJSS17] de par leur présence récurrente dans nos différentes formulations.

Il est à préciser que nous avons aussi réalisé des travaux en fouille de données indépendamment de l'utilisation de la modélisation en logique propositionnelle. Nous avons en particulier introduit dans [JSST12] un cadre pour l'élimination des symétries dans l'extraction des motifs ensemblistes. En ce qui concerne aussi la notion de symétrie, nous avons proposé dans [JKS+13] un algorithme permettant l'élimination des symétries de manière dynamique dans la fouille de motifs ensemblistes. Nos travaux sur l'exploitation des symétries se sont poursuivis sur des plans similaires dans [BJSS14]. Par ailleurs, nous avons montré que l'utilisation de la fouille de données peut être bénéfique pour des cadres comme SAT et CSP. Nous avons en effet introduit dans [JSSU13, JRSS15] des méthodes de compression d'instances issues de ces deux cadres possédant comme base la fouille de motifs ensemblistes.

1.4 Représentation des connaissances et raisonnements

Nous décrivons ici brièvement nos travaux dans l'axe de recherche de la représentation des connaissances et raisonnements, en relation notamment avec l'utilisation de la logique propositionnelle. Nous aborderons en particulier certains travaux dans ce même axe qui ne seront pas détaillés dans ce mémoire.

Intéressons-nous, dans un premier temps, au raisonnement en présence de l'incohérence. Parmi nos contributions importantes dans ce contexte, mentionnons l'introduction de différentes mesures de l'incohérence [ARSO15, JMR+15, JMR+16, ASOR17]. De plus, nous avons établi plusieurs propriétés relatives à la compatibilité entre postulats utilisés dans la définition de telles mesures [ASOR17]. Il est à noter que dans le cadre de ce mémoire, nous décrirons dans le chapitre 5 une application des mesures de l'incohérence en fouille de données pour la tâche de regroupement. Toujours en lien avec le raisonnement sous l'incohérence, nous avons introduit un cadre simple et intuitif pour la définition de relations de conséquence logique paracohérentes, fondé sur une nouvelle notion nommée fonction de conséquence [Sal19a].

La théorie de l'argumentation inclut des processus de raisonnement ayant comme base la construction d'arguments et la caractérisation de ceux pouvant être acceptés à partir de conflits existant entre eux. Il y a notamment l'approche abstraite de Dung, où l'accent est mis sur les relations entre arguments, en particulier les relations d'attaque, et non la structure interne de ces derniers [Dun95]. Nous pouvons également mentionner l'approche reposant sur l'utilisation des logiques formelles et prenant en compte la structure interne des arguments [BH08].

Dans [KS14], nos travaux se rapportent à l'étude de l'approche fondée sur les logiques formelles en proposant un cadre d'argumentation permettant de raisonner de

manière constructive par l'utilisation de la logique intuitionniste. Concernant nos travaux dans [JRSS16], nous avons proposé une méthode pour l'intégration de préférences dans l'approche de Dung. De plus, nous avons introduit dans [Sal19c] un cadre pour la persuasion, où nous utilisons une nouvelle structure d'argument. Il importe de noter que nous faisons appel dans ce cadre à l'optimisation en logique propositionnelle en utilisant des formulations dans le problème MaxSAT partiel pondéré.

Dans le cadre de la thèse de M. Sioutis, nous avons apporté plusieurs contributions au sujet du raisonnement qualitatif. Ces contributions concernent principalement la définition de formalismes qualitatifs combinant le raisonnement spatial et celui temporel. Notamment, un nouveau cadre qualitatif spatio-temporel a été introduit dans [SCSM14]. Une étude en termes de complexité et de système formel a également été proposée pour une logique spatio-temporelle, combinant une logique temporelle avec un formalisme qualitatif spatial [SCSM15]. Nous avons par ailleurs proposé une étude sur les effets des techniques de décomposition pour le raisonnement qualitatif spatio-temporel [SSC15a, SSC17], où nous avons particulièrement mis en lumière une erreur dans une méthode de décomposition existante dans la littérature. En outre, nous avons effectué d'autres travaux dans le cadre du raisonnement qualitatif indépendamment de la thèse de M. Sioutis dans [CKS16, CRS16].

Dans ce mémoire, nous détaillerons uniquement une partie de nos travaux sur le raisonnement en présence de l'incohérence en abordant les mesures de l'incohérence et les relations de conséquence paracohérentes, ainsi que nos travaux sur la persuasion automatique dans le cadre de l'optimisation en logique propositionnelle.

1.5 Co-encadrements doctoraux et collaborations

La première thèse de doctorat que j'ai co-encadrée a été en collaboration avec l'université Mouloud Mammeri en Algérie. Il s'agit de la thèse de M. Ammoura, dont les travaux ont débuté en octobre 2012, et la soutenance a eu lieu en décembre 2016. Quant au sujet, il se rapportait aux mesures de l'incohérence. Les articles co-écrits dans ce cadre sont [ARSO15, ASOR17].

En octobre 2013, j'ai débuté le co-encadrement de la thèse de M. Sioutis sur le raisonnement qualitatif spatial et temporel, qui a été soutenue en février 2017. J'ai co-écrit dans le contexte de cette thèse les articles suivants [SCSM14, SCSM15, SSC15a, SCS+15, SS15, SSC15b, SSC17].

La troisième thèse de doctorat que j'ai co-encadrée est celle de A. Boudane, débutée en septembre 2015 et soutenue en septembre 2018. Le sujet gravitait autour de la modélisation de problèmes de fouille de données en logique propositionnelle. Les articles co-écrits en relation avec cette thèse sont [BJSY16, BJSS17a, BJSS17b, BJSS17c, BJSS18].

Je co-encadre depuis septembre 2016, la thèse de M. Y. Boumarafi, qui concerne la caractérisation de classes traitables en SAT, de même que certains autres problèmes qui en sont dérivés. Les articles co-écrits pour l'heure sont [BSS17, BS18].

Persuadé que la recherche est un travail d'équipe, où les échanges constituent un moteur stimulant et créateur, j'ai mené, parallèlement à mes contributions en autonomie,

une grande partie de mes travaux en collaboration avec d'autres chercheurs. Dans ce qui suit, je fournis la liste des co-auteurs :

Collaborations internationales

— Université of Birmingham, Royaume-Uni : David A. Randell (Senior Research Fellow).
— Université de Lisbonne, Portugal : João Marques-Silva (Professeur).
— Université Mouloud Mammeri, Algérie : Brahim Oukacha (Professeur).
— National Institue of Informatics, Japon : Takeaki Uno (Professeur).
— Université de Tunis, Tunisie : Boutheina Ben Yaghlane (Professeur).

Collaborations nationales

— Université Aix-Marseille, LSIS : Belaid Benhamou (Maître de Conférences).
— Université Claude Bernard, Lyon, LIRIS : Emmanuel Coquery (Maître de Conférences).
— Université de Lorraine, LORIA : Dominique Larchey-Wendling (Chargé de recherche CNRS), Didier Galmiche (Professeur),
— Université de Montpellier, LIRMM : Souhila Kaci (Professeur).
— Université Paris-Saclay, LRI : Yue Ma (Maître de Conférences).
— Université d'Artois, CRIL : Lakhdar Sais (Professeur) ; Said Jabbour (Maître de Conférences) ; Jean-François Condotta (Professeur) ; Badran Raddaoui (ATER pendant la collaboration, actuellement Maître de Conférences) ; Bertrand Mazure (Professeur) ; Karim Tabia (Maître de Conférences) ; Jerry Lonlac (Post-doctorant) ; Mehdi Khiari (Post-doctorant) ; Stéphanie Roussel (Ingénieur de recherche) ; Imen Ouled Dlala (Doctorante).

1.6 Plan du mémoire

Dans la figure 1.2, nous décrivons les contributions détaillées dans ce manuscrit avec leur répartition sur les différents chapitres. Nous utilisons dans cette figure LP, FD et RCR pour faire référence à nos contributions dans respectivement la logique propositionnelle de manière générale, la fouille de données, et la représentation des connaissances et raisonnements.

Le chapitre 2 est principalement dédié à la présentation des approches abordées tout au long des chapitres suivants. Ainsi, après une description de la logique propositionnelle et du problème de cohérence SAT, nous introduisons notre encodage en SAT pour les contraintes de cardinalité. Nous décrivons ensuite les approches fondées sur la modélisation en SAT que nous considérons dans ce mémoire. Pour illustrer ces dernières, nous utilisons nos travaux sur l'énumération des impliquants premiers et des impliquants premiers essentiels.

Dans le chapitre 3, nous présentons une partie importante de nos contributions concernant l'utilisation de formulations en SAT pour l'extraction de plusieurs types de motifs. Nous abordons premièrement des problèmes en fouille de données se rapportant aux motifs ensemblistes. Deuxièmement, nous considérons l'extraction de différents types de règles d'association. Nous examinons ensuite des problèmes liés aux motifs séquentiels.

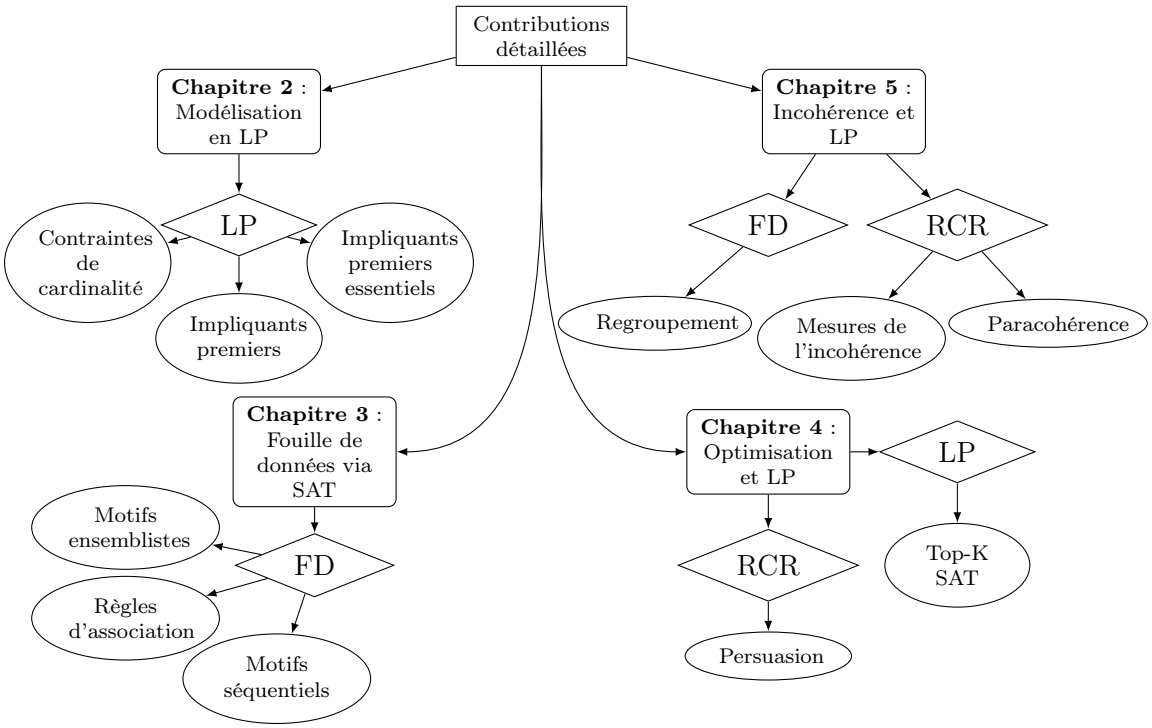


FIGURE 1.2 – Les contributions détaillées dans ce manuscrit : LP (Logique Propositionnelle), FD (Fouille de données), RCR (Représentation des connaissances et raisonnements)

Enfin, nous décrivons d'intéressants résultats expérimentaux issus d'une étude comparative. L'objectif visé par ce chapitre est de montrer que la modélisation en SAT, de par la modularité et la flexibilité de cette approche, constitue un cadre générique adapté pour l'extraction de différents types de motifs.

Le chapitre 4 porte sur nos travaux dans le cadre de l'optimisation en logique propositionnelle. Nous débutons par une brève description des problèmes d'optimisation liés à SAT les plus étudiés dans la littérature. Après cela, nous présentons un problème d'optimisation reposant sur SAT que nous avons introduit, appelé Top-K SAT. Enfin, nous introduisons un cadre pour la persuasion automatique, où nous utilisons des formulations dans une variante d'optimisation dérivée de SAT.

Le chapitre 5 est consacré à certains de nos résultats associés au raisonnement en présence de l'incohérence en logique propositionnelle. Dans un premier temps, nous présentons ceux en relation avec la notion de mesure de l'incohérence. Dans un deuxième temps, nous fournissons une description de notre approche pour la définition de relations de conséquence logique paracohérentes. Ensuite, nous introduisons une application des mesures de l'incohérence en fouille de données, et cela, en considérant la tâche de regroupement dans un cadre où les données sont représentées par des formules propositionnelles.

Pour conclure, nous exposons un ensemble de pistes de recherche en relation avec nos travaux.

Chapitre 2

Modélisation en logique propositionnelle

Ce chapitre a pour objet la description de différents éléments ayant trait à l'utilisation de la logique propositionnelle comme outil de modélisation. Nous décrirons notamment certains types de modèles et de propriétés importantes s'y rattachant. Nous aborderons dans ce contexte nos travaux dans [JSS13a, JSS14, JLSS14, JMSS14, Sal18].

2.1 Syntaxe et sémantique

Dans la syntaxe en logiques formelles, il s'agit de décrire la structure des formules logiques *bien formées*, autrement dit, le *langage* de la logique en question. Tout comme nous ne pouvons pas mettre les mots dans n'importe quel ordre dans un langage naturel, une formule est bien formée dans une logique formelle donnée uniquement lorsqu'elle satisfait certaines conditions décrites par la syntaxe de cette dernière.

Introduisons à présent les symboles, appelés *symboles primitifs*, qui correspondent aux briques de base dans la construction des formules :

- un ensemble dénombrable de variables propositionnelles notées en utilisant les lettres p, q, r, etc (avec éventuellement des indices) ;
- les constantes $\bot$ et $\top$ représentant respectivement les valeurs de vérité *vrai* et $faux$;
- les opérateurs logiques $\vee$ (disjonction), $\wedge$ (conjonction), $\rightarrow$ (implication), $\leftrightarrow$ (équivalence), $\neg$ (négation) ;
- les deux signes de ponctuation (et).

Intuitivement, les symboles primitifs peuvent être vus comme les « mots » de la logique propositionnelle.

En utilisant les symboles primitifs, les *formules propositionnelles* sont définies par induction comme suit :

- toute variable propositionnelle, comme toute constante, est une formule propositionnelle (on parle dans ce cas de *formules atomiques* ou *atomes*) ;
- si ϕ et ψ sont des formules propositionnelles, alors $(\phi \vee \psi)$, $(\phi \wedge \psi)$, $(\phi \rightarrow \psi)$, $(\phi \leftrightarrow \psi)$ et $(\neg\phi)$ le sont également ;
- toute séquence finie de symboles primitifs est une formule propositionnelle si et seulement si elle est construite avec les deux règles précédentes.

Nous utiliserons dans ce qui suit les lettres grecques ϕ, ψ et χ (avec éventuellement des indices) pour représenter les formules propositionnelles. Par ailleurs, nous utiliserons Prop et Form afin de nous référer à respectivement l'ensemble des variables propositionnelles et celui des formules propositionnelles. De plus, étant donné une formule propositionnelle ϕ, nous utiliserons $Var(\phi)$ pour représenter l'ensemble des variables propositionnelles apparaissant dans la formule ϕ.

Par exemple, la séquence de symboles primitifs $\phi = (((p \vee q) \rightarrow (r_1 \wedge r_2)) \leftrightarrow (\neg q))$ est une formule propositionnelle, alors que la séquence $(p\ q \vee \wedge)$ ne l'est pas. En outre, on a $Var(\phi) = \{p, q, r_1, r_2\}$.

Un *littéral* est soit une variable propositionnelle, appelée *littéral positif*, soit la négation d'une variable propositionnelle, appelée *littéral négatif*. Étant donné un littéral l, nous utiliserons $\bar{l}$ pour noter son complémentaire : si l est positif alors $\bar{l} = \neg p$, et s'il est négatif $\bar{l} = p$, où p est la variable utilisée dans l.

Intéressons-nous maintenant à la notion de *sous-formule*. Elle est définie par induction comme suit :

— ϕ est une sous-formule de ϕ ;
— si $(\psi\ op\ \chi)$ est une sous-formule de ϕ, alors ψ et χ le sont également pour tout $op \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$;
— si $(\neg\psi)$ est une sous-formule de ϕ, alors ψ l'est aussi.

Une *sous-formule propre* d'une formule ϕ est une sous-formule de ϕ différente de cette dernière. Reconsidérons encore la formule propositionnelle $(((p \vee q) \rightarrow (r_1 \wedge r_2)) \leftrightarrow (\neg q))$. L'ensemble de ses sous-formules est $\{(((p \vee q) \rightarrow (r_1 \wedge r_2)) \leftrightarrow (\neg q)), ((p \vee q) \rightarrow (r_1 \wedge r_2)), (p \vee q), (r_1 \wedge r_2), (\neg q), p, q, r_1, r_2\}$.

Afin d'alléger l'écriture des formules propositionnelles en omettant des parenthèses, nous suivons l'ordre décroissant de priorité suivant sur les opérateurs (règles de précédence) : $\neg > \wedge > \vee > \rightarrow > \leftrightarrow$. Par exemple, la formule $(((\neg p) \wedge q) \rightarrow (r \vee s))$ peut s'écrire de manière plus succincte $\neg p \wedge q \rightarrow r \vee s$. En outre, précisons que l'implication est associative à droite.

Décrivons à présent la *sémantique* de la logique propositionnelle. Rappelons d'abord que la sémantique pour une logique formelle permet l'étude du sens des formules bien formées, de la même manière que les phrases bien formées dans un langage naturel ont des sens et peuvent être interprétées.

Afin d'interpréter les formules propositionnelles, on utilise des fonctions, appelées *interprétations booléennes* (en l'honneur du mathématicien Georges Bool), associant des valeurs de vérité aux variables propositionnelles.

Définition 2.1. *Une interprétation booléenne $\mathcal{B}$ d'une formule propositionnelle ϕ est une fonction associant à chaque variable propositionnelle dans $Var(\phi)$ une valeur dans $\{0, 1\}$, où 0 et 1 représentent respectivement $faux$ et $vrai$.*

Nous noterons parfois une interprétation booléenne par $\{p_1 \mapsto v_1, \ldots, p_n \mapsto v_n\}$ pour exprimer que la variable propositionnelle p_i prend la valeur de vérité $v_i \in \{0, 1\}$ pour $i \in \{1, \ldots, n\}$.

Les interprétations booléennes sont étendues par induction aux formules propositionnelles comme suit :

— $\mathcal{B}(\bot) = 0$ et $\mathcal{B}(\top) = 1$;
— $\mathcal{B}(\neg\phi) = 1$ si $\mathcal{B}(\phi) = 0$, $\mathcal{B}(\neg\phi) = 0$ sinon ;
— $\mathcal{B}(\phi \vee \psi) = 1$ si $\mathcal{B}(\phi) = 1$ ou $\mathcal{B}(\psi) = 1$, $\mathcal{B}(\phi \vee \psi) = 0$ sinon ;
— $\mathcal{B}(\phi \wedge \psi) = 1$ si $\mathcal{B}(\phi) = 1$ et $\mathcal{B}(\psi) = 1$, $\mathcal{B}(\phi \wedge \psi) = 0$ sinon ;
— $\mathcal{B}(\phi \rightarrow \psi) = 1$ si $\mathcal{B}(\phi) = 0$ ou $\mathcal{B}(\psi) = 1$, $\mathcal{B}(\phi \rightarrow \psi) = 0$ sinon ;
— $\mathcal{B}(\phi \leftrightarrow \psi) = 1$ si $\mathcal{B}(\phi) = \mathcal{B}(\psi)$, $\mathcal{B}(\phi \leftrightarrow \psi) = 0$ sinon.

Considérons par exemple la formule $\phi = (p \vee q) \rightarrow (p \wedge q)$ et deux de ses interprétations booléennes $\mathcal{B} = \{p \mapsto 1, q \mapsto 0\}$ et $\mathcal{B}' = \{p \mapsto 1, q \mapsto 1\}$. Sachant que $\mathcal{B}(p \vee q) = 1$, $\mathcal{B}'(p \vee q) = 1$, $\mathcal{B}(p \wedge q) = 0$ et $\mathcal{B}'(p \wedge q) = 1$, on obtient $\mathcal{B}(\phi) = 0$ et $\mathcal{B}'(\phi) = 1$.

Définition 2.2 (Modèle)**.** *Un modèle d'une formule propositionnelle ϕ est une interprétation booléenne $\mathcal{B}$ de cette dernière telle que $\mathcal{B}(\phi) = 1$.*

En d'autres termes, un modèle d'une formule propositionnelle est une interprétation booléenne rendant cette formule vraie. Nous utiliserons $Mods(\phi)$ pour noter l'ensemble des modèles de ϕ.

Définition 2.3 (Satisfiabilité)**.** *Une formule est satisfiable (ou cohérente) si elle admet au moins un modèle.*

Définition 2.4 (Validité)**.** *Une formule est valide (ou un théorème) si toutes ses interprétations booléennes sont des modèles.*

En logique propositionnelle classique, il existe clairement une complémentarité entre la notion de satisfiabilité et celle de validité : ϕ est valide si et seulement si $\neg\phi$ n'est pas satisfiable.

On dit que deux formules ϕ et ψ sont *équivalentes*, écrit $\phi \equiv \psi$, si $\phi \leftrightarrow \psi$ est une formule valide. On peut par exemple noter la présence des équivalences suivantes dans la logique propositionnelle : $\top \equiv \bot \rightarrow \bot$ et $\phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$.

Par ailleurs, soit Γ un ensemble fini de formules propositionnelles et ϕ une formule propositionnelle. On dit que ϕ est une *conséquence logique* de Γ, écrit $\Gamma \vdash \phi$, si $\bigwedge \Gamma \rightarrow \phi$ est une formule valide, avec $\bigwedge\{\psi_1, \ldots, \psi_k\} = \psi_1 \wedge \cdots \wedge \psi_k$ et $\bigwedge \emptyset = \top$.

Introduisons maintenant quelques conventions de notation. Étant donné une interprétation booléenne $\mathcal{B}$ et un ensemble de littéraux $\{l_1, \ldots, l_k\}$ tel que $Var(l_i) \neq Var(l_j)$ pour tous $1 \leqslant i, j \leqslant k$ avec $i \neq j$, nous utiliserons $\mathcal{B}_{\{l_1 \mapsto v_1, \ldots, l_k \mapsto v_k\}}$, avec $v_1, \ldots, v_k \in \{0, 1\}$, pour noter l'interprétation booléenne $\mathcal{B}'$ portant sur le même ensemble de variables que $\mathcal{B}$ et définie comme suit :

$$\mathcal{B}'(p) = \begin{cases} \mathcal{B}(p) & si\ p \notin \{Var(l_i) \mid 1 \leqslant i \leqslant k\} \\ v_i & si\ \exists i \in 1..k,\ p = l_i \\ 1 - v_i & si\ \exists i \in 1..k,\ \neg p = l_i \end{cases}$$

De plus, étant donné une interprétation booléenne $\mathcal{B}$ d'une formule ϕ et un ensemble de variables propositionnelles E tel que $E \subseteq Var(\phi)$, nous utiliserons $\mathcal{B}_{|E}$ pour noter la restriction de $\mathcal{B}$ à l'ensemble E.

2.2 Le problème SAT

Dans cette section, nous considérons le problème de cohérence en logique propositionnelle, appelé SAT, qui est certainement l'un des problèmes NP-complets les plus étudiés dans la littérature. L'intérêt accordé à ce problème vient probablement du fait qu'il regroupe deux caractéristiques importantes : la simplicité de sa définition et la puissance de son expressivité.

Le problème SAT.
- Entrée : une formule propositionnelle ϕ.
- Sortie : déterminer si ϕ est satisfiable (admet un modèle).

Dans la majeure partie des outils proposés dans la littérature pour résoudre le problème SAT, les formules propositionnelles en entrée doivent être en forme normale conjonctive. Une formule est en *forme normale conjonctive* (en anglais Conjunctive Normal Form, CNF) si elle est une conjonction de clauses, où une *clause* est une disjonction de littéraux. Par exemple, $p \vee q$ est une clause et $(p \vee q) \wedge (\neg p \vee q) \wedge (\neg q)$ est en forme normale conjonctive. Nous écrirons *formules CNF* pour nous référer aux formules en forme normale conjonctive.

Etat donné une formule CNF ϕ, nous utiliserons $Lit(\phi)$ pour noter l'ensemble des littéraux apparaissant dans les clauses de ϕ.

En employant l'élimination de la double négation ($\neg\neg\phi \equiv \phi$), les lois de De Morgan et des lois relatives à la distributivité, toute formule propositionnelle peut être transformée en une formule CNF équivalente. Cependant, cette transformation peut entraîner une formule CNF de taille exponentielle en la taille de la formule d'origine, où la taille d'une formule correspond au nombre de symboles qu'elle contient. Pour s'en convaincre, considérons la formule propositionnelle $(p_1 \wedge q_1) \vee \cdots \vee (p_n \wedge q_n)$. La formule CNF que l'on obtient par la précédente transformation est $\bigwedge_{X \subseteq \{1,\ldots,n\}} (\bigvee_{i \in X} p_i \vee \bigvee_{j \in \{1,\ldots,n\} \setminus X} q_j)$, qui est une formule contenant un nombre de clauses égal à 2^n.

Il est possible d'éviter l'explosion exponentielle avec une transformation préservant la satisfiabilité au lieu de construire une formule CNF équivalente.

Définition 2.5 (Équi-satisfiabilité)**.** *Deux formules propositionnelles ϕ et ψ sont* équi-satisfiables *lorsque ϕ est satisfiable si et seulement si ψ est satisfiable.*

Par exemple, les deux formules $(p \wedge q)$ et $(p' \wedge q')$, avec p, q, p' et q' des variables deux à deux distinctes, sont équi-satisfiables sans être pour autant équivalentes.

Il existe une approche, proposée initialement par Tseitin [Tse68], pour transformer toute formule propositionnelle en une formule CNF qui lui est équi-satisfiable avec une augmentation linéaire de la taille. Le point clé de cette transformation consiste à associer par des implications logiques de nouvelles variables aux occurrences de sous-formules. Considérons encore une fois la formule $\phi = (p_1 \wedge q_1) \vee \cdots \vee (p_n \wedge q_n)$. En associant à chaque occurrence de sous-formule $(p_i \wedge q_i)$ la nouvelle variable r_i, nous obtenons la formule CNF équi-satisfiable de taille linéaire en la taille de ϕ suivante : $(r_1 \vee \cdots \vee r_n) \wedge (\neg r_1 \vee p_1) \wedge (\neg r_1 \vee q_1) \wedge \cdots \wedge (\neg r_n \vee p_n) \wedge (\neg r_n \vee q_n)$. Il convient ici de noter que toute paire de clauses de la forme $(\neg r_i \vee p_i) \wedge (\neg r_i \vee q_i)$ est équivalente à

l'implication $r_i \rightarrow (p_i \wedge q_i)$.

Pour des raisons de commodité, nous considérons parfois dans ce manuscrit une formule CNF comme un ensemble de clauses et une clause comme un ensemble littéraux. Cela est permis principalement grâce au fait que toute clause de la forme $l \vee l \vee c$ est équivalente à $l \vee c$, et que toute formule CNF de la forme $c \wedge c \wedge \phi$ est équivalente à $c \wedge \phi$.

Un des algorithmes connus pour résoudre le problème SAT est l'algorithme de retour sur trace (*backtracking*) nommé DPLL (*Davis-Putnam-Logemann-Loveland*), voir par exemple [BHvW09]. L'idée principale derrière ce dernier consiste à choisir une variable propositionnelle et lui affecter une valeur de vérité, pour ensuite simplifier la formule en fonction de ce choix ; et si la formule simplifiée est incohérente, on retourne en arrière pour affecter à la variable choisie la valeur de vérité opposée. Dans ce contexte, la simplification comporte essentiellement la suppression des clauses vraies et l'élimination des littéraux faux.

Les solveurs SAT modernes, quant à eux, reposent sur l'algorithme CDCL (*Conflict-Driven Clause Learning*) [SS96, SS99] qui peut être vu comme une amélioration de l'algorithme DPLL par de nouveaux mécanismes. Cet algorithme utilise notamment une méthode d'apprentissage de nouvelles clauses à partir des conflits permettant ainsi d'élaguer l'espace de recherche.

Une manière naïve d'étendre les solveurs SAT modernes au problème de l'énumération de tous les modèles d'une formule CNF consiste à ajouter pour chaque modèle trouvé une clause correspondant à sa négation afin d'empêcher la recherche de retourner de nouveau ce même modèle. Le principal désavantage de cette approche concerne la complexité notamment en espace, car le nombre de modèles, et en conséquence le nombre de clauses ajoutées, peut être exponentiel. En effet, aux clauses apprises à partir des conflits dans un solveur CDCL, viennent s'ajouter les clauses permettant l'exclusion des modèles trouvés. C'est pour cette raison qu'il est important d'utiliser des méthodes évitant de garder toutes les clauses correspondant aux négations des modèles à exclure. Dans ce contexte, nous avons proposé une approche combinant une recherche similaire à l'algorithme DPLL avec un solveur CDCL [JLSS14]. Intuitivement, à chaque modèle trouvé par une procédure CDCL, une procédure similaire à DPLL est utilisée pour retrouver les modèles qui lui sont proches.

2.3 Les contraintes de cardinalité

Les contraintes de cardinalité seront utilisées pour modéliser en logique propositionnelle différents problèmes considérés dans ce mémoire, notamment en fouille de données. C'est pour cette raison que nous exposons ici notre encodage de ces contraintes en formules CNF fondé sur une idée simple et intuitive [JSS13b, JSS14]. Il est important de mentionner que de nombreux autres encodages polynomiaux de ces contraintes existent dans la littérature (par exemple voir [BB03, Sin05, ES06, SL07, BBR09]).

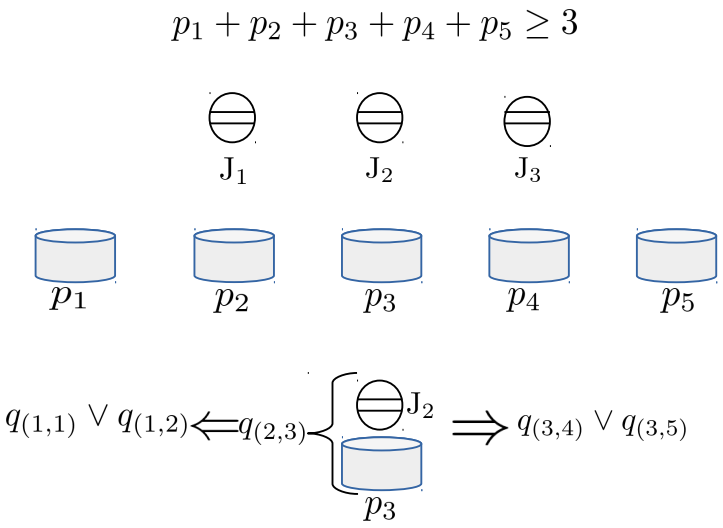


FIGURE 2.1 – Une approche pour l'encodage des contraintes de cardinalité

Rappelons qu'une contrainte de cardinalité est une expression de la forme suivante :

$$\sum_{i=1}^{n} p_i \geqslant \alpha$$

où $p_{i \in 1..n}$ est une variable propositionnelle et α un nombre entier naturel. Un interprétation $\mathcal{B}$ satisfait la précédente contrainte de cardinalité si $|\{\mathcal{B}(p_i) = 1 \mid i = 1..n\}| \geqslant \alpha$.

Pour simplifier la compréhension de notre encodage, nous regardons une contrainte de cardinalité comme une expression traduisant la mise d'au moins α jetons dans n emplacements possibles de telle sorte que chacun des emplacements ne peut contenir qu'au plus un jeton. Ainsi, afin de définir notre encodage nous considérerons un ensemble de α jetons distincts $\mathcal{J} = \{J_1, \ldots, J_\alpha\}$ avec le fait que les variables propositionnelles de la forme p_i joueront le rôle des emplacements disponibles.

Pour notamment réduire le nombre de variables propositionnelles utilisées dans notre encodage, nous imposons comme exigence ce qui suit : si le $i^{\text{ème}}$ jeton J_i est mis dans l'emplacement p_j, alors tous les jetons J_k pour $k < j$ devront être mis dans les emplacements suivants : $\{p_{j'} \mid j' < j\}$. Autrement dit, le $i^{\text{ème}}$ jeton devra toujours être mis après tous les jetons possédant des indices inférieurs à i. Il est clair que cette exigence ne change rien à la nature de la contrainte vu que tous les jetons sont identiques.

Notre encodage de la contrainte de cardinalité est défini en associant à chaque jeton $n - (\alpha - 1)$ variables propositionnelles distinctes. Ces variables sont utilisées pour représenter les emplacements possibles de chacun des jetons. Plus précisément, pour tout jeton $J_i \in \mathcal{J}$, on associe $n - (\alpha - 1)$ variables propositionnelles $q_{(i,i)}, \ldots, q_{(i,n-(\alpha-i))}$, où $q_{(i,j)}$ représente le fait que le jeton i est mis à l'emplacement p_j. Notons que nous associons au jeton J_i uniquement les emplacements $p_i, \ldots, p_{n-(\alpha-i)}$ dans l'objectif de laisser suffisamment d'emplacements aux jetons précédents et suivants. Par exemple, la mise du jeton J_1 dans l'emplacement p_n ne laissera aucun emplacement disponible aux

autres jetons si nous tenons compte de l'exigence décrite précédemment.

La première formule de notre encodage sert uniquement à mettre en relation les nouvelles variables associées aux jetons et les variables de la contrainte de cardinalité :

$$\bigwedge_{i=1}^{\alpha} \bigwedge_{j=i}^{n-(\alpha-i)} (\neg q_{(i,j)} \vee p_j) \tag{2.1}$$

La seconde formule permet d'exprimer que tout jeton occupe au moins un emplacement :

$$\bigwedge_{i=1}^{\alpha} \bigvee_{j=i}^{n-(\alpha-i)} q_{(i,j)} \tag{2.2}$$

La dernière formule traduit l'exigence selon laquelle le jeton J_i doit toujours être mis après tous ceux possédant des indices inférieurs à i :

$$\bigwedge_{i=2}^{\alpha} \bigwedge_{j=i}^{n-(\alpha-i+1)} (\neg q_{(i,j)} \vee \bigvee_{k=i-1}^{j-1} q_{(i-1,k)}) \tag{2.3}$$

Notre encodage, noté $\mathcal{CC}(\sum_{i=1}^{n} p_i \geqslant \alpha)$, est ainsi défini comme la conjonction des trois précédentes formules (2.1) $\wedge$ (2.2) $\wedge$ (2.3).

Nous démontrons dans ce qui suit que notre encodage satisfait *la propriété de cohérence d'arcs généralisée* [Bes06] par propagation unitaire, qui est une des propriétés importantes relatives à l'efficacité de résolution en présence de contraintes de cardinalité. Dans le cas d'un encodage de la contrainte $\sum_{i=1}^{n} p_i \geqslant \alpha$, cette propriété revient à vérifier les deux propriétés suivantes : (a) pour toute interprétation partielle attribuant à au moins $n - \alpha + 1$ variables la valeur 0, la propagation unitaire doit entraîner une contradiction, en l'occurence la clause vide, et (b) pour toute interprétation partielle attribuant à $n - \alpha$ variables la valeur 0, alors la propagation unitaire doit attribuer à toutes les autres variables la valeur 1. Le choix de la propagation unitaire vient du fait qu'il s'agit d'une procédure possédant une complexité en temps linéaire et que les outils modernes pour SAT la mettent en œuvre de manière très efficace.

Rappelons que la *propagation unitaire* est une procédure consistant à appliquer de manière itérative les deux règles suivantes pour toute clause unitaire l après avoir attribué la valeur de vérité à la variable correspondante (la valeur 0 si l est négatif, la valeur 1 sinon) : (i) supprimer toutes les clauses contenant l, et (ii) supprimer $\bar{l}$ de toutes les clauses contenant ce littéral. Par exemple, l'application de la propagation unitaire à la formule $p \wedge (\neg p \vee \neg q) \wedge (p \vee r) \wedge (q \vee r)$ produira le modèle suivant $\{p \mapsto 1, q \mapsto 0, r \mapsto 1\}$. En effet, l'application des deux règles avec le littéral positif p (attribuer à p la valeur 1) supprime la clause $p \vee r$ et produit $\neg q$ en supprimant $\neg p$ de la clause $\neg p \vee \neg q$; la propagation du littéral négatif $\neg q$ (attribuer à q la valeur 0) dans la clause $q \vee r$ produira le littéral r (attribuer à r la valeur 1).

Proposition 2.1 (Cohérence d'arcs généralisée)**.** *L'encodage* $\mathcal{CC}(\sum_{i=1}^{n} p_i \geqslant \alpha)$ *satisfait la propriété de cohérence d'arcs généralisée.*

Démonstration. Nous ne démontrerons ici que la propriété (b), car la propriété (a) peut en être aisément obtenue. Soit $\mathcal{B}$ une interprétation partielle de $\mathcal{CC}(\sum_{i=1}^{n} p_i \geqslant \alpha)$ attribuant 0 aux variables dans l'ensemble $\{p_{i_1}, \ldots, p_{i_{n-\alpha}}\}$. On suppose sans perte de généralité que $i_1 < \cdots < i_{n-\alpha}$. En utilisant la propagation unitaire (PU) sur (2.1), on obtient $q_{(i,j)} \mapsto 0$ pour toute variable $q_{(i,j)}$ avec $j \in I = \{i_1, \ldots, i_{n-\alpha}\}$. Par ailleurs, notons que la formule (2.3) comporte la conjonction des clauses binaires suivantes :

$$\neg q_{(i,i)} \vee q_{(i-1,i-1)} \ \textit{pour} \ i = 2..\alpha \tag{2.4}$$

Ainsi, en utilisant PU sur (2.4), on obtient $q_{(i,i)} \mapsto 0$ pour tout $i \in i_1..\alpha$, ce qui produira par PU les clauses binaires suivantes en utilisant les clauses ternaires dans (2.3) :

$$\neg q_{(i,i+1)} \vee q_{(i-1,i)} \ \textit{pour} \ i = i_1..\alpha \tag{2.5}$$

De la même manière, en utilisant PU sur (2.5), on obtient $q_{(i,i+1)} \mapsto 0$ pour tout $i \in i_2..\alpha$. Ainsi, en poursuivant l'application de PU sur (2.3), on obtient que pour tout $i \in E_\alpha = \{\alpha, \ldots, n\} \setminus I$ avec i différent de la valeur maximale dans E_α, notée j_α, $q_{(\alpha,i)} \mapsto 0$ et donc par PU sur (2.2) on a $q_{(\alpha,j_\alpha)} \mapsto 1$. En conséquence, par PU sur (2.2), on obtient $p_{j_\alpha} \mapsto 1$. On procède de la même façon jusqu'à obtenir $p_{j_1} \mapsto 1, \ldots, p_{j_{\alpha-1}} \mapsto 1$, où j_k est la valeur maximale dans $E_k = \{k, \ldots, n - (\alpha - k)\} \setminus (I \cup \{j_l \mid l > k\})$ pour $k = 1..(\alpha - 1)$. □

2.4 Problèmes d'énumération de solutions via SAT

Dans cette section, nous illustrons des aspects en lien avec la manière dont nous pouvons formuler en SAT un problème d'énumération de solutions. Nous mettons particulièrement l'accent sur certaines propriétés afférentes aux formulations en SAT de ce type de problèmes qui seront abordées à plusieurs endroits de ce manuscrit.

2.4.1 Avantages de l'utilisation de SAT

Comme mentionné précédemment, le problème de satisfiabilité en logique propositionnelle classique allie deux qualités essentielles, une simplicité qui lui confère un caractère d'outil de modélisation naturel, et une force dans l'expressivité permettant le traitement de problèmes complexes avec des représentations compactes. Le problème SAT est encore plus adapté à la modélisation dans un contexte où des contraintes de différents types peuvent être imposées sur les solutions recherchées. Cela est notamment le cas en fouille de données, où le nombre souvent important des motifs générés par des tâches de base requiert fréquemment des sélections sous plusieurs formes pour plus de pertinence. De telles sélections peuvent habituellement être réalisées dans le cadre du problème SAT en ajoutant par conjonction de nouvelles formules à des formulations de départ.

Par ailleurs, la constante évolution des solveurs SAT depuis plusieurs années constitue un argument central en faveur de l'élargissement du champ d'application des approches fondées sur la modélisation en SAT. Il est intéressant de noter que les solveurs modernes ont montré leur efficacité sur des instances de tailles très importantes issues de nombreuses applications industrielles, comme en particulier la vérification de modèle [BK18]. Cela mène à penser que ces solveurs sont adaptés pour des problèmes de fouille de données où nous avons souvent affaire à des bases de données volumineuses.

2.4.2 Formulation bijective

Nous fournissons ici une description de l'approche reposant sur SAT où l'ensemble des modèles est en bijection avec l'ensemble des solutions du problème considéré. L'intérêt de cette propriété réside dans le fait qu'elle permet d'adapter aisément la formulation pour différentes variantes du problème d'origine. Par exemple, avec la propriété de bijection, le problème de comptage de solutions revient tout simplement à compter le nombre de modèles de la formulation en SAT.

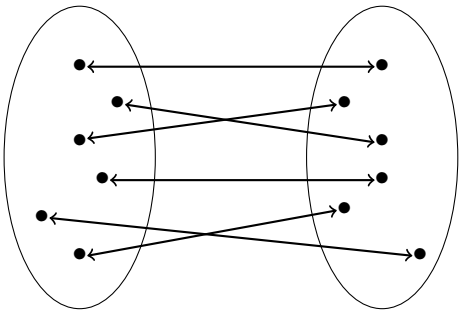


FIGURE 2.2 – Formulation en SAT bijective

Afin de présenter l'approche de manière concrète, nous considérons comme exemple le problème de l'énumération des impliquants premiers. En effet, nous décrivons ici notre formulation en SAT où une formule propositionnelle est associée à chaque instance du précédent problème, de telle sorte que les modèles de cette formule représentent tous les impliquants premiers de l'instance considérée [JMSS14].

Un *impliquant* d'une formule ϕ est un ensemble fini de littéraux I tel que $(\bigwedge_{l \in I} l) \rightarrow \phi$ est une formule valide.

Définition 2.6 (Impliquant premier)**.** *Soient ϕ une formule propositionnelle et I un impliquant de ϕ. On dit que I est un* impliquant premier *de ϕ si pour tout I' sous-ensemble propre de I ($I' \subset I$), I' n'est pas un impliquant de ϕ.*

Considérons par exemple la formule $\phi = (p \vee \neg q \vee r) \wedge (\neg p \vee \neg r) \wedge (q \vee \neg r)$. Cette formule admet $I = \{p, \neg r\}$ comme impliquant premier. En effet, le fait que p soit vrai permet de satisfaire la première clause, et le fait que r soit faux satisfait les deux clauses restantes ; de plus, satisfaire seulement p ou seulement $\neg r$ ne permet pas de satisfaire ϕ.

Problème de l'énumération des impliquants premiers (PEIP).
- Entrée : une formule CNF ϕ.
- Sortie : les impliquants premiers de ϕ.

Pour des raisons de commodité, nous considérons ici que les formules CNF ne peuvent pas contenir de clause tautologique. Une clause est dite tautologique si elle contient à la fois un littéral et sa négation. Il est aisé de voir qu'un ensemble de littéraux est un

impliquant premier d'une formule CNF si et seulement s'il est un impliquant premier de cette formule après suppression de toutes les clauses tautologiques.

Introduisons à présent notre formulation en SAT permettant de résoudre PEIP. Soit ϕ une formule CNF fournie en entrée. Nous associons à chaque littéral l apparaissant dans ϕ une nouvelle variable propositionnelle notée x_l. Nous notons $R(\phi)$ la formule CNF obtenue à partir de ϕ par : (i) le remplacement de chaque littéral l apparaissant dans cette dernière par sa variable associée x_l, et (ii) l'ajout de la clause $\neg x_p \vee \neg x_{\neg p}$ pour toute variable p telle que $p, \neg p \in Lit(\phi)$. Par exemple, considérons encore une fois la formule $\phi = (p \vee \neg q \vee r) \wedge (\neg p \vee \neg r) \wedge (q \vee \neg r)$. Alors, $R(\phi) = (x_p \vee x_{\neg q} \vee x_r) \wedge (x_{\neg p} \vee x_{\neg r}) \wedge (x_q \vee x_{\neg r}) \wedge (\neg x_p \vee \neg x_{\neg p}) \wedge (\neg x_q \vee \neg x_{\neg q}) \wedge (\neg x_r \vee \neg x_{\neg r})$.

Nous pouvons constater que les deux formules ϕ et $R(\phi)$ sont équi-satisfiables, car en particulier les clauses de la forme $\neg x_l \vee \neg x_{\bar{l}}$ permettent d'éviter d'affecter la même valeur de vérité à la fois à un littéral et à son complémentaire.

Pour établir une bijection entre les modèles de notre formulation et la formule CNF fournie en entrée, nous ajoutons par conjonction la formule suivante à $R(\phi)$ pour chaque littéral l apparaissant dans ϕ :

$$x_l \rightarrow \neg\Big(\bigwedge_{c \in R(\phi), x_l \in c} c \setminus \{x_l\}\Big) \tag{2.6}$$

Cette formule énonce que si un littéral est vrai alors il existe une clause qu'il est le seul à satisfaire.

Nous utiliserons $\mathcal{F}_{PEIP}(\phi)$ pour noter la formulation en SAT que nous venons de décrire : $R(\phi) \wedge (2.6)$.

Nous démontrons maintenant trois propriétés essentielles établissant l'adéquation de la formulation en SAT avec le problème de l'énumération des impliquants premiers. La première propriété, appelée *correction*, sert à démontrer que tous les modèles de la formulation représentent des solutions, des impliquants premiers dans le cas présent. Ensuite, la deuxième propriété, appelée *complétude*, est utilisée pour montrer que toutes les solutions sont représentées. Enfin, la troisième propriété, appelée *non-redondance*, permet de montrer qu'il n'existe pas de solution représentée par deux modèles distincts de la formulation. Cette dernière propriété montre donc qu'il y a une bijection entre l'ensemble des modèles et celui des solutions.

Proposition 2.2 (Correction)**.** *Si $\mathcal{B}$ est un modèle de $\mathcal{F}_{PEIP}(\phi)$, alors l'ensemble de littéraux $I_\mathcal{B} = \{l \in Lit(\phi) \mid \mathcal{B}(x_l) = 1\}$ est un impliquant premier de ϕ.*

Démonstration. En utilisant le fait que ϕ et $R(\phi)$ sont équi-satisfiables et que $R(\phi)$ n'est rien d'autre qu'un renommage des littéraux de ϕ (tout littéral l est renommé par x_l), on obtient que $I_\mathcal{B}$ est un impliquant de ϕ. Supposons maintenant que $I_\mathcal{B}$ n'est pas un impliquant premier de ϕ. Alors, il existe un littéral $l_0 \in I_\mathcal{B}$ tel que $I_\mathcal{B} \setminus \{l_0\}$ est un impliquant de ϕ. En partant de cela, on définit une interprétation $\mathcal{B}'$ de $\mathcal{F}_{PEIP}(\phi)$ comme suit :

$$\mathcal{B}'(x_l) = \begin{cases} \mathcal{B}(x_l) & si\ l \neq l_0 \\ 0 & sinon \end{cases}$$

Étant donné que $I_\mathcal{B} \setminus \{l_0\}$ est un impliquant de ϕ, l'interprétation $\mathcal{B}'$ est un modèle de $R(\phi)$. De plus, on a $\mathcal{B}(\psi) = \mathcal{B}'(\psi)$ avec $\psi = \bigwedge_{c \in R(\phi), x_{l_0} \in c} c \setminus \{x_{l_0}\}$ car x_{l_0} n'apparait

pas dans ψ. Ainsi, en utilisant le fait que $\mathcal{B}(x_{l_0}) = 1$ dans le contexte de la formule (2.6), on obtient $\mathcal{B}(\psi) = \mathcal{B}'(\psi) = 0$. Donc, en utilisant la définition de ψ, on obtient une contradiction avec le fait que $\mathcal{B}'$ est un modèle de $R(\phi)$. Par conséquent, en utilisant le raisonnement par l'absurde, $I_{\mathcal{B}}$ est un impliquant premier de ϕ. □

Proposition 2.3 (Complétude)**.** *Si I est un impliquant premier de ϕ, alors l'interprétation suivante $\mathcal{B}_I$ est un modèle de $\mathcal{F}_{PEIP}(\phi)$:*

$$\mathcal{B}_I(x_l) = \begin{cases} 1 & si\ l \in I \\ 0 & sinon \end{cases}$$

Démonstration. En utilisant le fait que ϕ et $R(\phi)$ sont équi-satisfiables, que $R(\phi)$ est simplement obtenu par renommage des littéraux de ϕ et que I est un impliquant premier de ϕ, $\mathcal{B}_I$ est un modèle de $R(\phi)$. Soit l un littéral dans ϕ. Si $\mathcal{B}_I(x_l) = 0$ alors on a clairement $\mathcal{B}_I(x_l \rightarrow \neg(\bigwedge_{c \in R(\phi), x_l \in c} c \setminus \{x_l\})) = 1$. Considérons maintenant le cas où $\mathcal{B}_I(x_l) = 1$. Ainsi on a $l \in I$. Si $\mathcal{B}_I(\bigwedge_{c \in R(\phi), x_l \in c} c \setminus \{x_l\}) = 1$ alors donner à l la valeur 1 n'est pas nécessaire pour satisfaire les clauses qui le contiennent, et par conséquent $I \setminus \{l\}$ est un impliquant de ϕ, ce qui est en contradiction avec le fait que I soit un impliquant premier. Donc, pour tout littéral l dans ϕ, on a $\mathcal{B}_I(x_l \rightarrow \neg(\bigwedge_{c \in R(\phi), x_l \in c} c \setminus \{x_l\})) = 1$. On en déduit que $\mathcal{B}_I$ est un modèle de $\mathcal{F}_{PEIP}(\phi)$. □

La propriété de non-redondance dans le cas de la formulation $\mathcal{F}_{PEIP}(\phi)$ est relativement triviale, car tout impliquant premier est représenté par la totalité du modèle qui lui correspond.

Proposition 2.4 (Non-redondance)**.** *Il n'existe pas deux modèles distincts $\mathcal{B}$ et $\mathcal{B}'$ de $\mathcal{F}_{PEIP}(\phi)$ tels que $I_{\mathcal{B}} = \{l \mid \mathcal{B}(x_l) = 1\} = I_{\mathcal{B}'} = \{l \mid \mathcal{B}'(x_l) = 1\}$.*

En combinant la non-redondance avec la correction et la complétude, nous aboutissons au fait que l'ensemble des modèles de $\mathcal{F}_{PEIP}(\phi)$ est en bijection avec celui des solutions de PEIP pour la formule CNF ϕ. Comme nous l'avons évoqué précédemment, le fait que la formulation soit bijective facilite son utilisation pour d'autres variantes du problème d'origine, comme en particulier le problème du comptage du nombre de solutions. Pour illustrer encore ce point, nous pouvons considérer une variante où, étant donné deux littéraux l et l', il s'agit de déterminer s'il y a plus d'impliquants premiers contenant l que d'impliquants premiers contenant l'. Dans ce cas, le nombre d'impliquants premiers contenant l (resp. l') peut simplement être obtenu en comptant le nombre de modèles de $\mathcal{F}_{PEIP}(\phi) \wedge x_l$ (resp. $\mathcal{F}_{PEIP}(\phi) \wedge x_{l'}$).

2.4.3 Formulation non bijective

Nous décrivons maintenant le cas où l'ensemble des modèles de la formulation en SAT n'est pas en bijection avec les solutions du problème considéré. Nous montrons particulièrement que l'absence de cette propriété n'est pas un frein à l'utilisation de SAT, car nous pouvons adapter l'énumération des modèles pour éviter la redondance dans la génération des solutions. Cela dit, l'absence de la propriété de bijection rend difficile l'adaptation de la formulation à certaines variantes du problème d'origine, en particulier, les problèmes relatifs au nombre de solutions. À titre d'illustration, nous

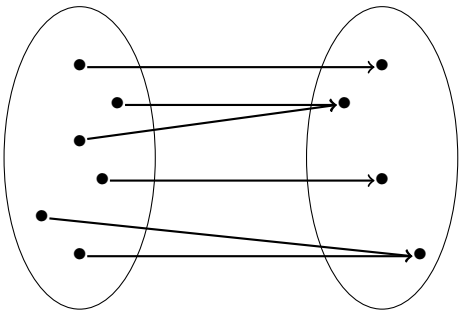


FIGURE 2.3 – Formulation en SAT non bijective

considérons comme exemple l'approche fondée sur SAT que nous avons proposée pour résoudre le problème de l'énumération des impliquants premiers essentiels [Sal18].

Définition 2.7 (Impliquant premier essentiel)**.** *Étant donné une formule propositionnelle ϕ, un impliquant premier I de ϕ est dit* essentiel *s'il existe un modèle $\mathcal{B}$ de ϕ tel que (i) $\mathcal{B}(\bigwedge_{l\in I} l) = 1$ et (ii) pour tout impliquant premier I' de ϕ différent de I, $\mathcal{B}(\bigwedge_{l\in I'} l) = 0$.*

En d'autres termes, un impliquant premier est essentiel s'il est le seul impliquant premier à couvrir un des modèles.

Problème de l'énumération des impliquants premiers essentiels (PEIP-E).
- Entrée : une formule CNF ϕ.
- Sortie : les impliquants premiers essentiels de ϕ.

De la même manière que dans le cas de PEIP, nous nous restreignons ici aux formules CNF qui ne contiennent pas de clause tautologique.

Algorithm 1: $Prime(\phi, \mathcal{B})$ pour calculer un impliquant premier à partir d'un modèle

Data: une formule CNF ϕ et un modèle $\mathcal{B}$ de ϕ.
Result: un impliquant premier de ϕ.
1 $S \leftarrow \{l \in Lit(\phi) \mid \mathcal{B}(l) = 1\}$;
2 $I \leftarrow S$;
3 **for** $l \in S$ **do**
4 | **if** $I \setminus \{l\}$ *est un impliquant de* ϕ **then** $I \leftarrow I \setminus \{l\}$;
5 **return** I;

On appelle *e-modèle* tout modèle qui n'est couvert que par un unique impliquant premier, qui est a fortiori essentiel. Plus précisément, un modèle $\mathcal{B}$ de ϕ est un e-modèle s'il existe un et un seul impliquant premier I tel que $\mathcal{B}(\bigwedge_{l\in I} l) = 1$.

Présentons maintenant quelques résultats importants concernant la complexité.

Théorème 2.1. *Le problème de décider si un modèle est un e-modèle est dans P.*

Démonstration. L'algorithme 2 permet de vérifier si un modèle est un e-modèle en temps polynomial. En effet, étant donné une formule CNF ϕ et un modèle $\mathcal{B}$ de ϕ, nous utilisons dans un premier temps l'algorithme 1 afin de calculer un impliquant premier quelconque I tel que $\mathcal{B}(\bigwedge_{l\in I} l) = 1$. On vérifie ensuite s'il existe un impliquant premier I' différent I tel que $\mathcal{B}(\bigwedge_{l\in I'} l) = 1$. Plus précisément, s'il existe $l \in I$ tel que $\mathcal{B}_{\{l\mapsto 0\}}$ satisfait ϕ, alors ϕ admet un impliquant qui ne contient pas l, et par conséquent, il existe un impliquant premier I' différent de I qui couvre $\mathcal{B}$. □

Algorithm 2: $IsEModel(\phi, \mathcal{M})$ pour vérifier si un modèle est un e-modèle

Data: une formule CNF ϕ et un modèle $\mathcal{B}$ de ϕ.
Result: Vrai ou Faux en fonction du fait si $\mathcal{B}$ est un e-modèle ou non.

```
I ← Prime(φ, B);
for l ∈ I do
    if B_{l↦0} satisfait φ then
        return Faux;
return Vrai ;
```

Théorème 2.2. *Le problème de décider si un impliquant premier est essentiel est NP-complet.*

Démonstration. En utilisant le théorème 2.1, on sait que le problème considéré est dans NP. En effet, étant donné une formule CNF ϕ et un impliquant premier I de ϕ, alors I est essentiel si et seulement s'il existe un e-modèle $\mathcal{B}$ de ϕ tel que $B(\bigwedge_{l\in I} l) = 1$. Ainsi, il existe un certificat qui permet de vérifier en temps polynomial si un impliquant premier est essentiel. Démontrons maintenant que le problème considéré est NP-difficile. Afin de réaliser cela, on utilise le problème SAT, qui est comme mentionné précédemment NP-complet. Soit ϕ une formule CNF. On associe à chaque clause $c = l_1 \vee \cdots \vee l_k$ une nouvelle variable propositionnelle p_c et la conjonction de clauses binaires $\psi_c = (p_c \vee \overline{l_1}) \wedge \cdots \wedge (p_c \vee \overline{l_k})$. On définit ensuite $E(\phi)$ comme étant la formule CNF $\bigwedge_{c\in\phi} \psi_c$. Nous démontrerons maintenant que ϕ est satisfiable si et seulement si $I = \{p_c \mid c \in \phi\}$ est un impliquant premier essentiel de $E(\phi)$. Avant cela, il convient de noter qu'étant donné que chaque clause de $E(\phi)$ ne partage qu'un unique littéral avec I, I est forcément un impliquant premier de $E(\phi)$.

La partie « si ». Considérons que I est un impliquant premier essentiel de $E(\phi)$. Alors, il existe un e-modèle $\mathcal{B}$ de $E(\phi)$ tel que $\mathcal{B}(\bigwedge_{l\in I} l) = 1$. Supposons que $\mathcal{B}' = \mathcal{B}_{|Var(\phi)}$ n'est pas un modèle de ϕ. Alors, il existe une clause $c = l_1 \vee \cdots \vee l_k$ dans ϕ telle que $\mathcal{B}'(l_1) = \cdots = \mathcal{B}'(l_k) = 0$. Par conséquent, $\mathcal{B}_{\{p_c\mapsto 0\}}$ est un modèle de $E(\phi)$, ce qui signifie que $\mathcal{B}$ n'est pas un e-modèle car il est couvert par un autre impliquant premier différent de I. Donc, on obtient une contradiction.

La partie « seulement si ». Considérons que ϕ admet un modèle $\mathcal{B}$. Clairement, $\mathcal{B}' = \mathcal{B} \cup \{p_c \mapsto 1 \mid c \in \phi\}$ est un modèle de $E(\phi)$. De plus, pour toute clause $c \in \phi$, il existe un littéral $l \in c$ tel que $p_c \vee \bar{l} \in E(\phi)$ et $\mathcal{B}'(l) = 1$. Donc, pour toute clause $c \in \phi$, $\mathcal{B}'_{\{p_c\mapsto 0\}}$ ne satisfait pas $E(\phi)$. Par conséquent, I est un impliquant essentiel de $E(\phi)$. □

Soient ϕ une formule CNF et $\mathcal{B}$ un modèle de ϕ. On utilise $U(\mathcal{B}, \phi)$ pour noter l'ensemble de littéraux $\{l \in Lit(\phi) \mid \mathcal{B}(l) = 1 \texttt{ et } \exists c \in \phi,\ \mathcal{B}(c \setminus \{l\}) = 0\}$.

Théorème 2.3. *Étant donné une formule CNF ϕ, un modèle $\mathcal{B}$ de ϕ est un e-modèle si et seulement si $U(\mathcal{B}, \phi)$ est un impliquant de ϕ (a fortiori un impliquant premier essentiel).*

Démonstration.

La partie « si ». Considérons que $U(\mathcal{B}, \phi)$ est un impliquant de ϕ. En utilisant la définition de $U(\mathcal{B}, \phi)$, il est un impliquant premier de ϕ. De plus, on sait que pour tout $l \in U(\mathcal{B}, \phi)$, $\mathcal{B}_{\{l \mapsto 0\}}$ ne satisfait pas ϕ. On obtient ainsi que $U(\mathcal{B}, \phi)$ est un impliquant premier essentiel de ϕ et $\mathcal{B}$ est un e-modèle de ϕ.

La partie « seulement si ». Considérons que $\mathcal{B}$ est un e-modèle de ϕ. Il existe alors un unique impliquant premier I couvrant $\mathcal{B}$. Par ailleurs, en utilisant la définition de $U(\mathcal{B}, \phi)$, on obtient $U(\mathcal{B}, \phi) \subseteq I$. Supposons maintenant qu'il existe $l \in I$ tel que $l \notin U(\mathcal{B}, \phi)$. On obtient alors que $\mathcal{B}_{\{l \mapsto 0\}}$ satisfait ϕ, car il n'existe aucune clause vraie grâce uniquement à la vérité de l. Ainsi, il existe un impliquant premier de ϕ qui ne contient pas l et qui couvre $\mathcal{B}$. On a donc une contradiction avec le fait que $\mathcal{B}$ est un e-modèle. Par conséquent, $U(\mathcal{B}, \phi)$ est un impliquant premier essentiel de ϕ. □

Introduisons maintenant notre encodage pour l'énumération des impliquants premiers essentiels. Étant donné une formule CNF ϕ, nous utilisons $\mathcal{F}_{PEIP-E}(\phi)$ pour noter la formule suivante :

$$\phi \wedge \Big(\bigwedge_{l \in Lit(\phi)} (x_l \leftrightarrow (l \wedge \bigvee_{c \in \phi, l \in c} \ \bigwedge_{l' \in c \setminus \{l\}} \overline{l'}))) \Big) \wedge \Big(\bigwedge_{c \in \phi} \ \bigvee_{l \in c} x_l \Big)$$

où les variables propositionnelles de la forme x_l sont, comme dans le cas de $\mathcal{F}_{PEIP}(\phi)$, de nouvelles variables. Autrement dit, pour tout littéral $l \in Lit(\phi)$, une nouvelle variable x_l est associée à l.

Proposition 2.5 (Correction)**.** *Si $\mathcal{B}$ est un modèle de $\mathcal{F}_{PEIP-E}(\phi)$, alors l'ensemble $I = \{l \in Lit(\phi) \mid \mathcal{B}(x_l) = 1\}$ est un impliquant premier essentiel de ϕ.*

Démonstration. Sachant que $\mathcal{B}$ est un modèle de $\mathcal{F}_{PEIP-E}(\phi)$, $\mathcal{B}' = \mathcal{B}_{|Var(\phi)}$ est un modèle de ϕ, car ϕ est une sous-formule de $\mathcal{F}_{PEIP-E}(\phi)$ qui doit être satisfaite. Par ailleurs, pour tout littéral $l \in Lit(\phi)$, en utilisant le fait que $x_l \rightarrow l$ est une conséquence logique de $\mathcal{F}_{PEIP-E}(\phi)$, on a $|\{x_{l'} \mid l' \in Lit(\phi), \mathcal{B}(x_{l'}) = 1\} \cap \{x_l, x_{\overline{l}}\}| \leqslant 1$. Donc, I ne contient pas deux littéraux complémentaires. En utilisant la sous-formule $\bigwedge_{c \in \phi} \ \bigvee_{l \in c} x_l$, on sait que I est un impliquant de ϕ. De plus, en utilisant les sous-formules de la forme $x_l \leftrightarrow (l \wedge \bigvee_{c \in \phi, l \in c} \ \bigwedge_{l' \in c \setminus \{l\}} \overline{l'})$, on obtient $I = U(\phi, \mathcal{B}')$. Ainsi, en utilisant le théorème 2.3, I est un impliquant premier essentiel de ϕ. □

Proposition 2.6 (Complétude)**.** *Si I est un impliquant premier essentiel de ϕ, alors il existe un modèle $\mathcal{B}$ de $\mathcal{F}_{PEIP-E}(\phi)$ où $I = \{l \in Lit(\phi) \mid \mathcal{B}(x_l) = 1\}$.*

Démonstration. En utilisant le fait que I est un impliquant premier essentiel, on sait qu'il existe un e-modèle $\mathcal{B}'$ de ϕ tel que $\mathcal{B}'(\bigwedge_{l \in I} l) = 1$. Donc, en utilisant le théorème 2.3, on a $I = U(\phi, \mathcal{B}')$. Étant donné que I est un impliquant de ϕ, $\mathcal{B}'' = \{x_l \mapsto 1 \mid l \in I\} \cup \{x_l \mapsto 0 \mid l \in Lit(\phi) \setminus I\}$ est un modèle de $\bigwedge_{c \in \phi} \ \bigvee_{l \in c} x_l$. De plus, on sait que $\mathcal{B} = \mathcal{B}' \cup \mathcal{B}''$ est un modèle de $x_l \leftrightarrow (l \wedge \bigvee_{c \in \phi, l \in c} \ \bigwedge_{l' \in c \setminus \{l\}} \overline{l'})$ car $I = U(\phi, \mathcal{B}')$. Par conséquent, $\mathcal{B}$ est un modèle de $\mathcal{F}_{PEIP-E}(\phi)$ où $I = \{l \in Lit(\phi) \mid \mathcal{B}(x_l) = 1\}$. □

Algorithm 3: Un algorithme fondé sur une formulation SAT pour PEIP-E

Data: une formule CNF ϕ.
Result: l'ensemble des impliquants premiers essentiels de ϕ.

$L \leftarrow \emptyset$;
$\psi \leftarrow \mathcal{F}_{PEIP-E}(\phi)$;
while $SAT(\psi)$ **do**
  /* $\mathcal{B}$ est un modèle ψ */
  $U \leftarrow \{l \in Lit(\phi) \mid \mathcal{B}(x_l) = 1\}$;
  $L \leftarrow L \cup \{U\}$;
  $\psi \leftarrow \psi \wedge \bigvee_{l \in U} \overline{l}$
return L;

Comme mentionné précédemment, la formulation que nous venons de présenter n'est pas bijective. Cela vient principalement du fait qu'un impliquant premier essentiel peut couvrir deux e-modèles distincts. Néanmoins, notre formulation devient bijective si le problème considéré est celui consistant à énumérer les e-modèles. Considérons, par exemple, la formule CNF $\phi = p \wedge (p \vee q)$. Cette formule admet un unique impliquant premier $I = \{p\}$, qui est forcément essentiel, et les modèles $\{p \mapsto 1, q \mapsto 0\}$ et $\{p \mapsto 1, q \mapsto 1\}$ sont deux e-modèles distincts couverts par le même impliquant premier essentiel I.

Le fait que notre formulation ne soit pas bijective ne signifie pas que son utilisation implique une redondance dans la génération des solutions. En effet, l'algorithme 3 décrit une méthode simple permettant d'éviter la génération multiple d'un même impliquant premier essentiel. Dans cet algorithme, à chaque modèle $\mathcal{B}$ trouvé, on ajoute simplement la clause $\bigvee_{l \in U(\phi,\mathcal{B})} \overline{l}$ à la place de la négation de tout le modèle $\bigvee_{l \in \{l \in Lit(\phi) | \mathcal{B}(l)=1\}} \overline{l}$ afin d'éviter dans les prochaines itérations l'impliquant premier essentiel trouvé.

Même s'il est possible d'adapter notre formulation non bijective $\mathcal{F}_{PEIP-E}(\phi)$ à de nombreuses variantes de PEIP-E, il existe divers types de problèmes où l'adaptation peut s'avérer problématique, tels que notamment celui des problèmes ayant trait au nombre de solutions.

2.5 Conclusion

Dans ce chapitre, nous avons décrit la logique propositionnelle classique en pointant particulièrement le problème SAT et l'énumération des modèles. Nous avons également présenté notre encodage pour les contraintes de cardinalité proposé dans [JSS13a, JSS14] en raison de leur utilité dans la modélisation via SAT. En outre, à travers des résultats de modélisation issus de nos travaux dans [JMSS14, Sal18], nous avons introduit deux approches dans la modélisation reposant sur SAT. D'abord une approche fondée sur les formulations bijectives, où l'ensemble des modèles est en bijection avec l'ensemble des solutions du problème considéré. L'intérêt de cette approche réside dans le fait que la formulation en SAT peut sans difficulté être adaptée à de multiples variantes du problème d'origine. Nous avons ensuite introduit une approche fondée sur les formulations non bijectives, où une solution du problème traité peut être associée à plusieurs modèles. Nous avons notamment montré que, même en l'absence d'une bijection entre les solutions

et les modèles, il est possible dans certains cas d'éviter la redondance dans le processus d'énumération. Cependant, il n'est pas facile d'adapter des formulations non bijectives à certaines variantes des problèmes de départ.

Chapitre 3

Fouille de données via SAT

Une partie importante de nos travaux concerne l'utilisation d'approches fondées sur SAT pour résoudre différents problèmes en fouille de données. Dans ce cadre, nous aborderons ici certaines de nos contributions dans le domaine de l'extraction de motifs en employant des formulations en SAT. Les résultats décrits dans ce chapitre sont majoritairement issus de nos travaux dans [CJSS12, JSS13b, JSS13c, JSS15, BJSY16, BJSS17b, JSS17, BJSS18]. Une partie de ces travaux a été réalisée dans le cadre du projet ANR DAG.

3.1 Approches déclaratives et fouille de données

L'utilisation d'approches déclaratives en fouille de données a initialement été proposée dans [DGN08] pour la réalisation de différentes tâches. Plus précisément, les auteurs montrent dans le précédent article que la programmation par contraintes est un outil approprié à plusieurs égards pour l'extraction de plusieurs types de motifs ensemblistes. Une des motivations principales de l'utilisation de ce cadre réside dans le fait qu'il constitue un modèle de représentation flexible et générique. En effet, de nouvelles contraintes nécessitent souvent de nouvelles implémentations pour les approches spécialisées en fouille de données, ce qui peut souvent être intégré de manière relativement simple dans des cadres déclaratifs. En outre, l'évolution continuelle en matière d'efficacité des outils dédiés à la résolution des problèmes pouvant être utilisés pour la modélisation, comme ASP (*Answer Set Programming*), CSP (*Constraint Satisfaction Problem*) et SAT, est un argument fort en faveur de l'utilisation d'approches reposant sur ces problèmes. Ainsi, à partir de ce travail précurseur, une nouvelle ligne de recherche s'est imposée au sein de la communauté de fouille de données. Nous assistons véritablement depuis plusieurs années à de nombreuses contributions dans la réalisation de différentes tâches en fouille de données par l'utilisation d'approches déclaratives. Nous pouvons, par exemple, mentionner l'utilisation de CSP pour l'extraction de motifs définis via plusieurs contraintes locales [KBC10]. Parmi les nombreux autres travaux faisant appel à la modélisation en CSP pour la fouille de données, nous pouvons aussi citer ceux dans [GND11, Gun15, UBLC15, UBC+17, BLM18]. Un autre exemple de l'utilisation d'une approche déclarative en fouille est l'emploi d'ASP pour l'extraction de motifs

séquentiels [GGQ+16]. Par ailleurs, il convient de signaler la proposition de langages édifiés sur l'expression de contraintes pour la résolution de problèmes en fouille de données, comme en particulier MiningZinc [GDN+17].

3.2 Motifs ensemblistes fréquents

3.2.1 Enoncés des problèmes

Considérons l'ensemble de tickets de caisse décrits dans la figure 3.1. Nous pouvons remarquer de prime abord qu'il y a des produits apparaissant dans tous les tickets, comme « Fromage ». De la même manière, il y a des produits que l'on trouve dans la majorité des tickets de caisse fournis, comme « Pain » et « Tomates » qui apparaissent respectivement dans cinq et quatre tickets sur les six donnés. Ce type d'informations peut être utilisé pour nous renseigner sur les produits les plus populaires, ce qui est clairement utile et pertinent dans un contexte commercial à l'instar de bien d'autres contextes. La tâche en fouille de données consistant à extraire les motifs ensemblistes fréquents fournit un cadre naturel et bien défini pour la capture d'informations de la même nature que celles que nous venons de décrire.

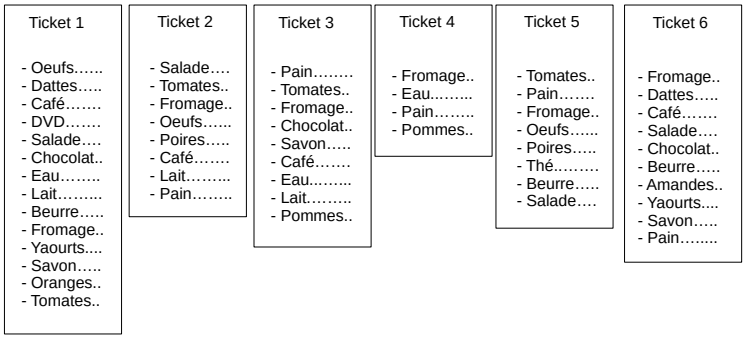


FIGURE 3.1 – Un ensemble de tickets de caisse

Le problème de l'énumération des motifs ensemblistes fréquents a été proposé pour la première fois dans l'article [AIS93]. Il a connu par la suite un succès important dans l'analyse de données reflété par le grand nombre de travaux qui ont suivi la proposition initiale.

L'extraction des motifs ensemblistes est réalisée sur une structure que l'on nomme *base de données transactionnelles*. Plus précisément, soit $\mathcal{I}$ un ensemble fini et non vide d'éléments nommés *items*. Une *transaction* sur $\mathcal{I}$ est un couple (id, I), où id correspond à son identifiant et I à un sous-ensemble de $\mathcal{I}$. Une *base de données transactionnelles* sur $\mathcal{I}$ est un ensemble fini et non vide de transactions sur $\mathcal{I}$ tel que chaque identifiant n'est associé qu'à une unique transaction, autrement dit, il apparait une seule fois. Par exemple, la base de données transactionnelles correspondant aux tickets de caisse dans la figure 3.1 est décrite dans la table 3.1.

Un *motif ensembliste*, appelé également *itemset*, est simplement défini comme un ensemble fini et non vide d'items. On dit qu'une transaction (id, I) *supporte* un motif

id	ensemble d'items
Ticket 1	$a, b, c, d, e, f, g, h, i, j, k, l, m, n$
Ticket 2	a, c, e, h, j, n, o, p
Ticket 3	$c, f, g, h, j, l, n, p, q$
Ticket 4	g, j, p, q
Ticket 5	a, e, i, j, n, o, p, s
Ticket 6	$b, c, e, f, i, j, k, l, p, r$

Oeufs (a)	Dattes (b)	Café (c)	DVD (d)	Salade (e)
Chocolat (f)	Eau (g)	Lait (h)	Beurre (i)	Fromage (j)
Yaourts (k)	Savon (l)	Oranges (m)	Tomates (n)	Poires (o)
Pain (p)	Pommes (q)	Amandes (r)	Thé (s)	

TABLE 3.1 – Une base de données transactionnelles

ensembliste E si l'on a $E \subseteq I$. Dans ce contexte, étant donné une base de données transactionnelles $\mathcal{D}$ et un motif ensembliste E, on définit la *couverture* de E dans $\mathcal{D}$, notée $\mathcal{C}(E, \mathcal{D})$, comme l'ensemble des transactions de $\mathcal{D}$ supportant $E : \mathcal{C}(E, \mathcal{D}) = \{(id, I) \in \mathcal{D} \mid E \subseteq I\}$. Le *support* de E dans $\mathcal{D}$, noté $\mathcal{S}(E, \mathcal{D})$, correspond à la taille de la couverture : $\mathcal{S}(E, \mathcal{D}) = |\mathcal{C}(E, \mathcal{D})|$. En utilisant ces notions, introduisons à présent le problème de l'énumération des motifs ensemblistes fréquents.

Problème de l'énumération des motifs ensemblistes fréquents (PEMEF).
- Entrée : une base de données transactionnelles $\mathcal{D}$ et un entier naturel non nul α jouant le rôle de quorum sur le support.
- Sortie : l'ensemble des motifs $\mathcal{MEF}(\mathcal{D}, \alpha) = \{E \mid \mathcal{S}(E, \mathcal{D}) \geqslant \alpha\}$.

Nous nous référerons à chacune des instances de PEMEF par le couple correspondant à l'entrée $(\mathcal{D}, \alpha)$.

En d'autres termes, le problème de l'énumération des motifs ensemblistes fréquents consiste à extraire tous les motifs ensemblistes dont les supports ne sont pas inférieurs au quorum fourni en entrée. Par exemple, dans la base de données transactionnelles décrite dans la table 3.1, si l'on considère un quorum valant 5, le motif $\{j, p\}$, qui correspond aux deux produits « Fromage » et « Pain », est fréquent car il apparait dans les transactions associées aux cinq derniers tickets de caisse. Il convient de noter qu'en utilisant le fait que le motif $\{j, p\}$ est fréquent, on est certain que ses deux sous-ensembles non vides $\{j\}$ et $\{p\}$ le sont également, sans même que l'on ait besoin d'explorer la base de données. Cela traduit un principe nommé anti-monotonie.

Proposition 3.1 (Anti-monotonie)**.** *Soient $\mathcal{D}$ une base de données transactionnelles et α un entier naturel non nul. Pour tout $E \in \mathcal{MEF}(\mathcal{D}, \alpha)$ et pour tout $E' \subset E$ avec $E' \neq \emptyset$, on a $E' \in \mathcal{MEF}(\mathcal{D}, \alpha)$.*

Démonstration. Cette proposition est une conséquence directe du fait que la couverture de E est incluse dans celle de $E' : \mathcal{C}(E, \mathcal{D}) \subseteq \mathcal{C}(E', \mathcal{D})$, pour tout $E' \subseteq E$ avec $E' \neq \emptyset$. □

Il résulte de la précédente proposition qu'il est possible d'obtenir tous les motifs fréquents en énumérant uniquement une partie de ces derniers. Des représentations condensées autour du principe de l'anti-monotonie ont ainsi été proposées. Dans ce qui suit, nous définissons les deux plus connues [Bay98, PBTL99].

Définition 3.1 (Motif ensembliste fermé)**.** *Soit* $(\mathcal{D}, \alpha)$ *une instance de PEMEF. Un motif fréquent* $E \in \mathcal{MEF}(\mathcal{D}, \alpha)$ *est dit* fermé *si, pour tout* E' *avec* $E \subset E'$, $\mathcal{C}(E', \mathcal{D})$ *est un sous-ensemble propre de* $\mathcal{C}(E, \mathcal{D})$*, à savoir* $\mathcal{C}(E', \mathcal{D}) \subset \mathcal{C}(E, \mathcal{D})$*.*

Définition 3.2 (Motif ensembliste maximal)**.** *Soit* $(\mathcal{D}, \alpha)$ *une instance de PEMEF. Un motif fréquent* $E \in \mathcal{MEF}(\mathcal{D}, \alpha)$ *est dit* maximal *si, pour tout* E' *avec* $E \subset E'$, $\mathcal{S}(E', \mathcal{D}) < \alpha$*.*

En d'autres termes, un motif fréquent est fermé s'il n'est inclus dans aucun autre motif possédant la même couverture, et il est maximal s'il n'est inclus dans aucun autre motif fréquent. Nous utiliserons $\mathcal{CMEF}(\mathcal{D}, \alpha)$ et $\mathcal{MMEF}(\mathcal{D}, \alpha)$ pour désigner respectivement l'ensemble des motifs fermés et celui des motifs maximaux dans $\mathcal{MEF}(\mathcal{D}, \alpha)$.

Il est clair que l'ensemble $\mathcal{MEF}(\mathcal{D}, \alpha)$ peut simplement être construit grâce à chacun des ensembles $\mathcal{CMEF}(\mathcal{D}, \alpha)$ et $\mathcal{MMEF}(\mathcal{D}, \alpha)$. Effectivement, en utilisant le principe d'anti-monotonie, on obtient $\mathcal{MEF}(\mathcal{D}, \alpha) = \bigcup_{E \in \mathcal{CMEF}(\mathcal{D}, \alpha)} \{E' \mid E' \subseteq E \text{ et } E' \neq \emptyset\} = \bigcup_{E \in \mathcal{MMEF}(\mathcal{D}, \alpha)} \{E' \mid E' \subseteq E \text{ et } E' \neq \emptyset\}$.

Remarquons que l'ensemble des motifs maximaux est un sous-ensemble de celui des fermés. En effet, tout motif maximal est fermé car un motif maximal n'est inclus dans aucun autre motif fréquent et, à plus forte raison, dans aucun autre motif fréquent ayant la même couverture, ce qui signifie qu'il est fermé. Ainsi, une question s'impose : l'extraction des motifs maximaux est-elle toujours plus intéressante que l'extraction des motifs fermés ? Si l'on tient compte uniquement du nombre de motifs, la réponse est certainement positive ; par contre, elle devient négative dans tout contexte où sont nécessaires les couvertures ou les supports des motifs fréquents, ce qui est en particulier le cas dans la génération des règles d'association. En effet, la couverture de tout sous-ensemble d'un motif fermé est égale à la couverture de ce dernier, et ainsi, avec les couvertures des motifs fermés, on a celles de tous les motifs fréquents, ce qui n'est pas vrai dans le cas des motifs maximaux.

3.2.2 Formulations en SAT

Nous présentons ici une formulation en SAT du problème de l'énumération des motifs ensemblistes fréquents ainsi que d'autres variantes de ce dernier, comme l'énumération des motifs fermés et ceux maximaux.

Considérons une instance de PEMEF $(\mathcal{D}, \alpha)$, où $\mathcal{I}$ est l'ensemble d'items apparaissant dans $\mathcal{D}$ et n le nombre de transactions de cette dernière. Pour la définition de notre formulation en SAT, nous associons à chaque item a une variable propositionnelle distincte notée p_a. Nous associons également à chaque transaction t dans $\mathcal{D}$ une variable propositionnelle distincte notée q_t. Intuitivement, les variables associées aux items serviront à représenter les motifs ensemblistes, et celles associées aux transactions serviront à capturer les couvertures des motifs.

Notre première formule est utilisée afin que les variables associées aux transactions représentent la couverture du motif ensembliste courant. Plus précisément, elle permet d'exprimer la propriété selon laquelle, pour toute transaction t, q_t possède 1 comme valeur de vérité si et seulement si le motif courant est supporté par t :

$$\bigwedge_{t=(id,I)\in\mathcal{D}} (\neg q_t \leftrightarrow \bigvee_{a\in\mathcal{I}\setminus I} p_a) \tag{3.1}$$

Concernant la formule suivante, elle signifie simplement qu'un motif ensembliste ne peut être vide :

$$\bigvee_{a\in\mathcal{I}} p_a \tag{3.2}$$

La contrainte de cardinalité suivante impose le respect du quorum sur le support :

$$\sum_{t\in\mathcal{D}} q_t \geqslant \alpha \tag{3.3}$$

Rappelons que notre encodage en SAT des contraintes de cardinalité est décrit dans la section 2.3.

Nous noterons $\mathcal{F}_{PEMEF}(\mathcal{D},\alpha)$ la formulation correspondant à la conjonction des trois précédentes formules : (3.1) $\wedge$ (3.2) $\wedge$ (3.3).

Il est clair que la formulation en SAT $\mathcal{F}_{PEMEF}(\mathcal{D},\alpha)$ du problème PEMEF est bijective. Cela vient principalement de l'utilisation du connecteur logique de l'équivalence dans la formule (3.3). Dans ce contexte, il est à noter que nous pouvons simplifier cette formulation en la rendant non bijective, sans pour autant perdre la correction et la complétude. Cela est en effet possible par le remplacement de (3.3) par la formule suivante :

$$\bigwedge_{t=(id,I)\in\mathcal{D}} (\neg q_t \leftarrow \bigvee_{a\in\mathcal{I}\setminus I} p_a) \tag{3.4}$$

Afin d'éviter la redondance dans l'énumération des solutions en utilisant la précédente formule, il suffit d'ajouter pour chaque motif trouvé sa négation au lieu de la négation du modèle lui correspondant. Plus précisément, si E est le motif trouvé, alors nous ajoutons la clause $\bigvee_{a\in E} \neg p_a \vee \bigvee_{b\in\mathcal{I}\setminus E} p_b$ pour éviter ce motif dans les prochaines itérations. Cela fonctionne bien entendu dans le cas de la formulation bijective.

La raison principale derrière l'utilisation du connecteur de l'équivalence à la place de celui de l'implication vient du fait que la valeur exacte de la couverture de chaque motif est nécessaire pour des représentations condensées comme les motifs fermés et les motifs maximaux. Considérons d'abord le cas des motifs fermés fréquents. Une formulation en SAT permettant de restreindre l'énumération à ces motifs est simplement obtenue en ajoutant par conjonction la formule suivante à $\mathcal{F}_{PEMEF}(\mathcal{D},\alpha)$:

$$\bigwedge_{a\in\mathcal{I}} ((\bigwedge_{t=(id,I)\in\mathcal{D}} (q_t \rightarrow a \in I)) \rightarrow p_a) \tag{3.5}$$

où les expressions de la forme $a \in I$ correspondent à des constantes : $\top$ si a appartient à I, $\bot$ sinon. Cette formule permet d'exprimer que, pour chaque item a, si la couverture

d'un motif E associé à un modèle est égale à celle de $E \cup \{a\}$, alors a est dans E, ce qui est exprimé par l'affectation de la valeur de vérité 1 à p_a. Ainsi, de l'obligation d'avoir la valeur exacte de la couverture, l'utilisation de (3.4) à la place de (3.3) ne permet pas ici de restreindre l'énumération aux motifs fermés.

De manière similaire, pour l'énumération des motifs maximaux fréquents, il suffit d'ajouter par conjonction la formule suivante à $\mathcal{F}_{PEMEF}(\mathcal{D}, \alpha)$:

$$\bigwedge_{a \in \mathcal{I}} \Big(\sum_{t=(id,I) \in \mathcal{D}} (q_t \wedge a \in I) \geqslant \alpha \rightarrow p_a \Big) \tag{3.6}$$

Cette formule permet d'exprimer que si l'ajout d'un item a au motif correspondant à un modèle ne réduit pas le support à une valeur inférieure au quorum α, alors a est dans ce motif.

Il est important de noter que des formulations en SAT pour plusieurs autres variantes de PEMEF peuvent être obtenues par de simples modifications de $\mathcal{F}_{PEMEF}(\mathcal{D}, \alpha)$. Par exemple, si nous voulons restreindre l'énumération aux motifs fermés ayant des tailles supérieures ou égales à β, il suffit d'ajouter la contrainte de cardinalité $\sum_{a \in \mathcal{I}} p_a \geqslant \beta$ à la formulation permettant l'énumération des motifs fermés. De même, si nous voulons restreindre l'énumération à des motifs contenant au moins un item dans un ensemble S donné, nous ajoutons simplement la clause $\bigvee_{a \in S} p_a$. Cela montre particulièrement la modularité et la flexibilité de l'approche fondée sur SAT.

3.3 Règles d'association

3.3.1 Enoncés des problèmes

Considérons de nouveau la base de données transactionnelles décrite dans la table 3.1 et fixons le quorum à 3. Dans ce cas, l'ensemble $E = \{\text{Fromage, Eau}\}$ correspond à un motif fréquent car il apparait dans trois tickets de caisse distincts (Ticket 1, Ticket 3 et Ticket 4). Ce motif traduit-il le fait que l'achat de fromage implique qu'il est fort vraisemblable que l'achat d'eau se réalise ? En fait, si l'on observe les tickets de caisse fournis, le fromage apparait dans tous les tickets, et c'est l'eau qui n'apparait que dans trois. Par conséquent, il y a autant de tickets de caisse où le fromage apparait avec l'eau que sans. Les motifs ensemblistes ne permettent donc pas de capturer le type de relations exprimé dans la précédente question, et c'est dans ce but que le concept de règle d'association a été introduit au même moment que les motifs ensemblistes fréquents [AIS93].

Définition 3.3 (Règle d'association)**.** *Une règle d'association est une structure de la forme $X \Rightarrow Y$, où X et Y sont des ensembles d'items vérifiant les deux propriétés suivantes : (i) $X \neq \emptyset$ et $Y \neq \emptyset$ et (ii) $X \cap Y = \emptyset$. Les parties X et Y sont respectivement appelées* antécédent *et* conséquent*.*

Afin d'évaluer la qualité d'une règle d'association $X \Rightarrow Y$ dans une base de données transactionnelles $\mathcal{D}$, on utilise en général deux mesures :

1. le *support*, noté $\mathcal{S}(X \Rightarrow Y, \mathcal{D})$, qui correspond à la valeur $\mathcal{S}(X \cup Y, \mathcal{D})$;
2. la *confiance*, notée $\mathcal{C}onf(X \Rightarrow Y, \mathcal{D})$, qui correspond à la valeur $\frac{\mathcal{S}(X \cup Y, \mathcal{D})}{\mathcal{S}(X, \mathcal{D})}$.

Autrement dit, le support d'une règle est le nombre de transactions la vérifiant, à savoir celles contenant à la fois l'antécédent et le conséquent, et la confiance est une estimation de la probabilité de l'événement correspondant au conséquent sachant l'antécédent.

Par exemple, {Fromage} ⇒ {Pain} est une règle d'association dans la base de données décrite dans la table 3.1 ayant comme support 4 et comme confiance $\frac{2}{3}$.

Tout comme dans le cas de l'énumération des motifs ensemblistes fréquents, le problème de l'énumération des règles d'association est défini en fixant des quorums sur le support et la confiance.

Problème de l'énumération des règles d'association (PERA).
- Entrée : une base de données transactionnelles $\mathcal{D}$, un entier naturel non nul α jouant le rôle de quorum sur le support, et un autre entier naturel non nul β (un pourcentage) jouant le rôle de quorum sur la confiance.
- Sortie : l'ensemble des règles d'association $\mathcal{RA}(\mathcal{D}, \alpha, \beta) = \{X \Rightarrow Y \mid \mathcal{S}(X \Rightarrow Y, \mathcal{D}) \geqslant \alpha$ *et* $\mathcal{C}onf(X \Rightarrow Y, \mathcal{D}) \geqslant \frac{\beta}{100}\}$.

Nous nous référerons à chaque instance de PERA par un triplet $(\mathcal{D}, \alpha, \beta)$.

Nous avons précédemment vu qu'il existe des représentations condensées permettant de réduire la taille de la sortie dans l'énumération des motifs ensemblistes fréquents. De la même manière, nous décrivons dans ce qui suit une représentation condensée pour les règles d'association proposée dans [Sza06]. Nous verrons que cette représentation est fortement liée à celle de motif ensembliste fermé exposée dans la définition 3.1.

Définition 3.4 (Règle d'association fermée)**.** *Soit* $(\mathcal{D}, \alpha, \beta)$ *une instance de PERA. Une règle d'association* $X \Rightarrow Y \in \mathcal{RA}(\mathcal{D}, \alpha, \beta)$ *est dite* fermée *si l'ensemble* $X \cup Y$ *est un motif fermé dans* $\mathcal{MEF}(\mathcal{D}, \alpha)$.

Nous utiliserons $\mathcal{CRA}(\mathcal{D}, \alpha, \beta)$ pour désigner l'ensemble des règles d'association fermées pour l'instance $(\mathcal{D}, \alpha, \beta)$.

Nous montrons dans la prochaine proposition qu'il est possible de reconstruire l'ensemble $\mathcal{RA}(\mathcal{D}, \alpha, \beta)$ de manière simple à partir de $\mathcal{CRA}(\mathcal{D}, \alpha, \beta)$.

Proposition 3.2. *Soient* $(\mathcal{D}, \alpha, \beta)$ *une instance de PERA, et* X *et* Y *deux ensembles d'items non vides. Alors,* $X \Rightarrow Y$ *est une règle d'association dans* $\mathcal{RA}(\mathcal{D}, \alpha, \beta)$ *si et seulement s'il existe un ensemble d'items* Z *tel que* $X \Rightarrow (Y \cup Z) \in \mathcal{CRA}(\mathcal{D}, \alpha, \beta)$.

Démonstration. Il est clair que pour tout ensemble d'items Z, si $X \Rightarrow (Y \cup Z) \in \mathcal{CRA}(\mathcal{D}, \alpha, \beta)$ alors on obtient $X \Rightarrow Y \in \mathcal{RA}(\mathcal{D}, \alpha, \beta)$. En effet, en utilisant la propriété d'anti-monotonie, on a $\mathcal{S}(X \cup Y \cup Z, \mathcal{D}) \leqslant \mathcal{S}(X \cup Y, \mathcal{D})$, ce qui permet de déduire que $\mathcal{S}(X \Rightarrow (Y \cup Z), \mathcal{D}) \leqslant \mathcal{S}(X \Rightarrow Y, \mathcal{D})$ et $\mathcal{C}onf(X \Rightarrow (Y \cup Z), \mathcal{D}) \leqslant \mathcal{C}onf(X \Rightarrow Y, \mathcal{D})$. Démontrons maintenant la partie « *seulement si* ». Soit $X \Rightarrow Y \in \mathcal{RA}(\mathcal{D}, \alpha, \beta)$. En utilisant la définition des motifs ensemblistes fermés, on sait qu'il existe un ensemble d'items Z tel que $X \cup Y \cup Z$ est un motif fermé dans la base $\mathcal{D}$ avec $\mathcal{S}(X \cup Y \cup Z, \mathcal{D}) = \mathcal{S}(X \cup Y, \mathcal{D})$. Par conséquent, on a $\mathcal{S}(X \to (Y \cup Z), \mathcal{D}) = \mathcal{S}(X \Rightarrow Y, \mathcal{D})$ et $\mathcal{C}onf(X \Rightarrow (Y \cup Z), \mathcal{D}) = \mathcal{C}onf(X \Rightarrow Y, \mathcal{D})$. Nous déduisons ainsi que $X \Rightarrow (Y \cup Z) \in \mathcal{CRA}(\mathcal{D}, \alpha, \beta)$. □

Il est important de noter que l'on ne peut pas obtenir la même propriété en considérant uniquement les règles d'association construites autour des motifs maximaux. Pour s'en convaincre, considérons l'instance simple $(\mathcal{D}, 1, 1)$ où $\mathcal{D}$ est la base suivante :

id	ensemble d'items
1	a, b, c
2	a, b

Avec un quorum sur le support égal à 1, il n'y a qu'un seul motif maximal $\{a, b, c\}$. Dans cet exemple, on a $\{a\} \Rightarrow \{b, c\} \notin \mathcal{RA}(\mathcal{D}, 1, 1)$ alors que l'on a $\{a\} \Rightarrow \{b\} \in \mathcal{RA}(\mathcal{D}, 1, 1)$.

Parmi les autres représentations condensées, il nous parait important de mentionner les règles minimales non redondantes [Kry98, BPT+00]. Ces dernières sont définies comme les règles possédant les plus petits antécédents et les plus larges conséquents au sens de l'inclusion, tout en préservant le support et la confiance.

Définition 3.5 (Règle minimale non redondante)**.** *Soit $(\mathcal{D}, \alpha, \beta)$ une instance de PERA. Une règle d'association $X \Rightarrow Y \in \mathcal{RA}(\mathcal{D}, \alpha, \beta)$ est dite* minimale non redondante *s'il n'existe pas de règle d'association $X' \Rightarrow Y' \in \mathcal{RA}(\mathcal{D}, \alpha, \beta)$ distincte de $X \Rightarrow Y$ avec le même support et la même confiance que cette dernière telle que $X' \subseteq X$ et $Y \subseteq Y'$.*

Comme constaté dans [BPT+00], toute règle minimale non redondante (RMNR) est également une règle fermée. Cela vient en effet du fait que le conséquent de chaque RMNR est maximal au sens de l'inclusion. En ce qui concerne les antécédents des RMNR, les auteurs de [BPT+00] fournissent une caractérisation simple de ces derniers à partir d'un concept appelé générateur minimal.

Définition 3.6 (Générateur minimal)**.** *Étant donné un motif ensembliste fermé E dans une base $\mathcal{D}$, un sous-ensemble $E' \subseteq E$ est un* générateur minimal *de E si (i) $\mathcal{S}(E', \mathcal{D}) = \mathcal{S}(E, \mathcal{D})$ et (ii) il n'existe pas de E'' sous-ensemble propre de E' tel que $\mathcal{S}(E'', \mathcal{D}) = \mathcal{S}(E, \mathcal{D})$.*

Pour être plus précis, tout antécédent d'une règle minimale non redondante est un motif de taille 1 ou un générateur minimal non vide d'un motif fermé. En effet, un générateur minimal peut être vide dans la situation où un motif apparait dans toutes les transactions. En conséquence, pour éviter les antécédents vides, on a inclus le cas des singletons dans la caractérisation des règles minimales non redondantes.

Nous utiliserons $\mathcal{MRA}(\mathcal{D}, \alpha, \beta)$ pour désigner l'ensemble des règles minimales non redondantes correspondant à l'instance $(\mathcal{D}, \alpha, \beta)$.

Proposition 3.3. *Soient $(\mathcal{D}, \alpha, \beta)$ une instance de PERA, et X et Y deux ensembles d'items non vides. Alors, $X \Rightarrow Y \in \mathcal{CRA}(\mathcal{D}, \alpha, \beta)$ si et seulement s'il existe $Z \subseteq X$ tel que $(X \setminus Z) \Rightarrow (Y \cup Z) \in \mathcal{MRA}(\mathcal{D}, \alpha, \beta)$.*

Démonstration. Comme évoqué précédemment, toute règle minimale non redondante est également une règle fermée, ce qui démontre la partie « *si* ». Considérons maintenant le cas de la partie « *seulement si* ». Soit $X \Rightarrow Y \in \mathcal{CRA}(\mathcal{D}, \alpha, \beta)$. Soit $Z \subseteq X$ tel que, pour tout $Z' \subseteq X$ avec $Z \subset Z'$, (1) $\mathcal{C}onf((X \setminus Z) \Rightarrow (Y \cup Z), \mathcal{D}) > \mathcal{C}onf((X \setminus Z') \Rightarrow (Y \cup Z'), \mathcal{D})$. En utilisant le fait que $X \cup Y \in \mathcal{CMEF}(\mathcal{D}, \alpha)$, on sait que pour tout item

$a \notin X \cup Y$, $\mathcal{S}(X \cup Y \cup \{a\}, \mathcal{D}) < \mathcal{S}(X \cup Y, \mathcal{D})$. Ainsi, (2) pour tout item $a \notin X \cup Y$, on a aussi $\mathcal{S}((X \setminus Z) \Rightarrow (Y \cup Z), \mathcal{D}) > \mathcal{S}((X \setminus Z) \Rightarrow (Y \cup Z \cup \{a\}), \mathcal{D})$. Par conséquent, en utilisant les propriétés (1) et (2), on obtient $(X \setminus Z) \Rightarrow (Y \cup Z) \in \mathcal{MRA}(\mathcal{D}, \alpha, \beta)$. □

La précédente proposition décrit la manière dont nous pouvons construire l'ensemble de toutes les règles fermées en utilisant les règles minimales non redondantes. De ce fait, en utilisant la proposition 3.2, l'ensemble de toutes les règles d'association valides peut également être construit en utilisant uniquement les règles minimales non redondantes.

Nous présentons maintenant une tâche en fouille de données liée aux règles d'association, appelée fouille des *règles indirectes* [TKS00, Kaz09]. Ces dernières permettent notamment la découverte des items apparaissant rarement ensemble mais qui sont très dépendants en présence d'un ensemble d'items appelé médiateur. Le choix ici de cette tâche vise principalement à montrer la modularité et la flexibilité de notre approche fondée sur la modélisation en SAT.

Définition 3.7 (Règle indirecte)**.** *Étant donné deux quorums sur le support α (comme seuil maximal) et α' (comme seuil minimal), avec $\alpha' > \alpha$, et un quorum sur la confiance*[1] *β sous forme de pourcentage, deux items a et b sont* indirectement associés *si les propriétés suivantes sont satisfaites :*

- $\mathcal{S}(\{a, b\}) \leqslant \alpha$,
- *il existe un ensemble non vide d'items M, appelé* médiateur*, tel que :*
  (a) $\mathcal{S}(\{a\} \cup M) \geqslant \alpha'$ *et* $\mathcal{S}(\{b\} \cup M) \geqslant \alpha'$,
  (b) $\mathcal{C}onf(\{a\} \Rightarrow M) \geqslant \frac{\beta}{100}$ *et* $\mathcal{C}onf(\{b\} \Rightarrow M) \geqslant \frac{\beta}{100}$.

La première propriété dans la définition précédente permet d'exprimer que les deux items considérés apparaissent rarement dans les mêmes transactions ; quant à la deuxième, elle exprime que ces deux items sont fortement liés à un même ensemble d'items. Une des applications intéressantes des règles indirectes réside dans le cadre de la recommandation via le Web [Kaz09].

3.3.2 Formulations en SAT

La formulation en SAT du problème de l'énumération des règles d'association est relativement proche de celle pour PEMEF. En réalité, la principale différence réside dans le fait que chaque item sera représenté par deux variables propositionnelles au lieu d'une seule, car une règle d'association implique deux ensembles d'items. Un des principaux objectifs visés par la présentation de cette formulation est notamment d'illustrer la manière dont nous pouvons encoder la recherche de plusieurs motifs ensemblistes entretenant différents types de relations.

Soit $(\mathcal{D}, \alpha, \beta)$ une instance de PERA, où $\mathcal{I}$ est l'ensemble d'items apparaissant dans $\mathcal{D}$. Pour la définition de notre formulation en SAT, nous associons à chaque item a deux variables propositionnelles distinctes notées x_a et y_a. Nous associons aussi à chaque transaction t dans $\mathcal{D}$ deux variables propositionnelles distinctes notées q_t et r_t. Les variables de la forme q_t serviront à représenter la couverture de l'antécédent, et celles de la forme r_t la couverture de la règle d'association.

1. Plusieurs autres mesures de dépendance ont été proposées dans la littérature (p. ex. [TKS02]).

Notre première formule énonce que l'antécédent et le conséquent ne partagent aucun item :

$$\bigwedge_{a \in \mathcal{I}} (\neg x_a \vee \neg y_a) \tag{3.7}$$

Quant à la formule suivante, elle permet d'exprimer que l'antécédent ainsi que le conséquent ne peuvent pas être vides :

$$(\bigvee_{a \in \mathcal{I}} x_a) \wedge (\bigvee_{a \in \mathcal{I}} y_a) \tag{3.8}$$

De la même manière que dans la formulation pour PEMEF, la formule suivante capture la couverture de l'antécédent :

$$\bigwedge_{t=(id,I) \in \mathcal{D}} (\neg q_t \leftrightarrow \bigvee_{a \in \mathcal{I} \setminus I} x_a) \tag{3.9}$$

Dans ce qui suit, la formule permettant de capturer la couverture de la règle d'association :

$$\bigwedge_{t=(id,I) \in \mathcal{D}} (\neg r_t \leftrightarrow (\neg q_t \vee \bigvee_{a \in \mathcal{I} \setminus I} y_a)) \tag{3.10}$$

La contrainte de cardinalité suivante permet de restreindre l'énumération aux règles ayant des supports qui ne sont pas inférieurs à α :

$$\sum_{t \in \mathcal{D}} r_t \geqslant \alpha \tag{3.11}$$

Concernant la contrainte suivante, elle est ajoutée pour respecter le quorum sur la confiance :

$$100 \times \sum_{t \in \mathcal{D}} r_t - \beta \times \sum_{t \in \mathcal{D}} q_t \geqslant 0 \tag{3.12}$$

Rappelons ici que β est présenté sous forme de pourcentage. En outre, il est important de noter qu'il existe plusieurs encodages polynomiaux efficaces des inégalités linéaires comme formules CNF dans la littérature (par exemple [ES06, BBR09]).

Nous notons $\mathcal{F}_{PERA}(\mathcal{D}, \alpha, \beta)$ la formulation correspondant à la conjonction de formules (3.7) $\wedge$ (3.8) $\wedge$ (3.9) $\wedge$ (3.10) $\wedge$ (3.11) $\wedge$ (3.12). Il est à signaler que, de par principalement l'utilisation du connecteur logique d'équivalence dans les formules (3.9) et (3.10), la formulation $\mathcal{F}_{PERA}(\mathcal{D}, \alpha, \beta)$ est bijective.

Pour restreindre l'énumération aux règles fermées, il faut simplement ajouter une formule reflétant le fait que l'union de l'antécédent et du conséquent correspond à un motif ensembliste fermé. Dans le cas de la formulation pour l'énumération des motifs ensemblistes fréquents, nous avons proposé la formule (3.5) pour exprimer la propriété de fermeture. Cette dernière se traduit comme suit dans le cas des règles d'association :

$$\bigwedge_{a \in \mathcal{I}} (((\bigwedge_{t=(id,I) \in \mathcal{D}} (r_t \rightarrow a \in I)) \wedge \neg x_a) \rightarrow y_a) \tag{3.13}$$

L'ajout des littéraux de la forme $\neg x_a$ vise uniquement à éviter les items apparaissant dans l'antécédent.

Pour l'énumération des règles minimales non redondantes, nous ajoutons seulement à la formulation des règles d'association fermées une formule permettant de considérer uniquement les règles où l'antécédent est un générateur minimal :

$$\overbrace{(\bigwedge_{a\in\mathcal{I}}(x_a \rightarrow (\bigvee_{t=(id,I)\in\mathcal{D},a\notin I}\ \bigwedge_{b\in\mathcal{I}\setminus(I\cup\{a\})}\neg x_b)))}^{sous-form.\ 1} \vee \overbrace{(\sum_{a\in\mathcal{I}} x_a = 1)}^{sous-form.\ 2} \tag{3.14}$$

La première sous-formule représente la propriété selon laquelle si un item a appartient à l'antécédent alors il existe une transaction contenant tous les items de l'antécédent excepté a, et par conséquent, si l'on réduit l'antécédent en supprimant a, cela aura comme conséquence l'augmentation du support. La deuxième sous-formule vient du fait que si l'antécédent ne contient qu'un unique item, alors il n'est pas possible de le réduire.

Pour réduire le nombre de clauses issues de (3.14), nous pouvons suivre l'approche de Tseitin, où notamment de nouvelles variables propositionnelles peuvent être utilisées pour représenter les sous-formules de la forme $\bigwedge_{b\in\mathcal{I}\setminus(I\cup\{a\})}\neg x_b : z \leftrightarrow (\bigwedge_{b\in\mathcal{I}\setminus(I\cup\{a\})}\neg x_b)$ avec z une nouvelle variable. Néanmoins, notons que même avec une telle transformation le nombre de clauses est de l'ordre $\mathcal{O}(|\mathcal{D}|\times|\mathcal{I}|^2)$. Ceci peut rendre problématique la tâche de l'énumération des modèles. Afin de réduire de manière plus importante le nombre de clauses, nous utilisons la formule suivante à la place de (3.14), où les variables de la forme z_t sont nouvelles :

$$(\bigwedge_{a\in\mathcal{I}}(x_a \rightarrow (\bigvee_{t=(id,I)\in\mathcal{D},a\notin I} z_t)) \wedge (\bigwedge_{t=(id,I)\in\mathcal{D}}(z_t \rightarrow \sum_{b\in\mathcal{I}\setminus I} x_b \leqslant 1))) \vee (\sum_{a\in\mathcal{I}} x_a = 1) \tag{3.15}$$

Cette transformation est possible car chaque sous-formule de la forme $\bigwedge_{b\in\mathcal{I}\setminus(I\cup\{a\})}\neg x_b$ dans (3.14) requiert les mêmes modèles que la contrainte AuPlusUn $\sum_{b\in\mathcal{I}\setminus I} x_b \leqslant 1$. Sachant qu'une contrainte AuPlusUn de la forme $\sum_{i=1}^{n} p_i \leqslant 1$ peut être encodée de manière linéaire en SAT (par exemple [Sin05]), le nombre de clauses issues de (3.15) est de l'ordre $\mathcal{O}(|\mathcal{D}|\times|\mathcal{I}|)$.

Intéressons-nous maintenant au problème de l'énumération des règles indirectes. Soit $(\mathcal{D},\alpha,\alpha',\beta)$ une instance de ce problème. Pour être en mesure d'obtenir une règle indirecte $\{a,b\}$ par une formulation en SAT, il faut saisir le support de $\{a,b\}$ ainsi que les supports et les confiances des deux règles d'association $\{a\}\Rightarrow M$ et $\{b\}\Rightarrow M$, où M est un médiateur. Ainsi, afin de représenter les deux éléments de chaque règle indirecte, nous associons à chaque item a deux variables distinctes x_a^1 et x_a^2. Nous associons aussi à chaque item a une variable supplémentaire y_a pour la représentation du médiateur. De plus, nous associons à chaque transaction t dans $\mathcal{D}$ quatre variables propositionnelles distinctes notées q_t^1, q_t^2, r_t^1 et r_t^2. Les variables de la forme q_t^1 et q_t^2 sont utilisées pour représenter les couvertures des deux composantes de chaque règle indirecte ; quant aux variables r_t^1 et r_t^2, elles servent à représenter les couvertures des règles d'association impliquant les médiateurs. De la même manière, nous associons également à chaque transaction t dans $\mathcal{D}$ une nouvelle variable propositionnelle notée s_t pour la représentation des couvertures des règles indirectes.

Dans un premier temps, nous introduisons la formule ψ_i qui permet d'encoder la règle d'association impliquant le médiateur liée au $i^{\text{ème}}$ item de la règle indirecte, pour $i = 1, 2$:

$$\begin{aligned}(\bigwedge_{a\in\mathcal{I}}(\neg x_a^i \vee \neg y_a)) \wedge ((\bigvee_{a\in\mathcal{I}} x_a^i) \wedge (\bigvee_{a\in\mathcal{I}} y_a)) \wedge (\bigwedge_{t=(id,I)\in\mathcal{D}} \neg q_t^i \leftrightarrow \bigvee_{a\in\mathcal{I}\setminus I} x_a^i) \wedge \\ (\bigwedge_{t=(id,I)\in\mathcal{D}} \neg r_t^i \leftrightarrow (\neg q_t^i \vee \bigvee_{a\in\mathcal{I}\setminus I} y_a)) \wedge \\ (\sum_{t\in\mathcal{D}} r_t^i \geqslant \alpha') \wedge (100 \times \sum_{t\in\mathcal{D}} r_t^i - \beta \times \sum_{t\in\mathcal{D}} q_t^i \geqslant 0) \wedge (\sum_{a\in\mathcal{I}} x_a^i \leqslant 1)\end{aligned} \tag{3.16}$$

En fait, la précédente formule correspond à la conjonction de $\mathcal{F}_{PERA}(\mathcal{D}, \alpha', \beta)$ avec la contrainte $\sum_{a\in\mathcal{I}} x_a^i \leqslant 1$ afin de ne considérer que les règles ayant des singletons comme antécédents.

La formule suivante permet de représenter les couvertures des règles indirectes en utilisant les variables de la forme s_t :

$$\bigwedge_{t\in\mathcal{D}} (s_t \leftrightarrow (q_t^1 \wedge q_t^2)) \tag{3.17}$$

Enfin, nous utilisons une contrainte de cardinalité pour exprimer la rareté du motif correspondant aux deux items composant chaque règle indirecte :

$$\sum_{t\in\mathcal{D}} s_t \leqslant \alpha \tag{3.18}$$

En conséquence, la formulation permettant de résoudre le problème de l'énumération des règles indirectes est la conjonction $\psi_1 \wedge \psi_2 \wedge$ (3.17) $\wedge$ (3.18). Une des remarques importantes relatives à cette formulation concerne la manière dont nous avons réutilisé notre formulation de départ pour les règles d'association. Elle montre en effet que les formulations peuvent être combinées de différentes façons pour répondre à plusieurs problèmes.

3.4 Motifs séquentiels fréquents

3.4.1 Enoncés des problèmes

Nous considérons dans cette section une tâche en fouille de données possédant une caractéristique intéressante autant dans la structure des données en entrée que dans la structure des motifs, qui est l'ordre entre les composantes de ces derniers. Il s'agit en effet de la tâche de l'énumération des motifs séquentiels fréquents [PRF+00, PCGS05, AU07]. Nos formulations en SAT ici montrent notamment comment la caractéristique de l'ordre peut être prise en compte.

Soit $\mathcal{I}$ un ensemble fini et non vide d'items. Une *séquence* sur $\mathcal{I}$ est une liste ordonnée $a_1 \cdots a_n$ d'items dans $\mathcal{I}$. Étant donné une séquence s, nous utiliserons $long(s)$ et $items(s)$ pour noter respectivement la longueur de s et l'ensemble des items apparaissant dans

cette dernière. En outre, nous utilisons $s[i]$ pour noter l'item apparaissant à la position i. Nous appelons *joker* un symbole particulier, noté $\circ$, distinct de tous les items possibles. Ce symbole est utilisé pour jouer le rôle de n'importe quel item dans un motif.

Définition 3.8 (Motif séquentiel)**.** *Un* motif séquentiel *sur l'ensemble d'items $\mathcal{I}$ est une séquence $\mu_1 \cdots \mu_k$ où $\mu_1 \in \mathcal{I}$, $\mu_k \in \mathcal{I}$ et $\mu_i \in \mathcal{I} \cup \{\circ\}$ pour $i = 2..k-1$.*

En d'autres termes, un motif séquentiel est une séquence sur $\mathcal{I} \cup \{\circ\}$, où le symbole $\circ$ n'est pas utilisé aux deux extrémités de cette dernière.

Étant donné une séquence $s = a_1 \cdots a_n$ et un motif séquentiel $\mu = \mu_1 \cdots \mu_k$, on dit que μ est inclus dans s à la position $i \in 1..(n-k+1)$, écrit $\mu \preceq_i s$, si pour tout $j \in 1..k$, on a $\mu_j = a_{i+j-1}$ ou $\mu_j = \circ$. La *couverture* de μ dans s, notée $\mathcal{C}(\mu, s)$, est l'ensemble des positions $\{i \in 1..(n-k+1) \mid \mu \preceq_i s\}$. Ainsi, le *support* de μ dans s, noté $\mathcal{S}(\mu, s)$, correspond à la valeur $|\mathcal{C}(\mu, s)|$. Par ailleurs, étant donné deux motifs μ et μ', nous écrivons $\mu \preceq \mu'$ s'il existe i tel que $\mu \preceq_i \mu'$, nous écrivons dans cette situation $\mu \prec \mu'$ si l'on a en plus $\mu \neq \mu'$.

Problème de l'énumération des motifs séquentiels fréquents (PEMSF).
- Entrée : une séquence s et un entier naturel non nul α jouant le rôle de quorum sur le support.
- Sortie : l'ensemble des motifs séquentiels $\mathcal{MSF}(s, \alpha) = \{\mu \mid \mathcal{S}(\mu, s) \geqslant \alpha\}$.

Considérons par exemple la séquence $s = aaccbcabcba$ ainsi que le motif séquentiel $\mu = a \circ c$. La couverture de μ est $\{1, 2, 7\}$ (voir la figure 3.2) et par conséquent son support est égal à 3. Par ailleurs, on a $\mathcal{MSF}(s, 3) = \{a, b, c, a \circ c\}$.

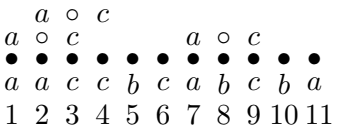


FIGURE 3.2 – Séquence d'items et motif séquentiel

Considérons maintenant une convention de notation utilisée par la suite. Étant donné un ensemble fini d'entiers naturels S et un entier naturel d, on utilisera $S + d$ pour noter l'ensemble $\{i + d \mid i \in S\}$.

Introduisons maintenant les deux représentations condensées les plus connues.

Définition 3.9 (Motif séquentiel fermé)**.** *Soit (s, α) une instance de PEMSF. Un motif séquentiel fréquent $\mu \in \mathcal{MSF}(s, \alpha)$ est dit* fermé *si, pour tout μ' avec $\mu \prec \mu'$, il n'y a pas d'entier naturel d tel que $\mathcal{C}(\mu, s) = \mathcal{C}(\mu', s) + d$.*

Définition 3.10 (Motif séquentiel maximal)**.** *Soit (s, α) une instance de PEMSF. Un motif séquentiel fréquent $\mu \in \mathcal{MSF}(s, \alpha)$ est dit* maximal *si, pour tout μ' avec $\mu \prec \mu'$, $\mathcal{S}(\mu', s) < \alpha$.*

Nous exposons à présent notre généralisation naturelle du problème de l'énumération des motifs séquentiels fréquents [JSS13b]. Elle est similaire en tout point à PEMSF, excepté que la tâche de fouille est effectuée sur une séquence d'ensembles d'items. Dans ce contexte, le rôle du joker est joué par l'ensemble vide de par son inclusion dans tout ensemble d'items. Notre généralisation permet d'exhiber des liens intéressants pour une meilleure compréhension des données issues de nombreux domaines. Par exemple, une séquence d'ensembles d'items peut être vue comme un enregistrement des articles achetés par un client sur une période donnée.

Une *séquence d'ensembles d'items* sur l'ensemble d'items $\mathcal{I}$ est une liste ordonnée $s = e_1 \cdots e_n$ d'ensembles d'items inclus dans $\mathcal{I}$. De la même manière que dans le cas des séquences d'items, nous utiliserons $long(s)$ et $items(s)$ pour noter respectivement la longueur de s, à savoir le nombre d'ensembles d'items, et l'ensemble d'items apparaissant dans cette dernière. Par ailleurs, un motif, appelé *ei-motif séquentiel*, est également défini comme une séquence d'ensembles d'items, sauf que les deux extrémités sont différentes de l'ensemble vide. De plus, on dit qu'un ei-motif séquentiel $\nu = \nu_1 \cdots \nu_k$ est inclus dans une séquence d'ensembles d'items $s = e_1 \cdots e_n$ à la position $i \in 1..(n-k+1)$, écrit $\nu \sqsubseteq_i s$, si pour tout $j \in 1..k$, on a $\nu_j \subseteq e_{i+j-1}$. Les notions de *couverture* et de *support* sont définies et notées de la même manière que dans le cas des séquences d'items en utilisant $\sqsubseteq_i$ à la place de $\preceq_i$. En outre, étant donné deux ei-motifs séquentiels ν et ν', nous écrivons $\nu \sqsubseteq \nu'$ s'il existe i tel que $\mu \sqsubseteq_i \mu'$, et nous écrivons $\nu \sqsubset \nu'$ si en plus $\nu \neq \nu'$. Quant aux représentations condensées, les ei-motifs séquentiels fermés et ceux maximaux sont définis comme dans respectivement la définition 3.9 et la définition 3.10 avec l'utilisation de $\sqsubset$ à la place de $\prec$.

Considérons par exemple la séquence d'ensembles d'items $s = \{a,b\}\{a,b\}\ \{c,d\}\{c,e\}\ \{f\}\{g\}\{d\}\{a,b,d\}\{f\}\{c\}$. Pour le ei-motif $\nu = \{a,b\}\{\}\{c\}$ on a $\mathcal{C}(\nu, s) = \{1,2,8\}$. De plus, le ei-motif ν est fermé contrairement à $\nu' = \{a\}\{\}\{c\}$ car $\mathcal{C}(\nu, s) = \mathcal{C}(\nu', s)$ et $\nu' \sqsubset \nu$.

3.4.2 Formulations en SAT

Nous débutons cette section par une description d'une formulation en SAT pour le problème de l'énumération des motifs séquentiels fréquents. Considérons ainsi une instance (s, α) de ce dernier problème. Nous associons à chaque item $a \in items(s)$, un ensemble de k_a variables propositionnelles distinctes notées $p^a_1, \ldots, p^a_{k_a}$, où $k_a = max(\mathcal{C}(a, s))$, autrement dit, la dernière position de a dans s. Intuitivement, la variable p^a_i est utilisée pour représenter le fait que a apparait dans le motif à la position i, ce qui explique l'arrêt à la position k_a. En particulier, pour une position i, si aucune variable de la forme p^a_i n'est vraie, alors cette position contient le joker. De plus, nous associons à chaque position $i \in 1..long(s) = n$ une variable distincte, notée q_i, pour la représentation de la couverture.

Nous commençons par une formule permettant d'éviter l'utilisation d'un joker au début d'un motif :

$$\bigvee_{a \in items(s)} p^a_1 \tag{3.19}$$

Il est évident que nous ne pouvons pas faire la même chose pour le dernier symbole dans un motif du fait de l'ignorance de la longueur de ce dernier a priori. Cependant, cela

ne constitue pas un problème, car il suffit de considérer le dernier symbole différent du joker comme la dernière composante du motif.

La formule suivante encode la couverture dans les variables de la forme q_i :

$$\bigwedge_{i\in 1..n} (\neg q_i \leftrightarrow (\bigvee_{a\in items(s)} \bigvee_{j\in 0..(k_a-1)} (p_j^a \wedge s[i+j] \neq a))) \tag{3.20}$$

Chaque constante de la forme $s[i+j] \neq a$ permet de vérifier que le motif n'apparait pas dans la position considérée i.

La contrainte de cardinalité suivante impose le respect du quorum sur le support :

$$\sum_{i=1}^{n} q_i \geqslant \alpha \tag{3.21}$$

Nous désignons par $\mathcal{F}_{PEMSF}(s,\alpha)$ la formulation correspondant à la conjonction (3.19) $\wedge$ (3.19) $\wedge$ (3.19). Nous pouvons constater d'ores et déjà que cette formulation est bijective du fait principalement de l'utilisation du connecteur logique de l'équivalence dans la formule (3.20).

À présent, afin de restreindre l'énumération aux motifs fermés, nous ajoutons les deux formules suivantes :

$$\bigwedge_{a\in items(s)} \bigwedge_{j\in 0..(k_a-1)} ((\bigwedge_{i\in 1..n} q_i \rightarrow s[i+j]=a) \rightarrow p_j^a) \tag{3.22}$$

$$\bigwedge_{a\in items(s)} \bigwedge_{j\in 1..k'_a} \neg(\bigwedge_{i\in 1..n} (q_i \rightarrow ((i-j \geqslant 1) \wedge s[i-j]=a))) \tag{3.23}$$

où $k'_a = n - min(\mathcal{C}(a,s))$. La première formule permet de minimiser le nombre d'occurrences du joker à droite tout en préservant le support ; quant à la seconde formule, elle permet d'éviter la possibilité d'ajouter des items à gauche sans réduction du support. Il est à noter que les valeurs entre 1 et k'_a sont utilisées pour capturer toutes les positions possibles de a à gauche d'un motif.

De manière similaire, pour l'énumération des motifs maximaux, nous ajoutons les deux formules suivantes :

$$\bigwedge_{a\in items(s)} \bigwedge_{j\in 0..(k_a-1)} ((\sum_{i=1}^{n} (q_i \wedge s[i+j]=a) \geqslant \alpha) \rightarrow p_j^a) \tag{3.24}$$

$$\bigwedge_{a\in items(s)} \bigwedge_{j\in 1..k'_a} \sum_{i=1}^{n} (q_i \wedge (i-j \geqslant 1) \wedge (s[i-j]=a)) < \alpha \tag{3.25}$$

La formule (3.24) exprime le fait que, tant que le support n'est pas inférieur à α, il faut réduire le nombre d'occurrences du joker à droite, tandis que la formule (3.25) exprime que l'ajout d'un item à gauche aboutira certainement à une réduction du support à une valeur inférieure au quorum.

Des formulations en SAT pour les problèmes décrits précédemment relatifs aux séquences d'ensembles d'items peuvent être simplement définies à partir des formulations que nous venons de présenter. Il suffit en réalité de remplacer les expressions de la forme $s[i+j] = a$, $s[i-j] = a$, $s[i+j] \neq a$ et $s[i-j] \neq a$ avec respectivement $a \in s[i+j]$, $a \in s[i-j]$, $a \notin s[i+j]$ et $a \notin s[i-j]$. Par exemple, la formulation suivante permet l'énumération des ei-motifs fréquents :

$$(\bigvee_{a \in items(s)} p_1^a) \wedge (\bigwedge_{i \in 1..n} (\neg q_i \leftrightarrow (\bigvee_{a \in items(s)} \bigvee_{j \in 0..(k_a - 1)} (p_j^a \wedge a \notin s[i+j]))))\wedge$$

$$(\sum_{i=1}^{n} q_i \geqslant \alpha)$$

Les légères modifications de nos formulations dans le cas des séquences d'items afin d'obtenir des formulations dans le cas des séquences d'ensembles d'items montrent clairement la grande flexibilité et la généricité de notre approche fondée sur SAT.

3.5 Résultats expérimentaux

Afin d'illustrer l'intérêt de l'approche reposant sur des formulations en SAT pour la fouille de données, nous décrivons dans cette section quelques résultats expérimentaux intéressants. Dans cet optique, nous concentrons cette étude sur des problèmes relatifs à l'énumération des règles d'association. Des résultats très similaires sont obtenus concernant les autres problèmes décrits précédemment.

Il s'agit principalement ici d'une évaluation expérimentale comparative avec des approches spécialisées. Nous considérons dans ce contexte les trois problèmes d'énumération des règles suivantes : les règles d'association (pures), les règles d'association fermées ainsi que les règles indirectes.

Pour énumérer l'ensemble des modèles des formules CNF issues de nos formulations, nous suivons l'approche que nous avons introduite dans [JLSS14] et qui est fondée sur une procédure de recherche DPLL. Dans nos expérimentations, l'heuristique de choix de variable pour la prise de décision se concentre en priorité sur les variables encodant les motifs. La puissance de cette approche consiste à utiliser la structure des littéraux observés pour effectuer efficacement la propagation unitaire. Notons également que les contraintes d'inégalités linéaires dédiées au support et à la confiance sont gérées dynamiquement sans transformation en forme CNF, conduisant ainsi à un algorithme d'énumération de modèles hybride SAT-CSP. En effet, ces contraintes sont gérées et propagées à la volée, comme cela est généralement réalisé en programmation par contraintes.

Dans nos résultats expérimentaux, nous utilisons `SAT4R` pour faire référence aux outils correspondant à notre approche, avec `R` $\in \{pure, closed, indirect\}$. La comparaison est faite avec deux outils spécialisés : `CORON`[2] [Sza06] et `SPMF`[3] [FGG+14]. Ces derniers sont implémentés en utilisant le langage de programmation JAVA et regroupent des solutions pour plusieurs tâches en fouille de données. Dans le cas des règles pures et fermées, notre

2. http ://coron.loria.fr/site/system.php
3. http ://www.philippe-fournier-viger.com/spmf/

approche est comparée à l'outil CORON. Concernant l'énumération des règles indirectes, la comparaison est faite avec l'outil SPMF.

Pour donner une idée des tailles des formules générées dans notre approche, la plus petite instance qui est associée à la base *Zoo-1* comporte 274 variables et 4379 clauses. Quant à la plus grande instance qui est associée à la base *Mushroom*, elle comporte 16486 variables et 1616795 clauses.

Dans notre comparaison avec les approches spécialisées considérées, nous procédons de la manière suivante :

— Pour les règles d'association pures et fermées, le support varie d'une valeur correspondant à 5% de la base à une valeur correspondant à 100% de cette dernière avec des sauts de 5%. La valeur de la confiance varie de la même manière. Ainsi, pour chaque base de données transactionnelles, nous générons 400 configurations.

— Concernant l'énumération des règles indirectes, pour toute instance $(\mathcal{D}, \alpha, \alpha', \beta)$, α varie d'une valeur correspondant à 10% de la base à une valeur correspondant à 100% de cette dernière avec des sauts de 10% ; α' et β varient de 20% à 100% avec des sauts de 20%. Cela conduit à 250 configurations pour chaque base de données transactionnelles.

Toutes les expérimentations ont été conduites sur une machine Intel Xeon quad-core avec 32 Go de RAM fonctionnant à 2,66 GHz. Par ailleurs, pour chacune des instances, nous fixons à 15 minutes le temps limite de résolution.

	SA4Pure		CORON (Pure)		SAT4Closed		CORON (Closed)	
data (#items, #trans, dens.)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)
Audiology (148, 216, 45)	20	855.002	20	855.01	20	855.001	20	855.019
Zoo-1 (36, 101, 44)	400	19.125	400	6.379	400	0.525	400	11.285
Tic-tac-toe (27, 958, 33)	400	0.090	400	0.240	400	0.097	400	0.230
Anneal (93, 812, 45)	101	709.508	101	678.417	147	604.097	103	679.313
Australian-credit (125, 653, 41)	245	370.171	264	321.628	268	323.298	226	403.723
German-credit (112, 1000, 34)	306	246.887	322	192.526	329	198.029	304	238.793
Heart-cleveland (95, 296, 47)	284	286.388	301	252.278	304	251.051	262	340.159
Hepatitis (68, 137, 50)	305	241.413	304	228.004	324	206.027	266	312.268
Hypothyroid (88, 3247, 49)	85	732.123	121	665.415	107	686.952	64	761.599
kr-vs-kp (73, 3196, 49)	172	552.929	203	487.739	192	523.669	146	590.893
Lymph (68, 148, 40)	336	181.648	338	170.373	387	63.220	291	281.351
Mushroom (119, 8124, 18)	366	109.12	387	46.0015	400	30.322	390	42.843
Primary-tumor (31, 336, 48)	400	3.688	400	1.173	400	2.031	400	18.826
Soybean (50, 650, 32)	400	2.909	400	1.507	400	0.177	400	7.947
Splice-1 (287, 3190, 21)	380	53.449	400	3.527	380	54.040	400	3.255
Vote (48, 435, 33)	380	66.7402	400	1.46463	400	32.4066	398	30.2207
Total	4560	279.761	**4741**	**247.292**	**4838**	**242.246**	4470	286.107

TABLE 3.2 – Règles d'associations pures et fermées : SAT4R vs CORON

Les tables 3.2 et 3.4 décrivent nos résultats comparatifs. Dans la première colonne, nous rapportons pour chaque base de données son nom et ses caractéristiques entre parenthèses : le nombre d'items ($\#items$), le nombre de transactions ($\#trans$) et la densité. De plus, pour chaque outil, nous rapportons le nombre de configurations résolues ($\#S$), et le temps moyen de résolution ($avg.\ time$ en secondes). Pour chaque instance non résolue, le temps est mis à 900 secondes (limite de temps). Dans la dernière ligne, nous donnons le nombre d'instances résolues et le temps moyen global en secondes.

Règles d'association pures. Les performances de CORON sont meilleures que SAT4Pure, car il résout 181 configurations de plus et il est meilleur sur tous les jeux de données considérés. Précisons que CORON procède en deux étapes pour l'énumération des règles

d'association pures : le calcul dans une première étape des motifs ensemblistes fréquents et ensuite la génération des règles d'association. Nous avons observé que `CORON` effectue la première étape de manière efficace. En effet, son temps ne dépasse pas quelques secondes sur la majorité des configurations considérées. Quant à la seconde étape, elle reste assez facile à réaliser. En conclusion, dans le cas des règles d'association pures, nous constatons que `CORON` est clairement meilleur que notre approche fondée sur SAT.

Règles d'association fermées. Sur cette catégorie de règles d'association, les performances de `SAT4Closed` sont meilleures que `CORON`. En effet, notre outil résout 368 configurations de plus que `CORON`. À l'exception du jeu de données *Splice-1*, `SAT4Closed` est le meilleur sur toutes les bases en termes de nombre de configurations résolues et de temps moyen. Par rapport à *Splice-1*, notons que le nombre de règles d'association fermées est faible (inférieur à 4000), ce qui pourrait expliquer pourquoi `CORON` est meilleur que notre outil sur ce jeu de données. Par ailleurs, il convient de signaler le fait que, généralement, plus la densité du jeu de données est élevée, meilleure est la performance de notre outil. Par exemple, sur le jeu de données *Lymph*, `SAT4Closed` est remarquablement plus efficace que `CORON` : il résout environ 100 configurations de plus.

Comme dans le cas des règles pures, `CORON` effectue l'extraction des règles fermées en deux étapes. Dans la première, l'ensemble de tous les motifs ensemblistes fermés fréquents est efficacement généré (en quelques secondes). La deuxième étape consistant à générer les règles fermées à partir des motifs trouvés prend, quant à elle, un temps plus important.

La figure 3.3 décrit le comportement des outils considérés sur deux jeux de données particuliers : *australian-credit* et *kr-vs-kp*. Nous varions un des paramètres (support ou confiance), tout en fixant le second. Dans le cas des règles pures, les deux outils `SAT4Pure` et `CORON` ont des comportements relativement semblables : la diminution du support augmente le temps nécessaire à la génération des règles. De plus, notons que les performances de notre outil s'approchent de celles de `CORON`, et parfois les surpassent, pour de grandes valeurs de la confiance et du support. Nous constatons un comportement similaire dans le cas des règles fermées.

support(%)	40	45	50	55	60	65	70
kr-vs-kp	7.67	5.68	3.64	2.99	2.46	1.95	1.67
australian-credit	12.38	8.13	5.61	4.29	3.23	2.62	2.01

TABLE 3.3 – Règles d'association pures vs Règles d'association fermées : #Règles pures/ #Règles fermées

Au long des expérimentations dans le cas des règles pures et celui des règles fermées, nous avons constaté de manière générale que `CORON` effectue efficacement la première étape de génération des motifs ensemblistes. Cependant, il prend un temps plus important pour effectuer la seconde étape de génération des règles à partir des motifs trouvés. Ce temps est plus important dans le cas des règles fermées, même si le nombre de ces dernières est inférieur à celui des règles pures. Dans la table 3.3, nous présentons la variation du rapport entre le nombre de règles pures et celui des règles fermées. Comme nous pouvons le constater, à mesure que le support diminue, le nombre de règles pures augmente rapidement par rapport au nombre de celles fermées. Cette dernière observation

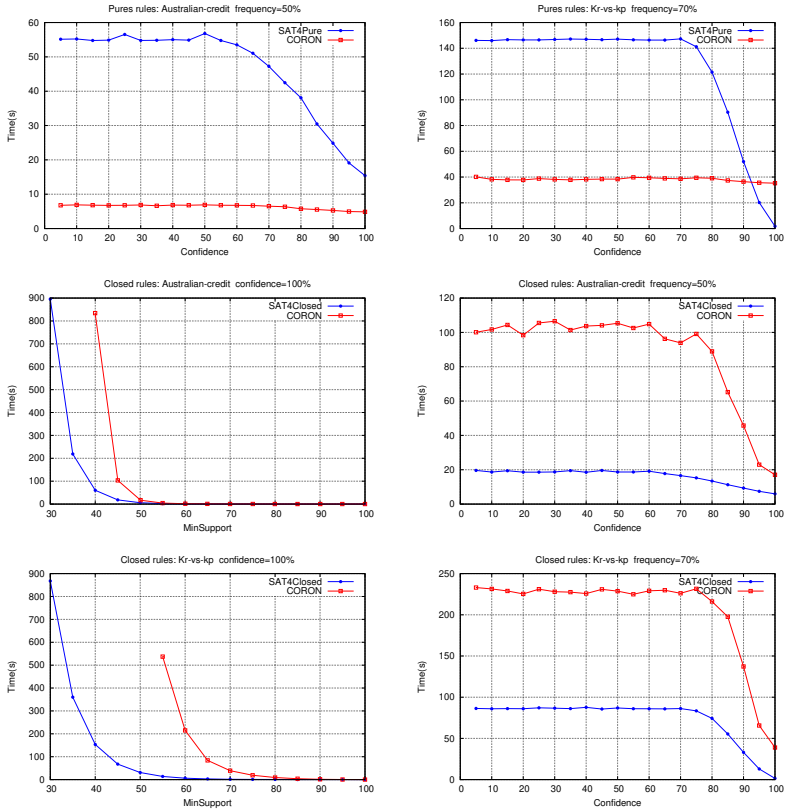


FIGURE 3.3 – Sélection d'instances : australian-credit et kr-vs-kp

pourrait expliquer pourquoi notre outil est plus efficace dans le cas des règles fermées. Globalement, `CORON` résout plus de configurations pour les règles pures que pour les règles fermées, tandis que notre approche est plus efficace pour l'extraction des règles fermées.

Règles indirectes. Les performances de notre outil sur ce type de règles sont très intéressantes. En effet, `SAT4Indirect` résout 572 instances de plus que `SPMF` (voir la table 3.4). De plus, `SAT4Indirect` est meilleur sur tous les jeux de données considérés. Comme nous pouvons le remarquer, le temps nécessaire à `SAT4Indirect` pour obtenir toutes les règles indirectes est relativement stable et très faible par rapport à `SPMF`. Même si le nombre de règles indirectes est très faible par rapport aux règles pures ou fermées, `SPMF` prend beaucoup de temps pour les trouver. Par exemple, si nous considérons le jeu de données *Hepatitis* avec les seuils : α'= 40%, β = 40% et α= 20%, `SPMF` prend plus de 122 secondes pour trouver seulement 359 règles, alors que le temps nécessaire pour `SAT4Indirect` pour les trouver ne dépasse pas une seconde. Nous pouvons également remarquer que pour certaines configurations, `SPMF` prend trop de temps sans trouver de règle indirecte sous la limite de temps de 900 secondes. En résumé, dans le cas des règles indirectes, `SAT4Indirect` surpasse largement `SPMF`.

Fort des résultats de notre étude expérimentale comparative, nous pouvons affirmer

	SAT4Indirect		SPMF	
data (#items, #trans, density)	#S	avg. time(s)	#S	avg. time(s)
Audiology (148, 216, 45)	124	453.743	61	680.451
Zoo-1 (36, 101, 44)	250	0.15684	250	9.12744
Tic-tac-toe (27, 958, 33)	250	0.09996	250	0.209
Anneal (93, 812, 45)	171	309.692	55	702.048
Australian-credit (125, 653, 41)	232	121.061	156	339.561
German-credit (112, 1000, 34)	244	49.0707	210	154.49
Heart-cleveland (95, 296, 47)	235	64.9748	203	300.488
Hepatitis (68, 137, 50)	245	32.9806	205	187.929
Hypothyroid (88, 3247, 49)	163	336.406	81	621.293
kr-vs-kp (73, 3196, 49)	204	206.47	114	499.337
Lymph (68, 148, 40)	250	6.1024	211	170.191
Mushroom (119, 8124, 18)	250	8.8906	250	29.6236
Primary-tumor (31, 336, 48)	250	0.15144	250	2.63892
Soybean (50, 650, 32)	250	0.05732	250	0.76692
Splice-1 (287, 3190, 21)	250	61.7394	250	0.50184
Vote (48, 435, 33)	250	0.8462	250	1.4814
Total	**3618**	**103.277**	3046	231.258

TABLE 3.4 – Règles d'association indirectes : `SAT4Indirect` vs `SPMF`

que pour les tâches d'extraction combinant plusieurs contraintes, notre approche fondée sur SAT peut être préférable à des outils d'extraction spécialisés.

3.6 Conclusion

Nous avons présenté dans ce chapitre une partie de nos contributions concernant l'extraction de motifs en fouille de données, et cela, en utilisant des formulations en SAT. Ces contributions ont permis de montrer que la modélisation en SAT, de par sa modularité et sa flexibilité, constitue un cadre adapté pour résoudre différents types de problèmes d'extraction de motifs. Un fait notable qu'il convient de relever est que l'utilisation de notre approche permet parfois de surpasser en performance des outils spécialisés, comme constaté dans le cadre de l'étude expérimentale précédemment présentée. En outre, il est à noter que nous avons également apporté des contributions originales pour la résolution de problèmes d'extraction de motifs autres que ceux présentés dans ce chapitre. Elles concernent particulièrement le problème de satisfiabilité des supports sur les motifs ensemblistes [SJS12], l'extraction de motifs avec des formes de préférence entre ces derniers [JKSS16, JSS17], et l'extraction de règles d'association non redondantes suivant une définition introduite dans [Zak04].

Chapitre 4

Optimisation et logique propositionnelle

Dans ce chapitre, nous exposons une partie de nos travaux en relation avec l'optimisation dans des cadres dérivés de SAT. Après une brève description des variantes d'optimisation de SAT les plus étudiées dans la littérature, nous présentons un problème que nous avons introduit dans nos travaux, appelé Top-K SAT. Nous décrivons ensuite une application d'un cadre d'optimisation fondé sur SAT dans le domaine de la persuasion automatique. Les éléments présentés ici proviennent majoritairement de nos travaux dans [JSS13c, JSS17, Sal19c].

4.1 Problèmes d'optimisation dérivés de SAT

Précisons d'abord ce que nous entendons par « optimisation » dans le cadre du problème SAT. Il s'agit de rechercher une ou plusieurs interprétations booléennes d'une formule propositionnelle en réalisant une sélection selon une ou plusieurs relations de préférence. Dans ce contexte, il convient de noter que les interprétations recherchées de la formule en entrée ne sont pas forcément des modèles de cette dernière.

Motivées par des utilisations pratiques, différentes variantes d'optimisation de SAT ont été proposées dans la littérature. Elles sont souvent fondées sur des fonctions objectif traduisant des relations de préférence sur les interprétations. Certaines de ces fonctions sont en lien avec les clauses falsifiées, tandis que d'autres reposent sur les valeurs de vérité affectées aux variables propositionnelles.

À titre d'illustration, considérons en premier lieu le problème MinCostSAT [FM06b]. Dans une instance de ce problème, il s'agit de trouver un modèle avec un coût minimal, où le coût d'un modèle est défini comme la somme des coûts des variables propositionnelles auxquelles on affecte la valeur de vérité 1. Ce problème utilise donc une fonction objectif relative aux valeurs de vérité affectées aux variables. Il est à mentionner que dans le cadre de nos travaux, nous avons utilisé une formulation dans ce problème pour effectuer une tâche en fouille de données [JKSS16].

En ce qui concerne l'utilisation d'une fonction objectif relative aux clauses falsifiées, le problème le plus étudié est sans conteste MaxSAT (voir par exemple [BHvW09]). Ce problème prend en entrée une formule en forme normale conjonctive et retourne une interprétation booléenne falsifiant un nombre minimum de clauses. Par exemple,

considérons la formule CNF $\phi = (p \vee q) \wedge (\neg p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee \neg q)$. Cette formule est clairement incohérente et une des solutions de MaxSAT est l'interprétation $\mathcal{B} = \{p \mapsto 1, q \mapsto 1\}$ qui ne falsifie qu'une unique clause, à savoir $(\neg p \vee \neg q)$.

Un premier problème dérivé de MaxSAT, appelé MaxSAT partiel, consiste à diviser l'ensemble de clauses en une partie dure et une partie souple. Une solution pour une instance de ce problème est une interprétation booléenne vérifiant l'ensemble des clauses appartenant à la partie dure et falsifiant un nombre minimum de clauses dans la partie souple. Considérons encore une fois la formule CNF ϕ fournie précédemment. Si l'on considère que la partie dure est uniquement composée de la clause $(\neg p \vee \neg q)$ dans ϕ, alors $\mathcal{B}$ ne peut être une solution, par contre l'interprétation $\mathcal{B}' = \{p \mapsto 1, q \mapsto 0\}$ l'est car elle satisfait à la fois la partie dure et un nombre maximum de clauses souples.

Une généralisation du problème MaxSAT partiel, appelé MaxSAT partiel pondéré (PW-MaxSAT), a été proposée en associant des coûts aux clauses souples. À cet égard, toute clause d'une instance de PW-MaxSAT est soit dure, soit souple avec un coût associé. L'objectif est ici de trouver une interprétation booléenne vérifiant toutes les clauses dures et minimisant la somme des coûts des clauses falsifiées. Par exemple, soit $\phi = \phi_h \wedge \phi_s$ une instance de PW-MaxSAT, où $\phi_h = (p \vee q) \wedge (p \vee r)$ est la partie dure et $\phi_s = (3 : \neg p) \wedge (1 : \neg q) \wedge (1 : \neg r)$ est la partie souple. Il n'est clairement pas possible de satisfaire à la fois la partie dure et la partie souple. L'interprétation booléenne $\mathcal{B} = \{p \mapsto 0, q \mapsto 1, r \mapsto 1\}$ est une solution de ϕ, car le coût de $\mathcal{B}$ est égal à 2 et le coût de tout autre modèle de ϕ_h est supérieur ou égal à 3.

De nombreux outils modernes dédiés aux précédents problèmes NP-difficiles reposent sur l'utilisation d'un solveur SAT comme sous-procédure profitant ainsi des grandes avancées relatives à SAT. En effet, depuis son introduction dans [FM06a], nous assistons depuis plusieurs années à de multiples améliorations de cette approche (voir par exemple [MP08, DB11, NB14, AG17]).

4.2 Top-K SAT

Les problèmes d'optimisation que nous avons décrits précédemment consistent à trouver au plus une interprétation booléenne pour chaque instance. Nous présentons dans cette section le problème Top-k SAT que nous avons introduit dans [JSS17], où une solution peut correspondre à plusieurs interprétations. En effet, ce problème consiste à calculer un type particulier de modèles appelés top-k modèles, où k est un entier naturel non nul. Intuitivement, un top-k modèle d'une formule est un modèle ayant moins de k modèles qui lui sont préférés selon une relation de préférence donnée. Dans [JSS17], nous avons notamment montré que le problème Top-k SAT est une généralisation de MaxSAT partiel. Nous avons également proposé un algorithme employant un solveur SAT comme sous-procédure pour résoudre ce problème.

Étant donné une formule propositionnelle ϕ, une *relation de préférence* $\succeq$ sur $Mods(\phi)$ est une relation binaire réflexive et transitive. L'expression $\mathcal{B} \succeq \mathcal{B}'$ signifie que $\mathcal{B}$ est au moins aussi préféré que $\mathcal{B}'$. Nous utilisons $P(\phi, \mathcal{B}, \succeq)$ pour noter le sous-ensemble de

$Mods(\phi)$ défini comme suit :

$$P(\phi, \mathcal{B}, \succeq) = \{\mathcal{B}' \in Mods(\phi) \mid \mathcal{B}' \succ \mathcal{B}\}$$

où $\mathcal{B}' \succ \mathcal{B}$ signifie que l'on a $\mathcal{B}' \succeq \mathcal{B}$ sans avoir $\mathcal{B} \succeq \mathcal{B}'$. L'ensemble précédent correspond donc à l'ensemble des modèles strictement préférés à $\mathcal{B}$.

Introduisons maintenant une relation d'équivalence, notée $\approx_X$, sur $P(\phi, \mathcal{B}, \succeq)$, où X est un ensemble de variables propositionnelles :

$$\mathcal{B}' \approx_X \mathcal{B}'' \text{ si et seulement si } {\mathcal{B}'}_{|X} = {\mathcal{B}''}_{|X}$$

Ainsi, il existe une partition de l'ensemble $P(\phi, \mathcal{B}, \succeq)$ en classes d'équivalence par $\approx_X$ que nous notons $[P(\phi, \mathcal{B}, \succeq)]^X$. Chaque classe d'équivalence est un ensemble de modèles préférés à $\mathcal{B}$ affectant les mêmes valeurs de vérité aux variables dans X. Dans notre étude, la relation d'équivalence $\approx_X$ est utilisée pour ne prendre en compte qu'un sous-ensemble de variables propositionnelles. En particulier, on introduit souvent de nouvelles variables pour la transformation en forme normale conjonctive de Tseitin [Tse68], et de telles variables peuvent n'avoir aucun influence sur les relations de préférence.

Définition 4.1 (Top-k Modèle)**.** *Soient ϕ une formule propositionnelle, $\mathcal{B}$ un modèle de ϕ, $\succeq$ une relation de préférence sur $Mods(\phi)$ et X un ensemble de variables propositionnelles tel que $X \subseteq Var(\phi)$. On dit que $\mathcal{B}$ est un modèle top-k suivant la relation $\succeq$ et l'ensemble X si et seulement si $|[P(\phi, \mathcal{B}, \succeq)]^X| \leqslant k - 1$.*

Le problème Top-k SAT.
Entrée : une formule propositionnelle ϕ, une relation de préférence $\succeq$ sur les modèles de ϕ, un ensemble de variables propositionnelles $X \subseteq Var(\phi)$ et un entier naturel $k > 0$.
Sortie : un ensemble $\mathcal{L}$ de modèles top-k de ϕ selon $\succeq$ et X vérifiant les deux propriétés suivantes :

1. pour tout modèle top-k $\mathcal{B}$, il existe $\mathcal{B}' \in \mathcal{L}$ tel que $\mathcal{B} \approx_X \mathcal{B}'$;
2. pour toute paire de modèles $\{\mathcal{B}, \mathcal{B}'\}$ dans $\mathcal{L}$, $\mathcal{B} \not\approx_X \mathcal{B}'$.

La première propriété permet d'avoir un représentant pour toute classe d'équivalence, quant à la seconde, elle permet de se restreindre à un unique représentant par classe d'équivalence.

La définition suivante introduit un type particulier de relations de préférence, appelées relations de δ-préférence. Ces relations nous ont notamment permis de proposer un algorithme par séparation et évaluation (en anglais *branch and bound*) pour résoudre le problème Top-k SAT.

Définition 4.2. *Soient ϕ une formule propositionnelle et $\succeq$ une relation de préférence sur les modèles de ϕ. Alors $\succeq$ est une relation de δ-préférence s'il existe une fonction $\delta^{\phi}_{\succeq}$ de $Mods(\phi)$ dans l'ensemble des formules propositionnelles, appelée* fonction de borne, *telle que (i) elle peut être calculée en temps polynomial en la taille de ϕ, et (ii) pour tout $\mathcal{B} \in Mods(\phi)$, $\mathcal{B}'$ est un modèle de $\phi \wedge \delta^{\phi}_{\succeq}(\mathcal{B})$ si et seulement si $\mathcal{B} \not\succ \mathcal{B}'$, pour toute interprétation booléenne $\mathcal{B}'$.*

Notons que pour un modèle $\mathcal{B}$ d'une formule ϕ, $\delta^{\phi}_{\succeq}(\mathcal{B})$ est une formule telle qu'en l'ajoutant par conjonction à ϕ avec la négation de $\mathcal{B}$, les modèles de la formule résultante sont différents de $\mathcal{B}$ et ils correspondent aux modèles de ϕ qui ne sont pas moins préférés que $\mathcal{B}$. Intuitivement, cela peut être vu comme un moyen d'introduire une borne inférieure durant le processus de sélection des interprétations composant une des solutions.

Clairement, si nous ignorons la condition (i), toute relation de préférence est également une relation de δ-préférence. En effet, pour chaque relation $\succeq$, nous avons uniquement besoin de définir la fonction de borne $\delta^{\phi}_{\succeq}$ comme suit :

$$\delta^{\phi}_{\succeq}(\mathcal{B}) = \bigwedge_{\mathcal{B}' \in Mods(\phi), \mathcal{B} \not\succeq \mathcal{B}'} \overline{\mathcal{B}'}$$

où $\overline{\mathcal{B}'}$ est la clause correspondant à la négation de $\mathcal{B}'$. Cette définition signifie que l'on exclut tous les modèles de ϕ qui sont moins préférés que $\mathcal{B}$.

Il est à relever que nous avons proposé dans nos travaux plusieurs applications en fouille de données de la modélisation en Top-k SAT. Nous avons, entre autres, considéré le problème lié aux motifs ensemblistes introduit dans [WHLT05], ainsi qu'une variante similaire dans le cas des motifs séquentiels.

4.3 Un cas d'utilisation : la persuasion

Les technologies de persuasion visent principalement à inciter les utilisateurs à apporter des changements psychologiques et/ou physiques (p. ex. pensées, sentiments, comportements), et cela, à des fins liées à plusieurs domaines tels que la santé, l'éducation, la politique, et le marketing (pour une définition intéressante et plus complète, voir [Fog98, Hun18]). L'un des points clés dans l'activité de persuasion est l'utilisation explicite et appropriée d'arguments convaincants [Hun16, Hun18]. Dans ce contexte, il est important de noter que dans [Hun14] l'auteur propose des conditions intéressantes pour la persuasion centrée sur l'utilisation d'arguments. Certaines de ces conditions sont essentielles dans le travail décrit dans cette section, notamment celle consistant à toujours privilégier les courtes séquences d'arguments.

Parmi les travaux intéressants en relation avec l'activité de persuasion par arguments, nous pouvons premièrement mentionner ceux utilisant l'approche dialogique, où la persuasion est définie comme un dialogue entre deux agents essayant de se convaincre l'un l'autre en échangeant des arguments (p. ex. [APM00, AMP00, Pra05, BM11, AdS13, HT16]). Nous pouvons aussi mentionner l'approche asymétrique proposée dans [Hun15]. Le mot « asymétrique » fait référence au fait que l'agent à persuader ne peut pas utiliser de contre-arguments, mais il peut uniquement accepter ou rejeter chacun des arguments employés. L'avantage principal d'un système de persuasion asymétrique est qu'il permet d'éviter des traitements lourds relatifs aux langages naturels. Nous avons en particulier proposé une approche asymétrique dans [Sal19c]. Plusieurs autres approches récentes ont été proposées dans [HP17a, HP17b, HH18, CHH+18].

Nous présentons dans cette section un cadre pour la persuasion automatique centré sur l'utilisation d'arguments que nous avons introduit dans [Sal19c]. Ce cadre repose

sur l'utilisation de formulations en MaxSAT partiel pondéré. Il s'agit ici de montrer l'utilité de l'optimisation en logique propositionnelle grâce à un cas d'utilisation issu de nos travaux.

4.3.1 Approches de persuasion

Nous considérons dans notre étude que l'activité de persuasion consiste à employer une séquence d'arguments permettant à l'agent persuadeur d'atteindre son objectif en utilisant l'une des trois approches suivantes :

- **Approche forte :** atteindre l'objectif en montrant qu'il est une conséquence des connaissances, croyances et souhaits de l'agent à persuader.
- **Approche faible :** atteindre l'objectif en montrant qu'il a pour conséquence des souhaits de l'agent à persuader (rendre l'objectif attractif).
- **Approche mixte :** combiner de différentes manières les deux approches précédentes. Par exemple, utiliser l'approche forte pour atteindre une partie de l'objectif et l'approche faible pour atteindre la partie restante.

Par exemple, en employant l'approche forte, le persuadeur peut utiliser l'argument selon lequel le fait que l'agent à persuader sait qu'un produit donné est efficace et pas cher a pour conséquence que ce produit doit être acheté : *l'achat du produit considéré* est le but de l'agent persuadeur et le fait que *ce produit soit efficace et peu coûteux* sont des croyances de l'agent à persuader. Par ailleurs, en employant l'approche faible, l'agent persuadeur peut utiliser l'argument affirmant que la pratique d'un sport a pour conséquence le développement d'amitiés : *faire du sport* est le but de l'agent persuadeur et *développer des amitiés* est un souhait de l'agent à persuader. Précisons que l'argument précédent ne signifie pas que le sport est l'unique moyen de développer des amitiés. En résumé, dans l'approche forte, l'agent persuadeur débute la persuasion par les connaissances, les croyances et les souhaits de celui à persuader, tandis que dans l'approche faible, il commence par l'objectif visé.

Précisons que nous définissons dans notre cadre de persuasion un argument comme un couple d'ensembles de littéraux. En un sens, cela peut être vu comme une abstraction de la structure d'argument standard fondée sur la logique (voir par exemple [BH08]). En effet, un argument fondé sur des formules logiques est défini comme un couple composé d'un ensemble de formules, représentant le support, et d'une formule représentant la conclusion. De cette manière, au lieu d'utiliser des formules logiques, nous utilisons ici des littéraux. Dans ce contexte, notre objectif est d'utiliser une structure intermédiaire simple située entre l'approche logique et celle abstraite de Dung [Dun95]. Dans l'approche de Dung la structure interne des arguments est totalement ignorée.

4.3.2 Définition formelle

Nous définissons dans cette partie la notion de cadre de persuasion considéré et quelques notions connexes. Ensuite, nous introduisons un problème de décision, appelé satisfiabilité de persuasion, qui est défini comme le problème de déterminer s'il existe une séquence d'arguments débutant par un ensemble donné de littéraux, appelé état initial, et permettant d'obtenir un autre ensemble de littéraux, appelé état objectif.

Comme convention de notation, nous utilisons $\mathcal{L}(V)$ pour noter l'ensemble $V \cup \{\neg p \mid p \in V\}$, où V est un ensemble de variables propositionnelles.

Définition 4.3 (Cadre de persuasion)**.** *Un* cadre de persuasion *est un triplet* $(\mathcal{S}, \mathcal{A}, W)$*, où* $\mathcal{S}$ *est un ensemble fini non vide de variables propositionnelles, représentant des énoncés de manière abstraite,* $\mathcal{A}$ *un ensemble fini d'arguments construits à partir de* $\mathcal{S}$*, et* W *une fonction associant un poids (un entier) à chaque argument dans* $\mathcal{A}$*. Un* argument *a sur* $\mathcal{S}$ *est un couple* $\langle X, C\rangle$*, où* $X, C \subseteq \mathcal{L}(\mathcal{S})$*,* $X \cup C$ *est cohérent,* $X \cap C = \emptyset$ *et* $C \neq \emptyset$*. L'ensemble* X *est appelé* support *de l'argument a et l'ensemble* C *est appelé* conclusion *de l'argument a.*

Nous utilisons $Supp(a)$ et $Conc(a)$ pour noter respectivement le support et la conclusion de l'argument a.

Le poids d'un argument est utilisé pour représenter le coût de ce dernier. En particulier, la fonction de poids peut être utilisée comme une mesure de la faiblesse des arguments quant à l'impact sur l'agent à persuader.

Exemple 4.1. *Considérons d'abord l'ensemble des énoncés* $\mathcal{S}$ *:*

- ad_i *: la publicité numéro i est convaincante ;*
- h *: penser que le produit p est bon pour la santé ;*
- eff *: penser que le produit p est efficace ;*
- exp *: penser que le produit p est cher ;*
- $pack$ *: aimer l'emballage du produit p ;*
- buy *: être convaincu de l'achat du produit p.*

Dans cet exemple, nous considérons l'ensemble d'arguments $\mathcal{A}$ *:* $a_1 = \langle\{ad_1\}, \{h, eff\}\rangle$*,* $a_2 = \langle\{ad_2\}, \{\neg exp\}\rangle$*,* $a_3 = \langle\{ad_3\}, \{pack\}\rangle$*,* $a_4 = \langle\{h, eff, \neg exp, pack\}, \{buy\}\rangle$*,* $a_5 = \langle\{h, eff, pack\}, \{buy\}\rangle$ *et* $a_6 = \langle\{pack\}, \{buy\}\rangle$*. Par exemple, l'argument* a_6 *signifie que si l'agent à persuader aime l'emballage du produit p, alors il peut être convaincu de l'achat de ce dernier. Nous pouvons raisonnablement affirmer que l'argument* a_4 *est plus fort que* a_5*, qui est quant à lui plus fort que* a_6*. Cela peut être représenté par une fonction de poids attribuant les valeurs suivantes :* $W(a_4) = 0$*,* $W(a_5) = 1$ *et* $W(a_6) = 2$*.*

Définition 4.4 (Chemin d'arguments)**.** *Étant donné un cadre de persuasion* $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$ *et un ensemble de littéraux* $I \subseteq \mathcal{L}(\mathcal{S})$*, un* I-chemin *dans* $\mathcal{F}$ *est une séquence* $s = a_1, \ldots, a_k$ *d'arguments distincts dans* $\mathcal{A}$ *où, pour tout entier* $i \in 1..k$*, on a la propriété* $Supp(a_i) \subseteq I \cup \bigcup_{1 \leqslant j < i} Conc(a_j)$*.*

Notre condition sur une séquence d'arguments pour être un I-chemin signifie seulement que l'on doit obtenir le support d'un argument avant de faire appel à lui. Notons que l'ensemble I représente l'état initial.

Étant donné un I-chemin $s = a_1, \ldots, a_k$, nous utilisons $Arg(s)$, $Leng(s)$, $Weight(s)$ et $Conc(s)$ pour noter respectivement $\{a_1, \ldots, a_k\}$, la longueur k, $\sum_{i=1}^{k} W(a_i)$ et $I \cup \bigcup_{i=1}^{k} Conc(a_i)$.

Définition 4.5 (Cohérence)**.** *Un* I*-chemin* $s = a_1, \ldots, a_k$ *est dit* cohérent *si l'ensemble de littéraux* $Conc(s)$ *est cohérent.*

En d'autres termes, un I-chemin cohérent est une séquence d'arguments qui peuvent être utilisés ensemble en partant de l'état initial I et sans impliquer des informations contradictoires. Considérons encore une fois l'exemple 4.1. La séquence a_1, a_2, a_3, a_4 est un $\{ad_1, ad_2, ad_3\}$-chemin cohérent, mais elle n'est pas un $\{ad_1, ad_2, ad_3, exp\}$-chemin cohérent, car a_2 produit le littéral $\neg exp$ qui est en contradiction avec l'état initial.

Définition 4.6 (Satisfiabilité de persuasion)**.** *Étant donné un cadre de persuasion* $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$*, un ensemble cohérent de littéraux* $I \subseteq \mathcal{L}(\mathcal{S})$*, appelé* état initial*, un ensemble non vide et cohérent de littéraux* $O \subseteq \mathcal{L}(\mathcal{S})$*, appelé* état objectif*, une borne sur la longueur* $k \in \mathbb{N} \cup \{\infty\}$ *et une borne sur le poids* $b \in \mathbb{Z} \cup \{\infty\}$*, le problème de la satisfiabilité de persuasion consiste à déterminer s'il existe un* I*-chemin cohérent* s *dans* $\mathcal{F}$ *tel que* $O \subseteq Conc(s)$*,* $Leng(s) \leqslant k$ *et* $Weight(s) \leqslant b$*.*

Nous notons chaque instance du problème précédent comme un tuple de la forme $(\mathcal{F}, I, O, k, b)$. Précisons que la valeur ∞ est uniquement utilisée pour représenter l'absence de borne.

Exemple 4.2. *Nous fournissons ici un exemple inspiré de [CHH+18]. Considérons l'ensemble suivant d'énoncés :* s_1 *= bonne santé,* s_2 *= arrêter de fumer,* s_3 *= longue vie,* s_4 *= bonne apparence,* s_5 *= soutenir sa famille. Soit* $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$ *un cadre de persuasion tel que* $\mathcal{S} = \{s_1, s_2, s_3, s_4, s_5\}$*,* $\mathcal{A} = \{a_1 = \langle\{s_1\}, \{s_2\}\rangle, a_2 = \langle\{s_3\}, \{s_1\}\rangle, a_3 = \langle\{s_4\}, \{s_1\}\rangle, a_4 = \langle\{s_5\}, \{s_3\}\rangle, a_5 = \langle\{s_4\}, \{s_2\}\rangle\}$*, et* $W(a_1) = 3$*,* $W(a_2) = 2$*,* $W(a_3) = 2$*,* $W(a_4) = 1$ *et* $W(a_5) = 6$*. Dans ce contexte, nous considérons l'instance du problème de la satisfiabilité de persuasion* $P = (\mathcal{F}, \{s_5\}, \{s_2\}, 3, 6)$*. En utilisant l'approche forte, l'objectif dans* P *est de convaincre l'agent à persuader d'arrêter de fumer en utilisant le fait qu'il est important pour lui de soutenir sa famille. Le* $\{s_5\}$*-chemin* a_4, a_2, a_1 *est une solution de* P *qui correspond à l'utilisation dans l'ordre des éléments suivants : soutenir sa famille* $\rightarrow$ *longue vie* $\rightarrow$ *bonne santé* $\rightarrow$ *arrêter de fumer.*

En utilisant particulièrement le problème de cycle hamiltonien, nous avons obtenu le théorème suivant.

Théorème 4.1. *Le problème de satisfiabilité de persuasion est NP-complet.*

4.3.3 Formulation en MaxSAT partiel pondéré

Nous proposons dans cette section une formulation en PW-MaxSAT pour résoudre le problème de la satisfiabilité de persuasion. En fait, vu que l'outil de modélisation est un problème d'optimisation, notre formulation permet de trouver une solution possédant la plus petite valeur possible pour le poids.

Soit $P = (\mathcal{F}, I, O, k, b)$ une instance du problème de la satisfiabilité de persuasion, où $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$. Afin de définir notre formulation pour P, nous avons besoin de certains éléments syntaxiques. Premièrement, nous associons à chaque littéral $l \in \mathcal{L}(\mathcal{S})$ un ensemble de $k+1$ variables propositionnelles distinctes, notées $p_l^0, \ldots, p_l^k$. La variable p_l^i est employée pour exprimer que le littéral l est produit à la $i^{\text{ème}}$ étape (le $i^{\text{ème}}$ argument quand $i \geqslant 1$). En particulier, p_l^0 obtient la valeur 1 si et seulement si l appartient à l'état initial I. De plus, étant donné un sous-ensemble de littéraux $X \subseteq \mathcal{L}(\mathcal{S})$ et $0 \leqslant i \leqslant k$, nous utilisons $R(X, i)$ pour noter l'ensemble de variables $\{p_l^i \mid l \in X\}$. Nous associons aussi à chaque argument $a \in \mathcal{A}$ une variable propositionnelle distincte q_a et un ensemble de k autres variables propositionnelles, notées $r_a^1, \ldots, r_a^k$. La variable q_a est utilisée pour exprimer que l'argument a apparait dans la solution, et de manière similaire, la variable r_a^i est utilisée pour exprimer que l'argument a apparait dans la solution à l'étape i. Par ailleurs, pour tout littéral $l \in \mathcal{L}(\mathcal{S})$, nous employons $Arg(l, \mathcal{A})$ pour noter l'ensemble d'arguments $\{\langle X, C\rangle \in \mathcal{A} \mid l \in C\}$. Nous généralisons cette notation aux ensembles de

littéraux comme suit : $Arg(Y, \mathcal{A}) = \{\langle X, C\rangle \in \mathcal{A} \mid Y \cap C \neq \emptyset\}$.

Commençons par la description de la partie dure de notre formulation. Dans ce cadre, notre première formule exprime que I est l'état initial :

$$\bigwedge_{l \in I} p_l^0 \wedge \bigwedge_{l' \in \mathcal{L}(\mathcal{S}) \setminus I} \neg p_{l'}^0 \tag{4.1}$$

Dans ce qui suit, nous exigeons des I-chemins cohérents :

$$\bigwedge_{e \in \mathcal{S}} \bigwedge_{i=0}^{k} \bigwedge_{j=0}^{k} (\neg p_e^i \vee \neg p_{\neg e}^j) \tag{4.2}$$

La troisième formule est utilisée pour lier la vérité de chaque variable de la forme p_l^i à l'utilisation à l'étape i d'au moins un argument contenant l dans sa conclusion :

$$\bigwedge_{l \in \mathcal{L}(\mathcal{S})} \bigwedge_{i=1}^{k} (p_l^i \rightarrow \bigvee_{a \in Arg(l, \mathcal{A})} r_a^i) \tag{4.3}$$

La prochaine formule signifie que l'utilisation d'un argument à l'étape i requiert la vérité de son support avant cette étape :

$$\bigwedge_{a = \langle X, C\rangle \in \mathcal{A}} \bigwedge_{i=1}^{k} (r_a^i \rightarrow (\bigwedge_{l \in X} \bigvee_{j=0}^{i-1} p_l^i)) \tag{4.4}$$

Nous introduisons maintenant la formule énonçant que si un argument est utilisé à l'étape i, alors les littéraux de sa conclusion sont vrais à cette étape :

$$\bigwedge_{a = \langle X, C\rangle \in \mathcal{A}} \bigwedge_{i=1}^{k} (r_a^i \rightarrow \bigwedge R(C, i)) \tag{4.5}$$

où $\bigwedge R(C, i)$ représente la conjunction des littéraux dans $R(C, i)$.
La formule suivante traduit le fait que chaque argument est employé au plus une fois :

$$\bigwedge_{a \in \mathcal{A}} \sum_{i=1}^{k} r_a^i \leqslant 1 \tag{4.6}$$

De manière similaire, la prochaine formule signifie qu'au plus un argument est utilisé à chaque étape :

$$\bigwedge_{i=1}^{k} \sum_{a \in \mathcal{A}} r_a^i \leqslant 1 \tag{4.7}$$

La conjonction de clauses suivante représente le fait que l'objectif doit être obtenu :

$$\bigwedge_{l \in O \setminus I} \bigvee_{i=1}^{k} p_l^i \tag{4.8}$$

La dernière formule de la partie dure est uniquement employée pour associer la vérité de chaque variable de la forme q_a à l'utilisation de l'argument a :

$$\bigwedge_{a \in \mathcal{A}} ((\bigvee_{i=1}^{k} r_a^i) \leftrightarrow q_a) \tag{4.9}$$

En ce qui concerne la partie souple de notre formulation, elle est composée des clauses pondérées suivantes :

$$W(a) : \neg q_a \quad \textit{pour chaque } a \in \mathcal{A} \tag{4.10}$$

Par conséquent, chaque clause souple permet d'associer chaque argument à son poids.

À partir de maintenant, nous utilisons $\mathcal{E}_{Pers}(P)$ pour se référer à la formulation que nous venons de définir.

Proposition 4.1 (Correction)**.** *Soit $P = (\mathcal{F}, I, O, k, b)$ une instance du problème de la satisfiabilité de persuasion. Alors, P admet une solution si et seulement si $\mathcal{E}_{Pers}(P)$ admet une solution $\mathcal{B}$ telle que $cost(\mathcal{B}) \leqslant b$, où $cost(\mathcal{B})$ correspond au coût de l'interprétation $\mathcal{B}$.*

Afin de montrer la flexibilité de l'approche ayant comme base PW-MaxSAT, nous considérons certaines variantes du problème de la satisfiabilité de persuasion. Notre première variante repose sur une propriété sur les chemins, appelée cohérence faible, qui formalise le fait que l'agent peut masquer des parties des conclusions des arguments utilisés afin de présenter un chemin cohérent.

Définition 4.7 (Cohérence faible)**.** *Un I-chemin $a_1, \ldots, a_k$ est dit* faiblement cohérent *si l'ensemble de littéraux $I \cup \bigcup_{i=1}^{k} Supp(a_i)$ est cohérent.*

Considérons par exemple le cadre de persuasion $\mathcal{F} = (\{a, b, c, d\}, \{a_1 = \langle\{a\}, \{b, c\}\rangle, a_2 = \langle\{d\}, \{\neg c\}\rangle\}, W)$, où $W(a_1) = 1$ et $W(a_2) = 1$, et l'instance $P = (\mathcal{F}, \{a, d\}, \{b, \neg c\}, 2, 2)$. Clairement, il n'existe pas de $\{a, d\}$-chemin cohérent solution de P. Cependant, la séquence a_1, a_2 est un $\{a, d\}$-chemin faiblement cohérent permettant d'obtenir l'objectif $\{b, \neg c\}$ à partir de $\{a, d\}$.

Afin de prendre en compte la cohérence faible à la place de la cohérence dans notre formulation en PW-MaxSAT, nous avons seulement besoin de supprimer la formule (4.5). En effet, cette formule est utilisée pour imposer la satisfaction des variables propositionnelles représentant les conclusions des arguments considérés. Il est important de préciser ici que ce sont les formules (4.2) et (4.4) qui permettent de satisfaire les variables représentant les supports.

De manière générale, il serait intéressant de permettre à l'agent persuadeur de réaliser une sélection des littéraux pouvant être omis pour éviter l'incohérence. Par exemple, l'agent peut décider d'omettre uniquement les négations des littéraux appartenant à l'objectif ou l'état initial. Pour l'adaptation de notre formulation à cette situation, nous avons simplement besoin de remplacer (4.5) par :

$$\bigwedge_{a=\langle X,C\rangle \in \mathcal{A}} \bigwedge_{i=1}^{k} (r_a^i \rightarrow \bigwedge R(C \setminus T, i)) \tag{4.11}$$

où T est l'ensemble contenant les littéraux pouvant être omis. Cette formule énonce que si un argument est utilisé à l'étape i alors les littéraux de sa conclusion sont vrais excepté ceux dans T.

Les variantes précédentes montrent la flexibilité du cadre proposé et de notre solution fondée sur l'utilisation d'une formulation PW-MaxSAT. Dans ce contexte, nous pouvons facilement définir d'autres variantes prenant en compte d'autres aspects intéressants. Pour mieux illustrer ce point, nous proposons une variante où l'on considère des conflits entre les arguments disponibles. Plus précisément, étant donné un cadre de persuasion $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$, nous considérons un graphe de conflits $G = (\mathcal{A}, E)$, où chaque arrête $\{a, a'\}$ dans E exprime que les arguments impliqués a et a' ne peuvent être utilisés ensemble dans une même solution. Pour cette variante, notre formulation peut être adaptée en ajoutant la formule suivante à la partie dure :

$$\bigwedge_{\{a,a'\}\in E} (\sum_{i=1}^{k} r_a^i) + (\sum_{j=1}^{k} r_{a'}^j) \leqslant 1 \tag{4.12}$$

Supposons sans perte de généralité qu'il existe $i \in 1..k$ tel que r_a^i possède comme valeur de vérité 1. Alors, on a $\sum_{i=1}^{k} r_a^i = 1$, et en conséquence on obtient $\sum_{j=1}^{k} r_{a'}^j = 0$, ce qui assure que a' n'est pas utilisé dans la solution.

Plusieurs autres variantes peuvent être définies en raisonnant sur différents aspects, tels que l'argument d'ouverture et le dernier argument.

4.3.4 Optimum de Pareto

Dans le problème de la satisfiabilité de persuasion, nous utilisons explicitement des contraintes de borne sur la longueur et le poids de la solution. Le choix de ces bornes peut être arbitraire en l'absence de connaissances spécifiques en lien avec le cas considéré, ce qui peut être vu comme un réel inconvénient de notre approche. Pour éviter l'utilisation de contraintes de borne, nous considérons ici la notion d'optimum de Pareto. Cette partie nous permet donc d'aborder l'optimisation multi-objective dans des cadres dérivés du problème SAT.

Nous définissons une *PO-instance* comme un triplet de la forme $P = (\mathcal{F}, I, O)$, où $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$ est un cadre de persuasion, $I \subseteq \mathcal{L}(\mathcal{S})$ un ensemble de littéraux cohérent (état initial), et $O \subseteq \mathcal{L}(\mathcal{S})$ un ensemble non vide de littéraux cohérent (objectif). Clairement, une PO-instance peut être vue comme une instance du problème de la satisfiabilité de persuasion sans les bornes sur la longueur et le poids. De ce point de vue, on dit qu'une séquence d'arguments s est une solution d'une PO-instance $P = (\mathcal{F}, I, O)$ si elle est une solution de l'instance du problème de la satisfiabilité de persuasion $(\mathcal{F}, I, O, \infty, \infty)$.

Définition 4.8 (Solution Pareto-optimale)**.** *Une* solution Pareto-optimale *d'une PO-instance P est une solution s de P où, pour toute solution s' de P, (i) si $Leng(s') < Leng(s)$ alors on a $Weight(s) < Weight(s')$, et (ii) si $Weight(s') < Weight(s)$ alors on a $Leng(s) < Leng(s')$.*

Étant donné deux solutions s et s' d'une PO-instance, on dit que s *domine* s' si au moins une des propriétés suivantes est vérifiée :

— $Leng(s) < Leng(s')$ et $Weight(s) \leqslant Weight(s')$;
— $Weight(s) < Weight(s')$ et $Leng(s) \leqslant Leng(s')$.

En d'autres termes, s domine s' si s est meilleure que s' sur un des critères (longueur ou poids) et s n'est pas plus mauvaise que s' sur l'autre critère. Notons qu'une solution est Pareto-optimale si et seulement si elle n'est pas dominée par une autre solution.

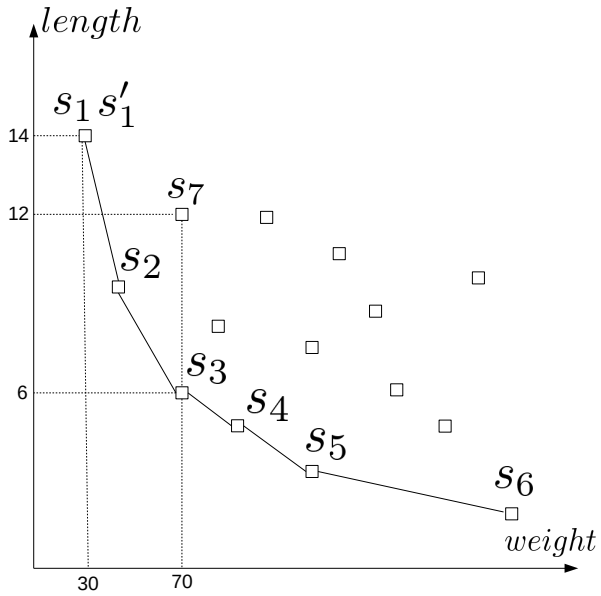


FIGURE 4.1 – Front de Pareto

À titre d'illustration, dans la figure 4.1, nous montrons les solutions d'une PO-instance donnée. Il est important de préciser au préalable qu'un point peut représenter plus d'une solution, par exemple s_1 et s'_1, car il peut exister plusieurs solutions de même longueur et de même poids. En outre, les solutions s_1, s'_1 et s_3 sont Pareto-optimales, ce qui n'est pas le cas de la solution s_7 qui est dominée par s_3 ($Leng(s_3) = 6 < Leng(s_7) = 12$ et $Weight(s_3) = Weight(s_7) = 70$). Par ailleurs, s_1 ne domine pas s_3 car $Leng(s_3) = 6$ et $Leng(s_1) = 14$, et s_3 ne domine pas s_1 car $Weight(s_1) = 30$ et $Weight(s_3) = 70$.

Nous appelons *front de Pareto* d'une PO-instance l'ensemble de toutes les solutions Pareto-optimales. Le front de Pareto dans l'exemple décrit dans la figure 4.1 est l'ensemble $\{s_1, s'_1, s_2, s_3, s_4, s_5, s_6\}$.

Notre formulation en PW-MaxSAT pour résoudre le problème de la satisfiabilité de persuasion permet de calculer une solution vérifiant la contrainte relative à la borne sur la longueur et possédant la plus petite valeur possible pour le poids. Plus précisément, étant donné une instance $P = (\mathcal{F}, I, O, k, b)$, toute solution de la formulation $\mathcal{E}_{Pers}(P)$ vérifie $Leng(s) \leqslant k$ et $Weight(s) \leqslant Weight(s')$ pour toute solution s' de P. Partant de ce point, si s est une solution de P obtenue grâce à $\mathcal{E}_{Pers}(P)$, alors nous savons qu'il existe une solution Pareto-optimale s' de la PO-instance $(\mathcal{F}, I, O)$ telle que $Weight(s') = Weight(s)$.

Proposition 4.2. *Étant donné une PO-instance $P = (\mathcal{F}, I, O)$ avec $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$, la longueur de toute solution Pareto-optimale est inférieure ou égale à $min(|\mathcal{S}| - |I|, |\mathcal{A}|)$.*

Pour une PO-instance $P = (\mathcal{F}, I, O)$ avec $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$, nous utilisons $LB(P)$ pour noter la valeur $min(|\mathcal{S}| - |I|, |\mathcal{A}|)$. En outre, en utilisant la proposition 4.2, nous pouvons aisément obtenir une borne sur le poids. Effectivement, cette borne peut être définie comme la somme des poids des arguments appartenant à un ensemble de $LB(P)$ arguments ayant les plus grandes valeurs pour le poids. Plus précisément, le poids de chaque solution Pareto-optimale de P est inférieur ou égal à $\sum_{a \in \mathcal{A}'} W(a)$, où $\mathcal{A}' \subseteq \mathcal{A}$, $|\mathcal{A}'| = LB(P)$ et, pour tout $a' \in \mathcal{A} \setminus \mathcal{A}'$, $W(a') \leqslant min\{W(a) \mid a \in \mathcal{A}'\}$. Nous utilisons $WB(P)$ pour noter la borne $\sum_{a \in \mathcal{A}'} W(a)$.

Afin de présenter notre approche pour le calcul de solutions Pareto-optimales, nous proposons une formulation en MaxSAT partiel pondéré permettant de calculer une solution d'une PO-instance de longueur minimale. Soit $P = (\mathcal{F}, I, O)$ une PO-instance avec $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$. Nous utilisons $\mathcal{E}^{Length}_{Pers}(P)$ pour noter la formulation permettant de calculer une solution de longueur minimale. La partie dure est exactement la même que celle de $\mathcal{E}_{Pers}(P')$, où $P' = (\mathcal{F}, I, O, LB(P), \infty)$. Quant à la partie souple, elle est définie comme suit :

$$1 : \neg q_a \text{ pour chaque } a \in \mathcal{A} \tag{4.13}$$

En effet, en utilisant le fait que la partie dure est la même que celle de $\mathcal{E}_{Pers}(P')$, on sait que chaque solution de $\mathcal{E}^{Length}_{Pers}(P)$ correspond à une solution de P. De plus, en utilisant la proposition 4.2, la partie dure de $\mathcal{E}^{Length}_{Pers}(P)$ est cohérente si et seulement si P admet une solution. Nous pouvons facilement constater dans le cas de cette formulation que la partie souple permet de réduire le nombre d'arguments utilisés.

Une approche simple pour trouver certaines solutions Pareto-optimales d'une PO-instance donnée P consiste à employer deux formulations en PW-MaxSAT. En effet, nous débutons dans un premier temps par une solution s_0 de $\mathcal{E}^{Length}_{Pers}(P)$. Ensuite, chaque solution de $\mathcal{E}_{Pers}(P')$ avec $P' = (\mathcal{F}, I, O, Leng(s_0), \infty)$ est une solution Pareto-optimale de P. Effectivement, soit s une solution de P obtenue grâce à $\mathcal{E}_{Pers}(P')$. Alors, pour toute solution s' de P avec $Leng(s') \leqslant Leng(s_0)$, on obtient $Weight(s) \leqslant Weight(s')$. De plus, on sait que $Leng(s_0)$ est la plus petite valeur possible pour la longueur. En conséquence, la solution s n'est dominée par aucune autre solution.

Une approche similaire pour trouver des solutions Pareto-optimales consiste à calculer dans un premier temps la plus petite valeur possible pour le poids en utilisant notre formulation $\mathcal{E}_{Pers}(P)$, et ensuite, faire appel à une formulation permettant de trouver une des plus petites séquences ayant le poids calculé précédemment. Soit $P = (\mathcal{F}, I, O)$ une PO-instance. Premièrement, une solution s_0 de $\mathcal{E}_{Pers}(P')$ est calculée avec $P' = (\mathcal{F}, I, O, LB(P), \infty)$. Nous savons que $Weight(s_0)$ est la plus petite valeur possible pour le poids quant aux solutions de P. Après cela, nous utilisons la formulation en PW-MaxSAT $\mathcal{E}^{weight}_{Pers}(P, Weight(s_0))$ pour trouver une solution Pareto-optimale, qui est définie en ajoutant la contrainte suivante à la partie dure de $\mathcal{E}^{length}_{Pers}(P)$:

$$\sum_{a \in \mathcal{A}} W(a) * q_a \leqslant Weight(s_0) \tag{4.14}$$

Cette contrainte permet clairement de garantir que chaque solution de la formulation $\mathcal{E}_{Pers}^{weight}(P, Weight(s_0))$ possède la plus petite valeur possible pour le poids. En outre, pour toute solution s de $\mathcal{E}_{Pers}^{weight}(P, Weight(s_0))$, il n'y pas de solution de P qui à la fois a le poids $Weight(s_0)$ et est plus courte que s. Par conséquent, les solutions de $\mathcal{E}_{Pers}^{weight}(P, Weight(s_0))$ sont forcément Pareto-optimales.

4.4 Conclusion

Nous avons consacré ce chapitre à la description de nos travaux en rapport avec l'optimisation dans des cadres reposant sur SAT. Nous avons notamment présenté un problème d'optimisation ayant comme base SAT et intégrant des formes de préférence sur les modèles. Celui-ci a été utilisé dans nos travaux pour la résolution de différents problèmes en fouille de données [JSS13c, JSS17]. Nous avons aussi présenté un cadre pour la persuasion automatique, où nous utilisons la modélisation dans un problème d'optimisation dérivé de SAT [Sal19c]. Nous avons également abordé dans le contexte de ce cadre des aspects intéressants liés à l'optimisation bi-objective. Concernant ce dernier point, il convient de préciser que nous avons employé dans [JKSS16] l'optimisation bi-objective en lien avec SAT dans le contexte d'un problème en fouille de données.

Chapitre 5

Au-delà de la cohérence en logique propositionnelle

Ce chapitre est dévolu à nos travaux sur le raisonnement en présence de l'incohérence en logique propositionnelle. Nous abordons dans un premier temps nos contributions ayant trait aux mesures de l'incohérence. Nous présentons ensuite notre approche pour la définition de relations de conséquence logique paracohérentes fondée sur l'utilisation de la notion de fonction de conséquence. Nous présentons enfin une application des mesures de l'incohérence en fouille de données en considérant la tâche de regroupement. Une grande partie des contributions présentées ici viennent de nos travaux dans [ARSO15, ASOR17, BJSS17a, Sal19b].

5.1 Mesures de l'incohérence

En logique classique, le principe d'explosion énonce que toute formule peut être déduite à partir de l'incohérence, ce qui signifie que nous avons besoin d'approches autres que l'inférence classique pour raisonner sous l'incohérence. Dans ce contexte, les logiques paracohérentes, l'argumentation et la révision des croyances font partie des approches les plus étudiées pour le raisonnement sous l'incohérence. Dans la même veine, les mesures de l'incohérence ont été proposées comme une contrepartie naturelle dans le cas de l'incohérence des mesures d'information. De nombreuses mesures de ce type ont été proposées dans la littérature (voir par exemple [KLM03, HK10, GH13, JMR+16, Thi16, ASOR17, DGHK18, Thi18]), et leur utilité a été démontrée pour de multiples applications dans des domaines tels que le e-commerce [CZZ04], les bases de données [MPS+07] et les systèmes multi-agents [HPW14]. Nous avons également proposé une application dans le cadre du raisonnement qualitatif spatio-temporel [CRS16].

5.1.1 Approche de définition fondée sur des postulats

Une *mesure de l'incohérence* est simplement définie comme une fonction associant à des bases de formules des nombres réels non négatifs. Rappelons qu'une base dans ce contexte est un ensemble fini de formules propositionnelles. Dans [HK10], les auteurs

proposent une approche fondée sur des postulats saisissant des aspects rationnels pour la définition des mesures de l'incohérence. Ils proposent notamment les postulats suivants (I est une mesure de l'incohérence) :

- pour toute base finie B, $I(B) = 0$ si et seulement si B est cohérente (*Cohérence*) ;
- pour toute paire de bases finies B et B' avec $B \subseteq B'$, $I(B) \leqslant I(B')$ (*Monotonie*) ;
- pour toute base finie B et pour toute formule ϕ libre dans B, $I(B) = I(B \setminus \{\phi\})$ (*Formules libres*) ;
- pour toute base finie B et pour toutes formules cohérentes ϕ et ψ telles que $\phi \vdash \psi$, $I(B \cup \{\phi\}) \geqslant I(B \cup \{\psi\})$ (*Dominance*).

Rappelons qu'une *formule est libre* dans une base si elle n'appartient à aucun des sous-ensembles minimaux incohérents. Un sous-ensemble d'une base est *minimal incohérent* s'il est incohérent et il n'est inclus dans aucun autre sous-ensemble incohérent.

Définition 5.1 (Sous-ensemble minimal incohérent)**.** *Un sous-ensemble B' de B est* minimal incohérent *si (i) B' est incohérent, et (ii) pour tout sous-ensemble propre $B'' \subset B'$, B'' est cohérent.*

Nous utiliserons $MI(B)$ pour noter l'ensemble de tous les sous-ensembles minimaux incohérents de B.

Le postulat (*Cohérence*) signifie qu'une mesure de l'incohérence doit permettre de faire la distinction entre les bases cohérentes et celles incohérentes. Quant au postulat (*Monotonie*), il signifie que la quantité de contradiction ne diminue pas en ajoutant de nouvelles formules. Le postulat (*Formules libres*) signifie que les formules libres n'influencent pas la quantité de contradiction. Enfin, le postulat de (*Dominance*) stipule que la quantité de contradiction ne diminue pas en considérant une formule plus forte qu'une formule existante dans la base.

Il y a dans la littérature, notamment dans [Bes14] et dans nos travaux [ASOR17], des objections concernant le postulat (*Dominance*). Nous avons dans ce contexte proposé un affaiblissement permettant d'éviter des aspects négatifs en lien avec ce postulat :

- pour toute base finie B et pour toutes formules cohérentes ϕ et ψ telles que $\phi \notin B$ et $\phi \vdash \psi$, $I(B \cup \{\phi\}) \geqslant I(B \cup \{\psi\})$ (*Aff-Dominance*).

La condition ajoutée $\phi \notin B$ a comme conséquence $\phi \notin B \cup \{\psi\}$ dans le cas où $\phi \neq \psi$, ce qui permet d'exprimer que ϕ est remplacée par ψ et éviter par la même occasion que ce postulat soit en conflit avec le postulat (*Monotonie*).

Par ailleurs, parmi les postulats les plus étudiés, nous pouvons également mentionner les deux suivants introduits respectivement dans [Thi13] et [HK10] :

- pour toutes bases finies B et B' avec $B \cap B' = \emptyset$, $I(B \cup B') \geqslant I(B) + I(B')$ (*Super-Additivité*) ;
- pour toutes bases finies B et B' avec $MI(B) \cap MI(B') = \emptyset$ et $MI(B \cup B') = MI(B) \cup MI(B')$, $I(B \cup B') = I(B) + I(B')$ (*MI-Additivité*).

Dans nos travaux dans [ASOR17], nous avons mis en lumière des problèmes d'incompatibilité entre certains postulats, notamment ceux décrits dans cette section. Nous avons en particulier démontré la proposition suivante.

Proposition 5.1. *Les systèmes de postulats suivants ne peuvent être vérifiés par aucune mesure de l'incohérence :*

— *(Cohérence), (Dominance) et (MI-Additivité) ;*
— *(Cohérence), (Aff-Dominance) et (MI-Additivité) ;*
— *(Cohérence), (Dominance) et (Super-Additivité).*

Néanmoins, ce qui peut plaider en faveur de notre variante *(Aff-Dominance)* est la proposition suivante.

Proposition 5.2. *Le système de postulats suivant admet une mesure de l'incohérence : (Cohérence), (Monotonie), (Aff-Dominance), (Formules libres) et (Super-Additivité).*

En outre, nous avons montré dans [ASOR17] que les mesures de l'incohérence vérifiant le système fort de postulats proposé dans [Bes14] permettent uniquement de distinguer les bases cohérentes de celles incohérentes. Rappelons que ce système a principalement comme base la propriété nommée *orientation de subsomption* reposant sur une notion abstraite de conflit primitif. La proposition suivante présente notre résultat plus formellement.

Proposition 5.3. *Une mesure de l'incohérence I satisfait les postulats du système fort si et seulement si elle est définie comme suit :*

$$I(K) = \begin{cases} 0 & si\ K \nvdash \bot \\ n & sinon \end{cases}$$

où n est une constante différente de 0.

5.1.2 Une mesure de l'incohérence

Au nombre des mesures de l'incohérence les plus connues dans la littérature, nous pouvons citer la mesure I_{MI} définie comme le nombre des sous-ensembles minimaux incohérents [HK10]. Nous avons également introduit des mesures fondées sur la notion de sous-ensemble minimal incohérent dans [JMR+15, JMR+16].

Nous décrirons dans ce qui suit une mesure de l'incohérence que nous avons introduite dans [ARSO15, ASOR17] et qui est fondée, quant à elle, sur la notion de sous-ensemble maximal cohérent.

Définition 5.2 (Sous-ensemble maximal cohérent)**.** *Un sous-ensemble B' de B est* maximal cohérent *si (i) B' est cohérent, et (ii) pour tout $B'' \subseteq B$ avec $B' \subset B''$, B'' est incohérent.*

Nous utiliserons $MC(B)$ pour noter l'ensemble de tous les sous-ensembles maximaux cohérents de B.

Dans ce qui suit, nous nous restreignons aux bases qui ne contiennent que des formules cohérentes.

Étant donné une base de formules B, un sous-ensemble $\mathcal{C}$ de $MC(B)$ est appelé une *MC-couverture* de B si on a $\bigcup_{B' \in \mathcal{C}} B' = B$. Une MC-couverture est dite *normale* si elle n'est un sous-ensemble propre d'aucune autre MC-couverture. En outre, étant donné une base de formules cohérentes B, nous notons $\lambda(B)$ la valeur maximale dans l'ensemble $\{|\bigcap_{B' \in \mathcal{C}} B'| \mid \mathcal{C}$ est une MC-couverture de $B\}$.

Notre mesure I_{MCC} est définie comme suit :

$$I_{MCC}(B) = |B| - \lambda(B)$$

L'idée principale derrière notre définition est double. Premièrement, l'incohérence est quantifiée en considérant que toutes les formules sont possibles. Cela explique pourquoi nous utilisons la notion de MC-couverture. Deuxièmement, une base avec des sous-ensembles maximaux cohérents partageant beaucoup de formules doit se voir attribuer une valeur de l'incohérence inférieure à celle d'une base avec des sous-ensembles maximaux cohérents partageant un petit nombre de formules. Intuitivement, en prenant en compte les formules partagées entre les sous-ensembles maximaux cohérents, notre objectif est de capturer les formules impliquées dans un petit nombre de conflits.

Nous pouvons aisément vérifier que la mesure de l'incohérence I_{MCC} satisfait les trois postulats (*Cohérence*), (*Monotonie*) et (*Formules libres*). Cela dit, elle ne vérifie pas le postulat (*Dominance*) mais elle satisfait à la place le postulat (*Aff-Dominance*). Il est intéressant de noter que nous avons aussi démontré que I_{MCC} satisfait d'autres postulats importants proposés dans la littérature. De plus, nous avons proposé une interprétation épistémique de cette mesure en utilisant la logique modale S5.

5.2 Fonctions de conséquence

L'inférence paracohérente est l'une des approches centrales pour raisonner en présence de l'incohérence. Nous décrivons ici un cadre intuitif et flexible permettant de définir une variété de relations de conséquence logique paracohérentes. Nous introduisons dans ce contexte la notion de fonction de conséquence, qui est utilisée pour associer une valeur, appelée degré de conséquence, à chaque paire composée d'une base de croyances et d'une formule. Nous proposons des postulats associés à cette notion afin de saisir des aspects intéressants liés au raisonnement en présence de l'incohérence. Pour utiliser les fonctions de conséquence dans la définition de relations paracohérentes, l'idée principale consiste à utiliser un seuil de degré de conséquence pour la sélection des formules informatives. Notre cadre permet en particulier de définir des relations de conséquence paracohérenctes bien connues.

5.2.1 Base de croyances

Dans la littérature, une base de croyances est généralement définie comme un ensemble fini de formules. Nous utilisons ici une généralisation de cette notion en divisant une base de croyances en deux parties : la partie certaine et la partie possible. Intuitivement, la partie certaine contient les formules considérées comme des connaissances ne pouvant pas être remises en cause, tandis que la partie possible contient celles considérées comme éventuellement vraies. La partie certaine peut notamment contenir des contraintes d'intégrité.

Définition 5.3 (Base de croyances)**.** *Une* base de croyances *est un couple* $[\Gamma, \Delta]$*, où* Γ *et* Δ *sont des ensembles finis de formules et* $\Gamma \nvdash \bot$ *(*Γ *est cohérent). L'ensemble* Γ *est appelé* la partie certaine *et l'ensemble* Δ la partie possible.

Nous utilisons $\mathcal{BC}_{\mathsf{Form}}$ pour noter l'ensemble des bases de croyances.

En un sens, notre définition de base de croyances peut être vue comme un cas particulier de la notion de base de croyances stratifiée proposée dans [BDP96].

Exemple 5.1. *Considérons les énoncés suivants :*
- *Cette personne est une femme (F).*
- *Cette personne est un homme (H).*
- *Si cette personne est une femme, alors elle est Alice (A).*
- *Si cette personne est un homme, alors il s'agit de Bob (B) ou John (J).*

Ces énoncés peuvent être représentés par l'ensemble de formules suivant : $\Delta = \{F, H, F \rightarrow A, H \rightarrow (B \vee J)\}$*, qui correspond à la partie possible. En outre, il existe clairement d'autres informations connues et liées à la partie possible, comme celles dans* $\Gamma = \{\neg F \vee \neg H, \neg B \vee \neg J, A \rightarrow F, (B \vee J) \rightarrow H\}$*. Par exemple, la formule* $\neg B \vee \neg J$ *signifie qu'une personne ne peut pas être à la fois Bob et John. L'ensemble* Γ *peut être vu comme la partie certaine de la base.*

Une base de croyances $[\Gamma, \Delta]$ est dite *incohérente* si $\Gamma \cup \Delta \vdash \bot$. De plus, nous généralisons la relation de conséquence logique $\vdash$ à $\mathcal{BC}_{\mathsf{Form}}$ comme suit : $[\Gamma, \Delta] \vdash \phi$ si et seulement si $\Gamma \cup \Delta \vdash \phi$.

Dans la définition des fonctions de conséquence, nous avons besoin des notions suivantes qui sont des adaptations de notions vues précédemment à la structure de base de croyances que nous prenons ici en compte.

Définition 5.4 (Sous-ensemble cohérent maximal)**.** *Étant donné une base de croyances* $B = [\Gamma, \Delta]$*, un ensemble de formules* M *est un* sous-ensemble cohérent maximal *de* B *si (i)* $M \in MC(\Gamma \cup \Delta)$ *et (ii)* $\Gamma \subseteq M$*.*

Dans la précédente définition, nous exigeons l'inclusion de la partie certaine pour tenir compte du fait que la vérité des formules de cette partie ne peut être remise en question.

Étant donné une base de croyances B, nous utilisons $NMC(B)$ pour noter l'ensemble des sous-ensembles cohérents maximaux de B.

Définition 5.5 (Formule libre)**.** *Étant donné une base de croyances* $B = [\Gamma, \Delta]$*, une formule* $\phi \in \Gamma \cup \Delta$ *est* libre *dans* B *si* $\phi \in \bigcap_{M \in NMC(B)} M$*.*

Nous utilisons $Libres(B)$ pour faire référence à l'ensemble des formules libres de B.

5.2.2 Définition par des postulats

Nous introduisons dans cette section la notion de fonction de conséquence en tant qu'outil d'analyse de la vérité d'une conclusion par rapport à une base de croyances. Plutôt que de considérer toutes les formules comme des conclusions au même niveau d'une base de croyances incohérente, les fonctions de conséquence fournissent plus d'informations, en particulier un ordre induit entre les conclusions pouvant être utilisé dans un processus de sélection. Nous utilisons des postulats de rationalité pour formaliser différents aspects liés au raisonnement en présence de l'incohérence.

Une fonction de conséquence associe un *degré de conséquence* à une formule par rapport à une base de croyances. Un degré de conséquence peut être interprété comme

une valeur de vérité associée au fait qu'une formule est une conséquence logique d'une base de croyances. Formellement, une *fonction de conséquence* E est définie comme une fonction de $\mathcal{BC}_{\mathsf{Form}} \times \mathsf{Form}$ dans $[0, 1]$. Intuitivement, la valeur 0 représente le fait que l'on est certain de la fausseté d'une conclusion par rapport à une base de croyances, tandis que 1 représente le fait que l'on est certain de la vérité d'une conclusion par rapport à une base de croyances.

Nous définissons maintenant un type particulier de fonctions de conséquence en utilisant des postulats de rationalité capturant certains aspects fondamentaux.

Définition 5.6 (Fonction de conséquence rationnelle)**.** *Une fonction de conséquence E est dite* rationnelle *si elle satisfait les postulats suivants, pour toute base de croyances $B = [\Gamma, \Delta]$ et $\phi, \psi \in \mathsf{Form}$:*

- *si $\Gamma \vdash \phi$ alors $E(B, \phi) = 1$ (Conclusion vraie) ;*
- *si $\Gamma \vdash \phi$ alors $E(B, \neg\phi) = 0$ (Conclusion fausse) ;*
- *si $\phi \vdash \psi$ alors $E(B, \phi) \leqslant E(B, \psi)$ (Conséquence) ;*
- *si $\Gamma \cup \Delta \nvdash \bot$ alors on a si $\Gamma \cup \Delta \vdash \phi$ alors $E(B, \phi) = 1$, sinon $E(B, \phi) = 0$ (Cohérence).*

Le postulat (*Conclusion vraie*) signifie que si une formule est une conséquence logique de la partie certaine, alors elle l'est aussi de la base de croyances. De manière duale, (*Conclusion fausse*) signifie que si une formule est une conséquence logique de la partie certaine, alors sa négation ne peut pas être une conséquence de la base de croyances. Le troisième postulat (*Conséquence*) exprime qu'une formule logiquement plus faible ne peut être moins vraie. Nous utilisons ce postulat pour saisir le fait que les conséquences logiques d'une formule sont au moins aussi informatives que cette formule. Le dernier postulat (*Cohérence*) garantit que le raisonnement classique est préservé dans le cas des bases de croyances cohérentes.

Exemple 5.2. *Les deux fonctions de conséquence suivantes sont clairement rationnelles :*

$$E_{basique}([\Gamma, \Delta], \phi) = \begin{cases} 1 & si\ \Gamma \vdash \phi\ ou\ (\Gamma \cup \Delta \nvdash \bot\ et\ \Gamma \cup \Delta \vdash \phi) \\ 0 & sinon \end{cases}$$

$$E_{libres}(B, \phi) = \begin{cases} 1 & si\ Libres(B) \vdash \phi \\ 0 & sinon \end{cases}$$

En utilisant le fait que les formules qui sont des conséquences logiques de la partie certaine sont considérées comme vraies, il peut être intéressant d'ajouter le postulat suivant dans certains cas pour toute base de croyances $B = [\Gamma, \Delta]$ et toute formule $\phi \in \mathsf{Form}$:

— $E(B, \phi) = E([\Gamma, \Delta \setminus \{\psi \in \Delta \mid \Gamma \vdash \psi\ ou\ \Gamma \vdash \neg\psi\}], \phi)$ (*Affaiblissement*).

Ce postulat stipule que les formules dont on est certain qu'elles sont vraies, comme celles dont on est certain qu'elles sont fausses, peuvent être supprimées de la partie possible sans impacter le degré de conséquence. Nous n'avons pas considéré ce postulat dans notre définition de la notion de fonction de conséquence rationnelle car les formules supprimées peuvent dans certains cas être significatives. Considérons par exemple la base de croyances suivante décrivant des préférences par rapport à l'achat d'une voiture : $B = [\{Color\}, \{\neg Color \wedge (Price \wedge Speed)\}]$. Dans la base précédente, $Color$, $Price$

et *Speed* sont utilisés pour exprimer les critères de choix relatifs à respectivement la couleur, le prix et la vitesse de la voiture. Par exemple, la présence de *Color* dans la partie certaine signifie que le choix de la couleur doit obligatoirement être pris en compte. Clairement, la formule dans la partie certaine a comme conséquence la négation de celle dans la partie possible. Cependant, la sous-formule $Price \wedge Speed$ de la formule possible n'est pas en conflit avec la formule nécessaire et elle peut en conséquence être prise en compte dans le choix de la voiture.

Nous pouvons également considérer les postulats de rationalité supplémentaires suivants pour formaliser certains aspects intéressants, en relation notamment avec la notion de sous-ensembles cohérents maximaux. Pour toute base de croyances $B = [\Gamma, \Delta]$ et pour toutes formules $\phi, \psi \in \mathsf{Form}$:

- si $Libres(B) \vdash \phi$ et $Libres(B) \nvdash \psi$ alors $E(B, \phi) \geqslant E(B, \psi)$ (*Formules libres*) ;
- si (i) $\exists M \in NMC(B)$ t.q. $M \vdash \phi$ et (ii) $\forall M \in NMC(B)$ on a $M \nvdash \psi$, alors $E(B, \phi) \geqslant E(B, \psi)$ (*ENMC*) ;
- si (i) $\exists M \in NMC(B)$ t.q. $M \vdash \phi$ et (ii) $\forall M \in NMC(B)$ on a $M \nvdash \psi$, alors $E(B, \phi) > E(B, \psi)$ (*Fort-ENMC*) ;
- si (i) $\forall M \in NMC(B)$ on a $M \vdash \phi$ et (ii) $\exists M \in NMC(B)$ t.q. $M \nvdash \psi$, alors $E(B, \phi) \geqslant E(B, \psi)$ (*ANMC*) ;
- si (i) $\forall M \in NMC(B)$ on a $M \vdash \phi$ et (ii) $\exists M \in NMC(B)$ t.q. $M \nvdash \psi$, alors $E(B, \phi) > E(B, \psi)$ (*Fort-ANMC*).

Les postulats précédents donnent de manière générale une préférence aux conséquences issues des sous-ensembles maximaux cohérents.

Exemple 5.3. *Nous proposons ici une fonction de conséquence, notée E_{NMC}, reposant sur la notion de sous-ensemble cohérent maximal :*

$$E_{NMC}([\Gamma, \Delta], \phi) = \frac{|\{M \in NMC([\Gamma, \Delta]) \mid M \vdash \phi\}|}{|NMC([\Gamma, \Delta])|}$$

Par exemple, nous avons $E_{NMC}([\{\neg p \vee \neg q\}, \{p, \neg p\}], \neg q) = \frac{1}{2}$, car il y a deux sous-ensembles cohérents maximaux $M_1 = \{\neg p \vee \neg q, p\}$ et $M_2 = \{\neg p \vee \neg q, \neg p\}$ et uniquement M_1 a comme conséquence $\neg q$.

Proposition 5.4. *E_{NMC} est une fonction de conséquence rationnelle vérifiant les postulats (Formules libres), (Fort-ENMC) et (Fort-ANMC).*

5.2.3 Relations de conséquence paracohérentes

Nous proposons ici un cadre pour définir des relations de conséquence paracohérentes fondées sur l'utilisation des fonctions de conséquence (FC). L'idée principale consiste à utiliser un seuil sur les degrés de conséquence pour la sélection des formules informatives. Nous montrons en particulier que notre approche permet de définir des relations de conséquence paracohérentes existantes.

Définition 5.7 (Relation de conséquence fondée sur FC)**.** *Étant donné une fonction de conséquence E et une valeur $v \in [0, 1]$, on définit la relation de conséquence $\vdash_E^{v,\geqslant}$ (resp. $\vdash_E^{v,>}$) sur $\mathcal{BC}_{\mathsf{Form}} \times \mathsf{Form}$ comme suit : $B \vdash_E^{v,\geqslant} \phi$ (resp. $B \vdash_E^{v,>} \phi$) si et seulement si $E(B, \phi) \geqslant v$ (resp. $E(B, \phi) > v$).*

Dans les cas des FC rationnelles, nous obtenons la proposition suivante.

Proposition 5.5. *Soit E une FC rationnelle et $v \in [0,1]$ et $v' \in [0,1]$ avec $v \neq 0$ et $v' \neq 1$. Alors, on a les propriétés suivantes pour toute base $B = [\Gamma, \Delta] \in \mathcal{BC}_{\mathsf{Form}}$ et pour toute relation $\vdash_E \in \{\vdash_E^{v,\geqslant}, \vdash_E^{v',>}\}$:*

1. *$\forall \phi \in \mathsf{Form}$, si $\Gamma \vdash \phi$, alors on a $B \vdash_E \phi$ et $B \nvdash_E \neg\phi$;*
2. *si $\Gamma \cup \Delta \nvdash \bot$, alors $\forall \phi \in \mathsf{Form}$, $B \vdash_E \phi$ si et seulement si $\Gamma \cup \Delta \vdash \phi$;*
3. *$\forall \phi, \psi \in \mathsf{Form}$, si $B \vdash_E \phi$ et $\phi \vdash \psi$, alors $B \vdash_E \psi$.*

Intéressons-nous maintenant à certaines relations paracohérentes existantes. Il importe de noter que nous adaptons ces relations à notre définition de la notion de base de croyances. Considérons tout d'abord la relation $\vdash_{Libres}$ en la définissant d'une manière similaire à la relation introduite dans [BDP95, BDP96] : $B \vdash_{Libre} \phi$ si et seulement si $Libres(B) \vdash \phi$. Maintenant, définissons les relations $\vdash_{ANMC}$ et $\vdash_{ENMC}$ comme dans [RM70] : $B \vdash_{ANMC} \phi$ si et seulement si $\forall M \in NMC(B)$, $M \vdash \phi$; $B \vdash_{ENMC} \phi$ si et seulement si $\exists M \in NMC(B)$ tel que $M \vdash \phi$.

Rappelons que E_{Libres} et E_{NMC} sont définies dans les exemples 5.2 et 5.3 respectivement. La proposition suivante montre que $\vdash_{Libres}$, $\vdash_{ANMC}$ et $\vdash_{ENMC}$ peuvent être définies en utilisant E_{libres} et E_{NMC}.

Proposition 5.6. *Les propriétés suivantes sont vérifiées :*

- *$\forall B \in \mathcal{BC}_{\mathsf{Form}}$ et $\forall \phi \in \mathsf{Form}$, $B \vdash_{Libres} \phi$ si et seulement si $B \vdash_{E_{libres}}^{1,\geqslant} \phi$;*
- *$\forall B \in \mathcal{BC}_{\mathsf{Form}}$ et $\forall \phi \in \mathsf{Form}$, $B \vdash_{ANMC} \phi$ si et seulement si $B \vdash_{E_{NMC}}^{1,\geqslant} \phi$;*
- *$\forall B \in \mathcal{BC}_{\mathsf{Form}}$ et $\forall \phi \in \mathsf{Form}$, $B \vdash_{ENMC} \phi$ si et seulement si $B \vdash_{E_{NMC}}^{0,>} \phi$.*

Il est clair que les relations paracohérentes fondées sur les FC peuvent conduire à avoir comme conséquences des formules contradictoires à partir d'une même base de croyances. Afin d'éviter ce problème, nous pouvons exiger la propriété suivante :

- $E(B, \phi \wedge \psi) = min(E(B,\phi), E(B,\psi))$ (*Conjonction*).

En effet, cette propriété permet d'obtenir la proposition suivante.

Proposition 5.7 (Adjonction)**.** *Soit E une FC qui satisfait la propriété (Conjonction), $v \in [0,1]$ et $v' \in [0,1]$ avec $v \neq 0$ et $v' \neq 1$. Alors, pour toute base $B \in \mathcal{BC}_{\mathsf{Form}}$ et pour toutes formules $\phi, \psi \in \mathsf{Form}$, si $B \vdash_E \phi$ et $B \vdash_E \psi$, alors on a $B \vdash_E \phi \wedge \psi$ pour $\vdash_E \in \{\vdash_E^{v,\geqslant}, \vdash_E^{v',>}\}$.*

Proposition 5.8 (Non-Contradiction)**.** *Soit E une FC rationnelle qui satisfait la propriété (Conjonction), $v \in [0,1]$ et $v' \in [0,1]$ avec $v \neq 0$ et $v' \neq 1$. Alors, pour toute base $B \in \mathcal{BC}_{\mathsf{Form}}$ et pour tout ensemble fini $S \subseteq \{\phi \in \mathsf{Form} \mid B \vdash_E \phi\}$, on a $S \nvdash \bot$ pour $\vdash_E \in \{\vdash_E^{v,\geqslant}, \vdash_E^{v',>}\}$.*

5.2.4 Liens avec les mesures de l'incohérence

Comme nous l'avons vu précédemment, les mesures de l'incohérence concernent des ensembles de formules. Il est aisé d'adapter les postulats de ces mesures à notre structure de base de croyances. Par exemple, les postulats (*Cohérence*) et (*Monotonie*) peuvent être adaptés comme suit :

— pour toute base de croyances $B = [\Gamma, \Delta]$, $I(B) = 0$ si et seulement si $\Gamma \cup \Delta \nvdash \bot$;
— pour toute base de croyances $B = [\Gamma, \Delta]$ et pour tous ensembles de formules finis Γ' et Δ' avec $\Gamma \cup \Gamma' \nvdash \bot$, $I(B) \leqslant I([\Gamma \cup \Gamma', \Delta \cup \Delta'])$.

La plupart des mesures de l'incohérence qui ont été proposées dans la littérature utilisent l'ensemble $\mathbb{R}^+$, au lieu de l'intervalle $[0, 1]$. Ainsi, afin d'associer une fonction de conséquence à toute mesure de l'incohérence, nous avons besoin de reformuler les postulats (*Conclusion vraie*) et (*Conséquence*) pour utiliser $\mathbb{R}^+$ à la place de $[0, 1]$:

— pour toute base de croyances $B = [\Gamma, \Delta]$ et pour toutes formules ϕ, ψ, si $\Gamma \vdash \phi$ et $\Gamma \nvdash \psi$, alors $E(B, \phi) \geqslant E(B, \psi)$ (*Conclusion vraie 2*) ;
— pour toute base de croyances $B = [\Gamma, \Delta]$ avec $\Gamma \cup \Delta \nvdash \bot$ et pour toute formule ϕ, si $\Gamma \cup \Delta \vdash \phi$ alors $E(B, \phi) > 0$; et si $\Gamma \cup \Delta \nvdash \phi$ alors $E(B, \phi) = 0$ (*Conséquence 2*).

Présentons maintenant notre approche pour associer une fonction de conséquence à chaque mesure de l'incohérence. Elle consiste à considérer le degré de conséquence d'une formule par rapport à une base de croyances comme la quantité de contradiction introduite par la négation de cette formule dans la base considérée. Plus précisément, étant donné une mesure de l'incohérence I, la fonction de conséquence associée, notée E_I, est définie comme suit :

$$E_I([\Gamma, \Delta], \phi) = I([\Gamma, \Delta' \cup \{EQ(\Delta', \neg\phi)\}]) - I([\Gamma, \Delta'])$$

où $\Delta' = \Delta \setminus \{\psi \in \Delta \mid \Gamma \vdash \psi \textit{ ou } \Gamma \vdash \neg\psi\}$ et $EQ(\Delta', \neg\phi)$ correspond à n'importe quelle formule équivalente à $\neg\phi$ mais qui n'est pas dans Δ'. Il est à noter que $EQ(\Delta', \neg\phi)$ peut être calculée en temps linéaire en utilisant la loi de la double négation : ajouter la double négation $\neg\neg$ jusqu'à l'obtention d'une formule qui n'appartient pas à Δ'.

Nous supprimons de la partie possible les formules appartenant à $\{\psi \in \Delta \mid \Gamma \vdash \psi \textit{ ou } \Gamma \vdash \neg\psi\}$ car nous savons comment les prendre en compte en utilisant le fait que les formules dans Γ ne peuvent pas être remise en cause. En outre, nous utilisons $EQ(\Delta', \neg\phi)$ à la place de $\neg\phi$ pour prendre en compte le fait que $\neg\phi$ peut être dans Δ'. En effet, sans ce remplacement, nous aurions $E_I([\Gamma, \Delta], \phi) = 0$ pout tout $\neg\phi \in \Delta$.

Notons que pour toute mesure de l'incohérence I qui vérifie le postulat (*Monotonie*), et pour toute base B et toute formule ϕ, on a $E_I(B, \phi) \geqslant 0$.

Nos relations paracohérentes utilisent un seuil dans $[0, 1]$ suivant la définition 5.7, ceci est problématique lorsque l'on utilise $\mathbb{R}^+$ à la place de $[0, 1]$. Dans les deux définitions suivantes, nous proposons deux approches permettant d'éviter l'utilisation de seuils de manière explicite. Nous sélectionnons dans la première les formules ayant des degrés de conséquence plus grands que le degré d'une formule donnée, tandis que dans la deuxième, nous sélectionnons les formules dont les degrés de conséquence sont plus grands que ceux de leurs négations.

Définition 5.8. *Étant donné une fonction de conséquence E et une formule ϕ, nous définissons la relation paracohérente $\vdash_E^{\phi,\geqslant}$ (resp. $\vdash_E^{\phi,>}$) comme suit : $B \vdash_E^{\phi} \psi$ si et seulement si $E(B, \psi) \geqslant E(B, \phi)$ (resp. $E(B, \psi) > E(B, \phi)$).*

Définition 5.9. *Étant donné une fonction de conséquence E, nous définissons la relation paracohérente $\vdash_E^{\geqslant}$ (resp. $\vdash_E^{>}$) comme suit : $B \vdash_E^{\geqslant} \phi$ si et seulement si $E(B, \phi) \geqslant E(B, \neg\phi)$ (resp. $E(B, \phi) > E(B, \neg\phi)$).*

5.3 Une application en fouille de données

Nous présentons dans cette section une application de la notion de mesure de l'incohérence en fouille de données. Pour cela, nous nous intéressons au problème de regroupement (en anglais *clustering*) sur des données complexes représentées par des formules propositionnelles. L'intérêt principal derrière l'utilisation de la logique propositionnelle réside dans le fait qu'il s'agit d'un cadre permettant de représenter de manière compacte des entités hétérogènes. En outre, les nombreux systèmes formels de raisonnement pour cette logique peuvent être utilisés pour capturer des connaissances possédant des structures complexes. Nous utiliserons dans ce contexte les mesures de l'incohérence pour l'amélioration des groupes incohérents.

Il est important de préciser que nous nous focalisons dans la description que nous détaillons ici sur la définition du cadre général et non sur les aspects algorithmiques, l'objectif étant de montrer la manière dont nous pouvons intégrer la notion de mesure de l'incohérence dans le problème de regroupement.

5.3.1 Le problème de regroupement

Le regroupement, on parle également de partitionnement de données, est une des principales méthodes utilisées dans l'analyse de données. Il permet d'extraire des structures cachées sous forme de division en groupes dans des bases de données, de telle sorte que les groupes soient homogènes, et cela, en exigeant que les éléments de chaque groupe soient fortement similaires via des critères choisis, mais il est également important qu'il y ait une faible similarité entre groupes distincts. Il est ainsi important de préciser que les mesures de similarité utilisées dans la réalisation de la tâche de regroupement jouent un rôle de premier ordre.

Le problème de regroupement est très étudié du fait de son large champ d'application, comme par exemple en biologie, en analyse d'images ou bien encore en commerce (voir entre autres [AR13]). Les nombreux domaines d'application ont donné lieu à une variété de types de données, tels que les transactions, les séquences, les données textuelles, les graphes, ce qui a notamment conduit à la proposition de plusieurs techniques dans la réalisation des regroupements (voir [Ber06] pour une vue générale). Parmi ces types de données, les données symboliques [dCCL09, dSdC04, GD94] sont particulièrement appropriées pour traiter des objets complexes (voir par exemple [BD12, Boc00, DE03]). Nous pouvons de plus mentionner le regroupement conceptuel proposé dans [Mic80], qui est une tâche d'apprentissage automatique traitant d'un ensemble de descriptions d'objets (événements, faits, observations, etc.) et produisant un schéma de regroupement les concernant. Le regroupement conceptuel non seulement réalise la partition des données, mais génère des groupes pouvant être décrits de manière conceptuelle. Un des points intéressants relatifs aux tâches de regroupement conceptuel et symbolique réside dans le fait qu'elles permettent de traiter des objets complexes et hétérogènes.

5.3.2 Représentation par formules logiques

Nous décrivons dans cette section le problème de regroupement sur des données représentées par des formules propositionnelles. Afin d'illustrer l'intérêt de ce problème, nous débutons notre présentation par un exemple simple.

Exemple 5.4. *Il s'agit ici d'un exemple d'organisation d'un dîner de mariage. L'objectif est de proposer une distribution des invités sur les tables disponibles. Considérons que nous sommes en présence de m invités et n tables. Pour chaque invité* $i \in 1..m$*, nous associons une variable propositionnelle distincte* p_i*. Dans ce contexte, les préférences de chaque invité i sont exprimées par une formule propositionnelle notée* ϕ_i*. Par exemple, la formule* $p_1 \wedge \neg p_2 \wedge (p_3 \leftrightarrow p_4)$ *peut être utilisée pour exprimer les préférences de l'invité* 1 *qui ne veut pas être avec l'invité* 2 *et qui accepte d'être avec l'invité* 3 *si et seulement si l'invité* 4 *est également présent à la même table. De cette façon, la division de l'ensemble de formules* $B = \{\phi_1, \ldots, \phi_m\}$ *en au maximum n groupes correspond à une distribution des m invités autour des n tables disponibles. En outre, il est clair que pour respecter les préférences de tous les invités, tous les groupes doivent correspondre à des ensembles cohérents de formules. Par ailleurs, il est souvent utile de prendre en compte certaines préférences qui doivent être respectées par tous les groupes. Par exemple, dans le cas de notre exemple, Nous pouvons utiliser la formule* $(\bigvee_{i \in F} p_i) \wedge (\bigvee_{i \in H} p_i) \wedge (\sum_{i=1}^{m} p_i \leqslant 5)$*, où F représente l'ensemble des femmes parmi les invités et H celui des hommes, pour exprimer le fait que chaque table doit être mixte et qu'elle ne peut contenir plus de cinq invités. Nous appellerons* formules globales *ce type de formules. Il convient de préciser que les formules globales peuvent être vues comme des contraintes d'intégrité.*

Rappelons qu'une *partition* P d'un ensemble S est un ensemble de sous-ensembles de S tel que (i) $\forall S' \in P$, $S' \neq \emptyset$, (ii) $\bigcup_{S' \in P} S' = S$, et (iii) $\forall S', S'' \in P$ avec $S' \neq S''$, on a $S' \cap S'' = \emptyset$. Une *n-partition* P est une partition telle que $|P| = n$. De plus, étant donné un ensemble fini S, nous utilisons $\mathcal{P}(S)$ afin de noter l'ensemble de ses sous-ensembles.

Définition 5.10 (Partition ϕ-minimale)**.** *Soient B une base de formules cohérentes finie et non vide, P une n-partition de B et* ϕ *une formule. La partition P est dite* ϕ-minimale *si pout toute n-partition* P' *de B, on a* $|\{B' \in P' : B' \cup \{\phi\} \vdash \bot\}| \geqslant |\{B' \in P : B' \cup \{\phi\} \vdash \bot\}|$.

Nous définissons maintenant le concept de regroupement de formules à travers une approche ayant comme base des postulats pour la prise en compte d'aspects rationnels importants.

Définition 5.11 (Regroupement de formules)**.** *Étant donné une base de formules propositionnelles cohérentes finie et non vide B, un entier naturel non nul n et une formule cohérente* ϕ*, appelée* formule globale*, un* (n, ϕ)*-regroupement de B est un ensemble* $\mathcal{C} \subseteq \mathcal{P}(B \cup \{\phi\})$ *tel que :*

- $\bigcup_{B' \in \mathcal{C}} B' = B \cup \{\phi\}$ *(Complétude) ;*
- $\phi \in \bigcap_{B' \in \mathcal{C}} B'$ *(Formule globale) ;*
- $|\mathcal{C}| \leqslant n$ *(Borne supérieure) ;*
- *si B possède une partition P t.q.* $|P| \leqslant n$ *et* $Q \cup \{\phi\} \nvdash \bot$ *pour chaque* $Q \in P$*, alors,* $\forall B' \in \mathcal{C}$*,* $B' \nvdash \bot$ *(Partition cohérente) ;*
- $\forall B' \in \mathcal{C}$ *et* $\forall B'' \subset B' \backslash \{\phi\}$ *avec* $B' \vdash \bot$ *et* $B' \backslash B'' \nvdash \bot$*, et* $\forall \{B_1, \ldots, B_m\} \subseteq \mathcal{C} \backslash \{B'\}$ *avec* $|B''| \geqslant m$*, il n'existe pas de m-partition* $\{B''_1, \ldots, B''_m\}$ *de* B'' *t.q.* $B_i \cup B''_i \nvdash \bot$ *pour* $i \in 1..m$ *(Préférence de la cohérence) ;*
- *si* $\exists B' \in \mathcal{C}$ *t.q.* $B' \vdash \bot$*, alors* $|\mathcal{C}| = min(|B|, n)$ *(Nombre de groupes).*

Dans un (n, ϕ)-regroupement, n correspond au nombre maximum de groupes et la formule globale ϕ est utilisée pour représenter la propriété devant être partagée par tous

les groupes. Le postulat (*Complétude*) contraint à ce que toutes les formules de la base et la formule globale soient présentes dans le regroupement. Le postulat (*Formule globale*) exprime simplement que la formule globale doit être dans tous les groupes. Le postulat (*Borne supérieure*) est utilisé pour ne pas dépasser le nombre maximum de groupes fixé. Les postulats (*Partition cohérente*) et (*Préférence de la cohérence*) permettent d'exprimer que la préférence est toujours donnée à la constitution de groupes cohérents. Dans la même veine, le postulat (*Nombre de groupes*) stipule qu'il faut constituer un maximum de groupes en présence de groupes incohérents afin de permettre la réduction des conflits dans les groupes. Ceci peut être vu comme une manière de s'approcher de la cohérence.

Proposition 5.9. *Étant donné une base de formules cohérentes finie B, si P est une n-partition ϕ-minimale alors $\{B' \cup \{\phi\} \mid B' \in P\}$ est un (n, ϕ)-regroupement.*

Une des conséquences importantes de la proposition précédente réside dans le fait qu'elle montre qu'il est toujours possible de construire un (n, ϕ)-regroupements à partir d'une base.

Notons que suivant notre définition de la notion de (n, ϕ)-regroupement, deux groupes distincts peuvent partager en plus de la formule globale d'autres formules. Ceci peut particulièrement être adéquat dans certaines applications. Dans le cas de notre exemple de l'organisation d'un dîner de mariage, l'appartenance d'une formule à des groupes différents signifie simplement que l'invité qui y est associé peut être placé à différentes tables, information qui peut intéresser l'organisateur. Cela dit, il peut être pertinent dans d'autres types d'applications d'éviter des chevauchements entre groupes, excepté bien entendu celui de la formule globale. Dans ce cas, il suffit de considérer la propriété suivante dans la définition des (n, ϕ)-regroupements :

— $\forall B', B'' \in \mathcal{C}$, $B' \cap B'' = \{\phi\}$ (*Indépendance du chevauchement*).

Dans certains cas, il se peut que le fait de limiter le chevauchement à uniquement la formule globale soit trop restrictif sachant que des formules peuvent être des conséquences logiques d'autres formules. Un compromis peut ainsi être l'utilisation des deux propriétés suivantes :

— $\forall B' \in \mathcal{C}$ et $\forall \psi \in B$, si $B' \nvdash \bot$ et $B' \vdash \psi$, alors $\psi \in B'$ (*Conséquence logique*) ;

— $\forall B', B'' \in \mathcal{C}$ avec $B' \neq B''$, $B' \cap B'' \neq \{\phi\}$ si et seulement si $B' \nvdash \bot$, $B'' \nvdash \bot$, $(B' \setminus B'') \cup \{\phi\} \vdash \psi$ et $(B'' \setminus B') \cup \{\phi\} \vdash \psi$ pour tout $\psi \in B' \cap B''$ (*Intersection*).

La première propriété signifie principalement que le raisonnement classique est préservé sous la cohérence en augmentant chaque groupe cohérent avec ses conséquences logiques. Quant à la deuxième propriété, elle permet de restreindre les chevauchements aux conséquences logiques en présence de la cohérence.

Il est important de préciser que les propriétés relatives aux regroupements de formules que nous venons de présenter ne permettent pas de faire de distinction entre groupes cohérents, de même qu'elles ne permettent pas de faire de distinction entre groupes incohérents. En effet, l'unique distinction exprimée par ces propriétés est dans la préférence des groupes cohérents aux groupes incohérents. Des distinctions qualitatives entre groupes cohérents, comme entre groupes incohérents, sont primordiales pour l'obtention de regroupements pertinents, et notamment pour le fonctionnement de différentes approches existantes de regroupement. Par exemple, des mesures de qualité de

groupe peuvent être utilisées dans des approches hiérarchiques afin de choisir les groupes à améliorer, par fusion ou par division.

Dans le cas des groupes cohérents, nous sommes en possession d'un outil fort pour mesurer la qualité d'un groupe qui est la notion de modèle. Il est en effet possible de définir une telle mesure en accordant une préférence à l'augmentation du nombre de modèles liant les membres d'un groupe. Nous pouvons ainsi définir la mesure de qualité d'un groupe B avec l'expression suivante :

$$Q(B) = \frac{|Mods(\bigwedge_{\psi \in B} \psi)|}{|Mods(\bigvee_{\psi \in B} \psi)|}$$

où $Mods(\chi)$ correspond à l'ensemble des modèles de χ. Cette définition peut être vue comme une simple adaptation du coefficient de Jaccard [Jac01].

Afin d'être également en capacité de distinguer les groupes incohérents, nous utiliserons la notion de mesure de l'incohérence comme nous le verrons dans ce qui suit.

5.3.3 Regroupement fondé sur les mesures de l'incohérence

Nous décrivons dans cette section la manière dont la notion de mesure de l'incohérence peut être utilisée dans la tâche de regroupement de formules. L'idée principale consiste à employer cette notion dans l'amélioration des groupes incohérents. Nous introduisons d'abord un type particulier de regroupements pour lequel est utilisée une propriété indiquant la façon dont nous pouvons améliorer les groupes incohérents en utilisant une mesure de l'incohérence. Nous proposons ensuite des propriétés supplémentaires permettant de caractériser des aspects intéressants liés à la quantification des conflits.

Il est important de noter que nous considérons ici uniquement les mesures de l'incohérence vérifiant les postulats *(Cohérence)* et *(Monotonie)*, largement admis dans la littérature.

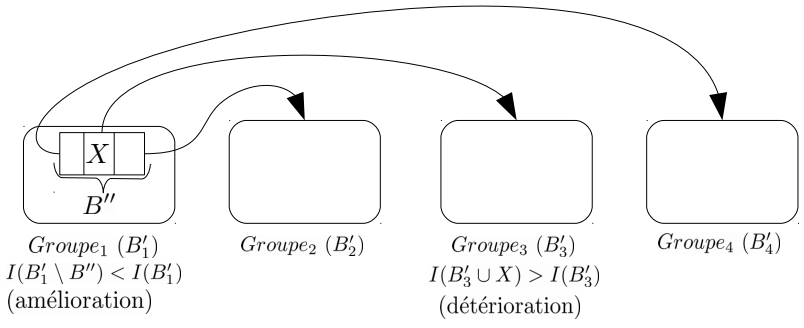


FIGURE 5.1 – Une illustration de la propriété de regroupement I-rationnel

Définition 5.12. *Étant donné une base de formules cohérentes finie et non vide B, une mesure de l'incohérence I, un entier naturel non nul n et une formule ϕ, un (n, ϕ)-regroupement $\mathcal{C}$ de B est dit* I-rationnel *s'il satisfait la propriété suivante :*

$\forall B' \in \mathcal{C}$ *et* $\forall B'' \subset B' \setminus \{\phi\}$ *avec* $I(B' \setminus B'') < I(B')$, *et* $\forall \{B_1, \ldots, B_m\} \subseteq \mathcal{C} \setminus \{B'\}$ *avec* $|B''| \geqslant m$, *il n'existe pas de* m*-partition* $\{B''_1, \ldots, B''_m\}$ *de* B'' *t.q.* $I(B_i \cup B''_i) = I(B_i)$ *pour* $i \in 1..m$.

En d'autres termes, un (n, ϕ)-regroupement est I-rationnel si la quantité de l'incohérence dans un groupe ne peut être réduite en déplaçant certaines formules vers d'autres groupes sans augmenter la quantité de l'incohérence dans au moins un de ces derniers. Cela s'apparente à un équilibre où l'action d'amélioration de tout groupe incohérent, par rapport à une mesure de l'incohérence donnée, implique nécessairement la détérioration d'au moins un autre.

Tout comme la proposition 5.9, la proposition suivante montre qu'il est possible d'obtenir des regroupements I-rationnels en restreignant la recherche aux regroupements construits à partir de partitions ϕ-minimales.

Proposition 5.10. *Étant donné une base de formules cohérentes finie et non vide* B, I *une mesure de l'incohérence et* P *une* n*-partition* ϕ*-minimale vérifiant la propriété suivante : pour toute* n*-partition* ϕ*-minimale* P' *avec* $|S_1| = |S_2|$ *où* $S_1 = \{B' \in P' \mid B' \cup \{\phi\} \vdash \bot\}$ *et* $S_2 = \{B'' \in P \mid B'' \cup \{\phi\} \vdash \bot\}$, $\sum_{B' \in S_1} I(B) \geqslant \sum_{B'' \in S_2} I(B'')$. *On a* $\{B' \cup \{\phi\} \mid B' \in P\}$ *est un* (n, ϕ)*-regroupement* I*-rationnel.*

Exemple 5.5. *Considérons encore l'exemple du dîner de mariage décrit précédemment, mais cette fois-ci l'objectif est la constitution de deux menus de repas. Les préférences des invités sont définies à travers les formules suivantes :* $\psi_1 = \texttt{soupe} \wedge \texttt{poisson}$ *;* $\psi_2 = \texttt{poisson} \wedge \texttt{glace}$ *;* $\psi_3 = \neg\texttt{viande} \wedge \texttt{fromage}$ *;* $\psi_4 = \texttt{salade} \wedge \texttt{viande} \wedge \texttt{fromage}$ *;* $\psi_5 = \neg\texttt{soupe} \wedge \texttt{viande} \wedge \texttt{glace}$. *La formule globale sera utilisée pour décrire le fait que chaque menu contient au plus une entrée, au plus un plat et au plus un dessert. Elle est ainsi définie comme suit :* $\phi = (\sum_{e \in E} e \leqslant 1) \wedge (\sum_{p \in P} p \leqslant 1) \wedge (\sum_{d \in D} d \leqslant 1)$, *où* E, P *et* D *sont les ensembles de respectivement les entrées, les plats et les desserts.*

Nous pouvons facilement constater que l'ensemble $B = \{\psi_1, \psi_2, \psi_3, \psi_4, \psi_5\}$ *ne peut pas être partitionné en deux sous-ensembles cohérents. Donc, chaque* $(2, \phi)$*-regroupement de* B *contient au moins un groupe incohérent. Parmi ces derniers, nous avons les regroupements* $\mathcal{C}_1 = \{\{\phi, \psi_1, \psi_2\}, \{\phi, \psi_3, \psi_4, \psi_5\}\}$ *et* $\mathcal{C}_2 = \{\{\phi, \psi_1\}, \{\phi, \psi_2, \psi_3, \psi_4, \psi_5\}\}$. *Dans* $\mathcal{C}_1$, *le premier groupe est cohérent et représente le menu* `soupe`, `poisson` *et* `glace` *; par contre le second groupe est incohérent, mais toutes ses formules ne rejettent pas* `salade` *et* `fromage`, *en plus, nous pouvons raisonnablement dire que* `viande` *est plus acceptée que* `poisson` *car* `viande` *est une conséquence logique de* ψ_4 *et* ψ_5. *Ainsi, en utilisant* $\mathcal{C}_1$, *nous pouvons proposer les deux menus suivants :* $m_1 =$ `soupe`, `poisson`, `glace` *et* $m_2 =$ `salade`, `viande`, `fromage`.

Considérons maintenant la mesure de l'incohérence I_{MI} *correspondant au nombre des sous-ensembles incohérents minimaux. Le* $(2, \phi)$*-regroupement* $\mathcal{C}_1$ *est* I_{MI}*-rationnel car nous avons* $I_{MI}(\{\phi, \psi_1, \psi_2\}) = 0$ *et* $I_{MI}(\{\phi, \psi_1, \psi_2\} \cup \{\chi\}) > 0$ *pour toute formule* $\chi \in \{\psi_3, \psi_4, \psi_5\}$. *Cependant,* $\mathcal{C}_2$ *ne l'est pas car nous avons* $I_{MI}(\{\phi, \psi_1\}) = I_{MI}(\{\phi, \psi_1, \psi_2\}) = 0$ *et nous avons de plus* $I_{MI}(\{\phi, \psi_2, \psi_3, \psi_4, \psi_5\}) = |\{\{\phi, \psi_2, \psi_3\}, \{\phi, \psi_2, \psi_4\}, \{\phi, \psi_2, \psi_5\}, \{\psi_3, \psi_4\}, \{\psi_3, \psi_5\}, \{\phi, \psi_4, \psi_5\}\}| = 6 > I_M(\{\phi, \psi_3, \psi_4, \psi_5\}) = 3$.

Afin d'aboutir à une forme de réduction globale de la quantité de conflits dans un regroupement, nous pouvons requérir la propriété suivante :

— il n'existe pas de (n, ϕ)-regroupement $\mathcal{C}'$ de B tel que $max\{I(B') \mid B' \in \mathcal{C}\} > max\{I(B') \mid B' \in \mathcal{C}'\}$.

Cette propriété permet de privilégier la répartition des conflits sur plusieurs groupes au lieu de les concentrer en peu de groupes. Nous pouvons généraliser cette propriété en considérant une fonction quelconque f sur les quantités de l'incohérence dans les différents groupes comme suit :

— il n'existe pas de (n, ϕ)-regroupement $\mathcal{C}'$ de B tel que $f\{I(B') \mid B' \in \mathcal{C}\} \triangleleft f\{I(B') \mid B' \in \mathcal{C}'\}$,

où $\triangleleft \in \{<, >\}$. Par exemple, nous pouvons utiliser les fonctions *moyenne* et *somme* avec l'inégalité $>$.

5.4 Conclusion

Dans ce chapitre, nous avons examiné des aspects relatifs à l'utilisation de la logique propositionnelle en présence de l'incohérence. Nous avons donc présenté une partie de nos travaux afférents à la notion de mesure de l'incohérence. Nous avons en particulier décrit de manière détaillée une des mesures de l'incohérence que nous avons introduites, ainsi que certains de nos résultats concernant la compatibilité entre postulats proposés dans la littérature. En outre, nous avons présenté notre cadre pour la définition de relations de conséquence paracohérentes fondé sur les fonctions de conséquence. L'idée principale consiste à utiliser un seuil sur les valeurs fournies par les fonctions de conséquence dans la sélection des conclusions. Nous avons également exposé un cadre pour la réalisation de la tâche de regroupement dans des bases constituées de formules propositionnelles. Nous avons pour cela utilisé une approche reposant sur des postulats pour l'introduction d'un cadre intuitif et simple. Nous avons par la suite présenté la manière dont la notion de mesure de l'incohérence peut être utilisée dans la tâche de regroupement de formules. L'idée consiste à utiliser cette notion dans l'amélioration des groupes incohérents.

Chapitre 6

Conclusion et Perspectives

Ce mémoire résume nos travaux de recherche autour de la modélisation en logique propositionnelle classique. Nous avons à cette fin décrit les approches que nous avons suivies reposant sur l'utilisation du problème de la cohérence SAT. L'accent a été mis sur des propriétés importantes en lien avec la modélisation, comme la correction, la complétude et la non-redondance. Nous avons tout d'abord exposé nos contributions relatives à différents types de problèmes d'extraction de motifs en fouille de données. En effet, nous avons présenté nos formulations en SAT pour des problèmes de fouille de motifs ensemblistes, de règles d'association et de motifs séquentiels. Nos contributions montrent la pertinence de l'emploi de SAT comme cadre générique adapté à la réalisation de diverses tâches en fouille de données.

Nous avons également abordé la modélisation dans des problèmes d'optimisation dérivés de la logique propositionnelle. Dans ce contexte, nous avons introduit notre cadre d'optimisation générique, appelé Top-K SAT, par l'intégration de formes de préférence sur les modèles. Dans nos travaux, nous avons en particulier montré que ce cadre généralise le problème MaxSAT partiel, et nous avons aussi proposé différentes applications en fouille de données de ce cadre. En outre, nous avons décrit une approche pour la persuasion automatique, où nous utilisons la modélisation dans un problème d'optimisation dérivé de SAT. Cette approche montre notamment l'intérêt des problèmes d'optimisation reposant sur la logique propositionnelle dans le domaine de la représentation des connaissances et raisonnements.

Nous nous sommes enfin intéressés à l'utilisation de la logique propositionnelle en présence de l'incohérence. Nous avons ainsi exposé nos contributions relatives à la notion de mesure de l'incohérence, en présentant une de nos mesures et en décrivant des propriétés concernant la compatibilité entre postulats proposés dans la littérature. Nous avons ensuite introduit notre cadre pour la définition de relations de conséquence logique paracohérentes fondé sur une notion nouvelle, appelée fonction de conséquence. En particulier, nous avons abordé les liens qu'entretient cette notion avec celle de mesure de l'incohérence. Fort des précédents outils pour raisonner en présence de l'incohérence, nous avons introduit une nouvelle tâche en fouille de données, où la logique propositionnelle est utilisée pour la représentation des données. Cette dernière offre plusieurs avantages dans la description de données complexes et hétérogènes, comme les préférences, les connaissances, les croyances, etc. En effet, notre attention s'est portée sur le

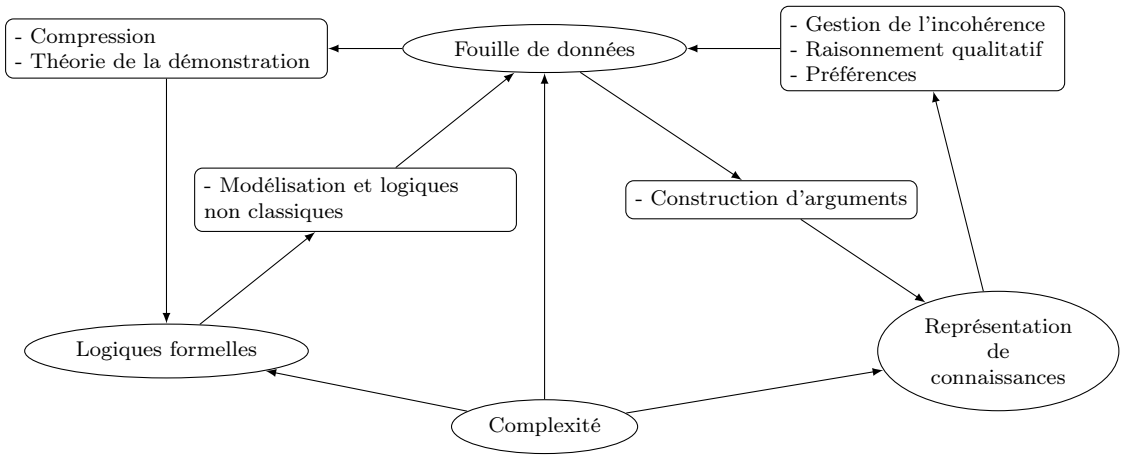


FIGURE 6.1 – Éléments principaux constituant mon projet de recherche

problème de regroupement, où les mesures de l'incohérence sont utilisées pour l'amélioration des groupes.

Dans la continuité de nos travaux de recherche, les principaux axes envisagés sont les suivants :

- La poursuite de l'utilisation d'approches fondées sur la logique propositionnelle dans la résolution de problèmes, notamment en fouille de données en insistant sur des problèmes complexes tels que ceux liés aux graphes.
- La poursuite de nos travaux sur le raisonnement en présence de l'incohérence dans les logiques formelles fondées sur la logique classique, comme en particulier les logiques modales dites normales.
- La fertilisation croisée entre la fouille de données et le domaine de la représentation des connaissances et raisonnements.
- La fertilisation croisée entre également la fouille de données et les logiques formelles.
- L'étude d'aspects liés à la complexité par rapport à l'utilisation des logiques formelles en caractérisant notamment des classes traitables.

À l'exception de la poursuite des travaux décrits dans ce manuscrit, nous exposons dans ce qui suit notre projet de recherche par rapport aux précédents axes. Ce projet est décrit de manière synthétique dans la figure 6.1.

6.1 Fouille de données et représentation des connaissances

Les contradictions pouvant exister entre connaissances extraites par des techniques de fouille de données constituent un argument fort en faveur de l'utilisation des cadres dédiés au raisonnement en présence de l'incohérence. C'est en particulier ce que nous avons fait en utilisant les mesures de l'incohérence dans la tâche de regroupement. Nous pensons qu'il est possible d'associer à la fouille de données d'autres cadres comme notamment la théorie de l'argumentation.

Une autre piste de recherche réside dans l'intégration du raisonnement qualitatif, notamment spatial et temporel [Lig13], en fouille de données par rapport à des tâches telles que l'extraction de motifs, le regroupement et la classification. Effectivement, la proximité du raisonnement qualitatif avec celui humain peut clairement être très bénéfique dans l'analyse de données complexes. Dans ce contexte, nos travaux sur les formalismes de raisonnement qualitatif peuvent être d'une grande aide.

En outre, nous avons utilisé dans [JKSS16] un cadre pour la représentation de préférences afin d'induire des préférences sur les motifs. Une troisième piste intéressante consiste donc à poursuivre nos travaux dans cette direction par l'incorporation en fouille de données d'autres cadres pour la représentation des préférences (voir [Kac11]). Cela peut entre autres permettre d'améliorer la pertinence dans l'extraction des motifs.

Par ailleurs, nous avons décrit dans le chapitre 4 notre cadre pour la persuasion automatique [Sal19c], où les arguments possèdent comme forme $\langle X, Y \rangle$, où X et Y sont des ensembles de littéraux représentant respectivement le support et la conclusion. Les arguments dans ce cas peuvent être vus comme des règles d'association particulières. L'extraction des règles d'association peut donc être utilisée dans la construction automatique d'arguments à partir de différents types d'informations, comme les réponses à un questionnaire. Il s'agit d'une piste où la fouille de données peut participer à l'enrichissement du domaine de la représentation des connaissances.

6.2 Fouille de données et logiques formelles

Nous avons proposé dans [JSSU13] une méthode utilisant la fouille de données pour la compression des formules propositionnelles en forme normale conjonctive. L'idée consiste à exploiter des structures cachées redondantes pour la réduction de la formule en entrée. Nous pensons que cette approche de compression peut être étendue à d'autre logiques formelles, notamment celles fondées sur la logique classique. En effet, l'approche de Tseitin en logique propositionnelle classique [Tse68], qui consiste à factoriser des sous-formules redondantes en les associant à des variables propositionnelles nouvelles, peut être adaptée à d'autres logiques, comme particulièrement des logiques modales.

Il nous semble également intéressant d'utiliser des techniques issues de fouille de données dans la théorie de la démonstration. Il est à noter qu'un grand nombre des systèmes de démonstration existants sont fondés sur des décompositions syntaxiques, où la notion de sous-formule joue un rôle central [TS96]. Il nous parait dans ce contexte que l'extraction de sous-formules par des mesures d'intérêt appropriées en utilisant la fouille de données pourrait être très bénéfique dans la construction de démonstrations.

De plus, nous avons pu constater dans ce mémoire que l'utilisation de la logique propositionnelle a permis d'enrichir la fouille de données. Nous sommes convaincus qu'il est également possible de tirer parti d'autres logiques formelles pour différents types de développements en fouille de données. Par exemple, nous avons montré dans [SJS12] la manière dont une logique modale peut être utilisée pour traiter un problème de cohérence relatif aux motifs ensemblistes introduit dans [Cal08].

6.3 Complexité

Je co-encadre actuellement une thèse sur l'étude de la complexité en SAT et certains autres problèmes qui en sont dérivés. Nous avons dans ce contexte proposé une nouvelle approche pour la caractérisation de classes traitables dans [BSS17]. L'idée principale consiste à transformer des instances de SAT en instances de classes traitables de problèmes dans la théorie des graphes. Dans [BS18], nous avons employé cette idée pour capturer des classes traitables dans Exactly-One SAT, qui est une variante du problème SAT. Nous envisageons d'utiliser des techniques similaires fondées sur la théorie des graphes pour l'énumération des modèles, ce qui pourrait avoir un impact important sur notre approche utilisant SAT pour l'extraction de motifs. En particulier, les graphes pourraient être des outils intéressants pour la définition de nouvelles représentations condensées pour certains types de motifs.

Troisième partie

Sélection d'articles

Liste des articles

Fouille de données

— S. Jabbour, L. Sais, Y. Salhi. Mining Top-k motifs with a SAT-based framework. Artificial Intelligence 244, 30-47 (2017).
— A. Boudane, S. Jabbour, L. Sais, Y. Salhi. A SAT-Based Approach for Mining Association Rules. 25th International Joint Conference on Artificial Intelligence, IJCAI 2016 : IJCAI/AAAI Press, 2472-2478.
— S. Jabbour, L. Sais, Y. Salhi : Boolean satisfiability for sequence mining. ACM International Conference on Information and Knowledge Management, CIKM 2013, ACM, 649-658.
— S. Jabbour, L. Sais, Y. Salhi. Decomposition Based SAT Encodings for Itemset Mining Problems. Advances in Knowledge Discovery and Data Mining - 19th Pacific-Asia Conference, PAKDD (2) 2015 : Lecture Notes in Computer Science 9078, 662-674.
— A. Boudane, S. Jabbour, L. Sais and Y. Salhi. Clustering Complex Data Represented as Propositional Formulas. Advances in Knowledge Discovery and Data Mining - 21st Pacific-Asia Conference, PAKDD (2) 2017, Lecture Notes in Computer Science 10235, 441-452.

Représentation des connaissances et raisonnements

— Y. Salhi. On an Argument-centric Persuasion Framework. International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2019. IFAAMAS, 1279-1287.
— M. Ammoura, Y. Salhi, B. Oukacha and B. Raddaoui. On an MCS-based inconsistency measure. International Journal of Approximate Reasoning 80. 443-459 (2017).
— S. Kaci, and Y. Salhi. A Constructive Argumentation Framework. AAAI Conference on Artificial Intelligence, AAAI 2014, AAAI Press, 1070-1076.

Mining Top-k motifs with a SAT-based framework

S. Jabbour, L. Sais, Y. Salhi. Mining Top-k motifs with a SAT-based framework. Artificial Intelligence 244, 30-47 (2017).

Mining Top-k Motifs with a SAT-based Framework

Said Jabbour and Lakhdar Sais and Yakoub Salhi

CRIL - CNRS, Université d'Artois, France
F-62307 Lens Cedex, France

Abstract

In this paper, we introduce a new problem, called Top-k SAT, that consists in enumerating the Top-k models of a propositional formula. A Top-k model is defined as a model with less than k models preferred to it with respect to a preference relation. We show that Top-k SAT generalizes two well-known problems: the partial Max-SAT problem and the problem of computing minimal models. Moreover, we propose a general algorithm for Top-k SAT. Then, we give an application of our declarative framework in data mining, namely, the problem of mining Top-k motifs in the transaction databases and in the sequences. In the case of mining sequence data, we introduce a new mining task by considering the sequences of itemsets. Thanks to the flexibility and to the declarative aspects of our SAT-based approach, an encoding of this task is obtained by a very slight modification of mining motifs in the sequences of items.

Keywords: Boolean satisfiability; Data Mining; Modeling; Top-k motifs

1. Introduction

The problem of mining frequent itemsets is well-known and essential in data mining, knowledge discovery and data analysis. It has applications in various fields and becomes fundamental for data analysis as datasets and datastores are becoming very large. Since the first article of Agrawal [1] on association rules and itemset mining, the huge number of works, challenges, datasets and projects show the actual interest in this problem (see [2] for a recent survey of works addressing this problem).

Email address: {jabbour, sais, salhi}@cril.fr (Said Jabbour and Lakhdar Sais and Yakoub Salhi)

We are also interested in frequent sequence data mining which is the problem of discovering frequent patterns shared across time among an input data-sequence. Sequence mining is a central task in computational biology, temporal sequence analysis and text mining. In this work, we consider the pattern discovery problem for a specific class of patterns with wildcards in a sequence. The data-sequence can be seen as a sequence of items, while the pattern can be seen as a subsequence that might contains wildcards or jokers in the sense that they match any item [3, 4, 5]. At the first sight, allowing wildcards to occur in a pattern can be seen as an even more restrictive type of patterns in general. However as argued in [3] "*studying patterns with wildcards has the merit of capturing one important aspect of biological features that often concerns isolated positions inside a motif that are not part of the biological feature being captured*".

Important progress has been achieved for data mining and knowledge discovery in terms of implementations, platforms, libraries, etc. As pointed out in [2], several works deal with designing highly scalable data mining algorithms for large scale datasets. An important problem of data mining problems, in general, concerns the huge size of the output, from which it is difficult for the user to retrieve relevant information. Consequently, for practical data mining, it is important to reduce the size of the output by exploiting the structure of the patterns. Computing for example, closed, maximal, condensed, discriminative patterns are some of the well-known and useful techniques. Most of the works on itemset and sequential mining require the specification of a minimum support threshold λ. This constraint allows the user to control at least to some extent the size of the output by mining only patterns covering at least λ transactions (locations). However, in practice, it is difficult for users to provide an appropriate threshold. As pointed out in [6], a too small threshold may lead to the generation of a huge number of patterns, whereas a too high value of the threshold may result in no answer. In [6], based on a complete ranking between itemsets, the authors propose to mine the n most interesting itemsets of arbitrary length. In [7], the proposed task consists in mining Top-k frequent closed itemsets of length greater than a given lower bound min, where k is the desired number of frequent closed itemsets to be mined, and min is the minimal length of each itemset. The authors show that setting the minimal length of the itemsets to be mined is much easier than setting the usual frequency threshold. Since the introduction of Top-k mining, several research works investigated its use in graph mining (e.g. [8, 9]) and other datamining tasks (e.g. [10, 11]). This

new framework can be seen as a nice way to mine the k preferred patterns according to some specific constraints or measures. Starting from this observation, our goal in this paper is to define a general logic based framework for enumerating the Top-k preferred patterns according to a predefined preference relation.

The notion of preference has a central role in several disciplines such as economics, operations research and decision theory in general. Preferences are relevant for the design of intelligent systems that support decisions. Modeling and reasoning with preferences play an increasing role in Artificial Intelligence (AI) and its related fields such as nonmonotonic reasoning, planning, diagnosis, configuration, constraint programming (CP) and other areas in knowledge representation and reasoning (KR). For example, in nonmonotonic reasoning the introduction of preferential semantics by Shoham [12] gives an unifying framework where nonmonotonic logic is reduced to a standard logic with a preference relation (order) on the models of that standard logic. Several models for representing and reasoning about preferences have been proposed. For example, soft constraints [13] are one of the most general way to deal with quantitative preferences, while CP-net (Conditional Preferences networks) [14] is most convenient for qualitative preferences. There is a huge literature on preferences (see [15, 16, 17] for a survey at least from the AI perspective). In data mining, preferences have also been investigated by several authors (e.g. [18, 19, 20]). For example, in [18], the authors introduced a new paradigm of pattern discovery based on Soft Constraints, where constraints are no longer rigid boolean functions. More recently, Ugarte et al. [20] introduced a generic and efficient method based on constraint programming for mining (soft-)skypatterns that enable to express user preference according to dominance relations. From the observation that dominance relations can be found in many pattern mining settings, in [21], the authors propose an algebra that combines constraints and dominance relations that can be used to describe a broad range of pattern mining settings.

In this paper we focus on qualitative preferences defined by a preference relation on the models of a propositional formula. Preferences in propositional satisfiability (SAT) has not received a lot of attention. In [22], a new approach for solving satisfiability problems in the presence of qualitative preferences on literals (defined as partial ordered set) is proposed. The authors particularly show how DPLL procedure can be easily adapted for computing optimal models induced by the partial order. The issue of computing optimal

models using DPLL has also been investigated in SAT [23].

Recently, a constraint programming (CP) based data mining (DM) framework was proposed by Luc De Raedt et al. in [24] for itemset mining (CP4IM). This new framework offers a declarative and flexible representation model. New constraints often require new implementations in specialized approaches, while they can be easily integrated in such a CP framework. It allows data mining problems to benefit from several generic and efficient CP solving techniques. The authors show how some typical constraints (e.g. frequency, maximality, monotonicity) used in itemset mining can be formulated for use in CP [25]. This study leads to the first CP approach for itemset mining displaying nice declarative opportunities. Encouraged by these promising results, several contributions addressed other data mining problems using the two well-known CP and SAT AI formalisms. For example, in [26], the authors proposed a SAT-based encoding for the problem of discovering frequent, closed and maximal patterns in a sequence of items and a sequence of itemsets. In [27], the authors solve the frequent itemset mining problem by compiling the set of all itemset into a binary decision diagram (BDD) (augmented with counts). Frequent itemset are then extracted by querying the BDD. By considering the relationship between local constraint-based mining and constraint satisfaction problems, Khiari et al. [28] proposed a model for mining patterns combining several local constraints, i.e., patterns defined by n-ary constraints. Also, several constraint-based language for modeling and solving data mining problems have been designed. Let us mention the constraint-based language defined in [29], which enables the user to define queries in a declarative way addressing pattern sets and global patterns. All primitive constraints of the language are modeled and solved using the SAT framework. More recently, Guns at al. [30], introduced a general-purpose declarative mining framework, called MiningZinc. Compared to CP4IM framework [31], MiningZinc supports a wide variety of different solvers (including DM algorithms and general purpose solvers) and uses a significantly more high-level modeling language.

This new research trend offers nice opportunities for cross-fertilization between AI and data mining. The work presented in this paper fit into this framework. Our goal is to provide a step forward towards the integration of AI and Data mining. Our approach is based on SAT as the underlined data mining constraints involve Boolean variables.

The contributions of this paper are the following:

1. Firstly, we propose a generic framework for dealing with qualitative preferences in propositional satisfiability. Our qualitative preferences are defined using a reflexive and transitive relation (preorder) over the models of a propositional formula. Such preference relation on models is first used to introduce a new problem, called Top-k SAT, defined as the problem of enumerating the Top-k models of a propositional formula. Here a Top-k model is defined as a model with no more than k-1 models preferred to it with respect to the considered preference relation. Then, we show that Top-k SAT generalizes the two well-known problems, the partial Max-SAT problem and the problem of generating minimal models. We also define a particular type of preference relations that allows us to introduce a general algorithm for solving the Top-k SAT problem.

2. Secondly, we introduce our first application of our declarative framework to data mining. More precisely, we consider the problem of mining Top-k frequent closed itemsets of minimum length min [32]. In this problem, the minimum support threshold usually used in frequent itemset mining is not known, while the minimum length can be set to 0 if one is interested in itemsets of arbitrary length. In itemset mining, the notion of Top-k itemset was introduced in [7] as an alternative to finding the appropriate value for the minimum support threshold. It is also an elegant way to control the size of the output. Consequently, itemset mining is clearly a nice application of our new defined Top-k SAT problem. In this context, we provide a SAT encoding and we show that computing the Top-k closed itemsets of length at least min corresponds to computing the Top-k models of the obtained propositional formula.

3. Thirdly, we provide an application of our declarative framework to sequence mining. Indeed, we propose an encoding of the problem of enumerating the Top-k patterns with wildcards in a sequence of items in our framework. The notion of Top-k pattern in this context is defined in the same way as in itemset mining. The SAT encoding that we use comes from our encoding proposed in [26].

4. In the fourth contribution, we consider a variant of the problem of discovering patterns with wildcards in a sequence, proposed in our work in [26], by considering a sequence of itemsets instead of a sequence of

items. In this extension the emptyset will simply play the same role as the wildcard symbol. Indeed, one can use the emptyset to match any itemset. This new problem admits some similarities and differences with the classical sequential pattern mining problem introduced in [33]. Indeed, given an alphabet or a set of items Σ, in both problems we consider a sequence s as an ordered list of itemsets $s_0, \ldots, s_n$ where $s_i \subseteq \Sigma$ for $i = 0, \ldots, n$. However, the first difference resides in the definition of a subsequence. Indeed, in the sequential patterns, we say that s' is a subsequence of s if there exists a one-to-one order-preserving function f that maps (inclusion relation) itemsets in s' with itemsets in s. In our new setting, the notion of subsequence is defined w.r.t. to a given location and by using empty itemsets as wildcards. The other difference is that in the sequential pattern mining we consider a database of sequences of itemsets, while in our setting, we consider only a single sequence of itemsets. In this case, an encoding of the problem of enumerating the Top-k patterns in our framework is derived from the one used for the sequences of items with a very slight modification demonstrating its flexibility.

In this paper, we extend our work published in [34]. The extension includes the application of our declarative framework to sequence mining (contributions 3 and 4) using our SAT encodings introduced in [26]. We also provide an extensive experimental results showing the feasibility of our proposed approach.

2. Preliminary definitions and notations

In this section, we describe the Boolean satisfiability problem (SAT), the itemset mining problem and introduce some necessary notations.

Let $\mathcal{P}$ be a propositional language of formulas $\mathcal{F}_{\mathcal{P}}$ built in the standard way, using usual logical connectives ($\vee$, $\wedge$, $\neg$, $\rightarrow$, $\leftrightarrow$) and a set of propositional variables. A *CNF formula* (Conjunctive Normal Form) Φ is a conjunction of clauses, where a *clause* is a disjunction of literals. A *literal* is a positive (p) or negated ($\neg p$) propositional variable. The two literals p and $\neg p$ are called *complementary*. A CNF formula can also be seen as a set of clauses, and a clause as a set of literals. Let us recall that any propositional formula can be translated to CNF using linear Tseitin's encoding [35]. We denote by $Var(\Phi)$ the set of propositional variables occuring in Φ.

An *interpretation* $\mathcal{M}$ of a propositional formula Φ is a function which associates a value $\mathcal{M}(p) \in \{0, 1\}$ (0 for *false* and 1 for *true*) to each propositional variable p in $Var(\Phi) \subseteq V$. A *model* of a formula Φ is an interpretation $\mathcal{M}$ that satisfies the formula. The *SAT problem* consists in deciding whether a given CNF formula admits a model.

We denote by $\bar{l}$ the complementary literal of l. More precisely, if $l = p$ then $\bar{l}$ is $\neg p$ and if $l = \neg p$ then $\bar{l}$ is p. For a set of literals L, $\bar{L}$ is defined as $\{\bar{l} \mid l \in L\}$. Moreover, we denote by $\overline{\mathcal{M}}$ ($\mathcal{M}$ is an interpretation over $Var(\Phi)$) the clause $\bigvee_{p \in Var(\Phi)} s(p)$, where $s(p) = p$ if $\mathcal{M}(p) = 0$, $\neg p$ otherwise. Let Φ be a CNF formula and $\mathcal{M}$ an interpretation over $Var(\Phi)$. We denote by $\mathcal{M}(\Phi)$ the set of clauses satisfied by $\mathcal{M}$. Let us now consider a set X of propositional variables such that $X \subseteq Var(\Phi)$. We denote by $\mathcal{M} \cap X$ the set of variables $\{p \in X | \mathcal{M}(p) = 1\}$. Moreover, we denote by $\mathcal{M}_{|X}$ the restriction of the model $\mathcal{M}$ to X.

Let us now introduce the itemset mining problem.
Transaction database. Let $\mathcal{I}$ be a set of *items*. A *transaction* is a couple (tid, I) where tid is the *transaction identifier* and I is an *itemset*, i.e., $I \subseteq \mathcal{I}$. A *transaction database* is a finite set of transactions over $\mathcal{I}$ where each transaction identifier refers to only one itemset.

Cover, support and frequency of an itemset. We say that a transaction (tid, I) *supports* an itemset J if $J \subseteq I$. The *cover* of an itemset I in a transaction database $\mathcal{D}$ is the set of transaction identifiers in $\mathcal{D}$ supporting I: $\mathcal{C}(I, \mathcal{D}) = \{tid \mid (tid, J) \in \mathcal{D}, I \subseteq J\}$. The *support* of an itemset I in $\mathcal{D}$ is defined by: $\mathcal{S}(I, \mathcal{D}) = |\mathcal{C}(I, \mathcal{D})|$. Moreover, the frequency of I in $\mathcal{D}$ is defined by: $\mathcal{F}(I, \mathcal{D}) = \frac{\mathcal{S}(I,\mathcal{D})}{|\mathcal{D}|}$.

For instance, consider the following transaction database:

tid	itemset
1	a, b, c, d
2	a, b, e, f
3	a, b, c, m
4	a, c, d, f, j
5	j, l
6	d
7	d, j

Transaction database $\mathcal{D}$

In this database, we have $\mathcal{S}(\{a, b, c\}, \mathcal{D}) = |\{1, 3\}| = 2$ and $\mathcal{F}(\{a, b\}, \mathcal{D}) = \frac{3}{7}$.

Let $\mathcal{D}$ be a transaction database over $\mathcal{I}$ and λ a minimum support threshold. The *frequent itemset mining problem* consists in computing the following set:

$$\mathcal{FIM}(\mathcal{D}, \lambda) = \{I \subseteq \mathcal{I} \mid \mathcal{S}(I, \mathcal{D}) \geq \lambda\}$$

Let us now define one of the well known condensed representation of frequent itemsets.

Definition 1 (Closed Itemset)**.** *Let $\mathcal{D}$ be a transaction database (over $\mathcal{I}$) and I an itemset ($I \subseteq \mathcal{I}$) such that $\mathcal{S}(I, \mathcal{D}) \geq 1$. I is closed if, for all itemset J with $I \subset J$, $\mathcal{S}(J, \mathcal{D}) < \mathcal{S}(I, \mathcal{D})$.*

One can easily see that all frequent itemsets can be obtained from the closed frequent itemsets by computing their subsets. Since the number of closed frequent itemsets is smaller than or equal to the number of frequent itemsets, enumerating all closed itemsets allows us to reduce the size of output without losing information.

3. Preferences and Top-k models

Let Φ be a propositional formula and Λ_Φ the set of all its models. A preference relation $\succeq$ over Λ_Φ is a reflexive and transitive binary relation (a preorder). The statement $\mathcal{M} \succeq \mathcal{M}'$ means that $\mathcal{M}$ is at least as preferred as $\mathcal{M}'$. We denote by $P(\Phi, \mathcal{M}, \succeq)$ the subset of Λ_Φ defined as follows:

$$P(\Phi, \mathcal{M}, \succeq) = \{\mathcal{M}' \in \Lambda_\Phi \mid \mathcal{M}' \succ \mathcal{M}\}$$

where $\mathcal{M}' \succ \mathcal{M}$ means that $\mathcal{M}' \succeq \mathcal{M}$ holds but $\mathcal{M} \succeq \mathcal{M}'$ does not. It corresponds to all models that are strictly preferred to $\mathcal{M}$.

We now introduce an equivalence relation $\approx_X$ over $P(\Phi, \mathcal{M}, \succeq)$, where X is a set of propositional variables. It is defined as follows:

$$\mathcal{M}' \approx_X \mathcal{M}'' \text{ iff } \mathcal{M}' \cap X = \mathcal{M}'' \cap X$$

Thus, the set $P(\Phi, \mathcal{M}, \succeq)$ can be partitioned into a set of equivalence classes by $\approx_X$, denoted by $[P(\Phi, \mathcal{M}, \succeq)]^X$. Each equivalence class is a set of equivalent models with respect to X that are strictly preferred to $\mathcal{M}$. Then, $|[P(\Phi, \mathcal{M}, \succeq)]^X|$ represents the number of equivalent classes in the set of strictly preferred models to $\mathcal{M}$ with respect to X. In our context, this equivalence relation is used to take into consideration only a subset of propositional variables. For instance, we introduce new variables in Tseitin's translation [35] of propositional formula to CNF, and such variables are not important in the case of some preference relations.

Definition 2 (Top-k Model)**.** *Let Φ be a propositional formula, $\mathcal{M}$ a model of Φ, $\succeq$ a preference relation over the models of Φ and X a set of propositional variables. $\mathcal{M}$ is a Top-k model w.r.t. $\succeq$ and X iff $|[P(\Phi, \mathcal{M}, \succeq)]^X| \leq k - 1$.*

Let us note that the number of the Top-k models of a formula is not necessarily equal to k. Indeed, it can be strictly greater or smaller than k. For instance, if a formula is unsatisfiable, then it does not have a Top-k model for any $k \geq 1$. Nevertheless, if the considered preference relation is a complete order, then the number of Top-k models is always smaller than or equal to k.

It is easy to see that we have the following *monotonicity property*: if $\mathcal{M}$ is a Top-k model and $\mathcal{M}' \succeq \mathcal{M}$, then $\mathcal{M}'$ is also a Top-k model.

Top-k SAT problem. Let Φ be propositional formula, $\succeq$ a preference relation over the models of Φ, X a set of propositional variables and k a strictly positive integer. We call the tuple $(\Phi, \succeq, X, k)$ a *Top-k instance*. The Top-k SAT problem consists in computing a set $\mathcal{L}$ of Top-k models of Φ with respect to $\succeq$ and X satisfying the two following properties:

1. for all Top-k model $\mathcal{M}$, there exists $\mathcal{M}' \in \mathcal{L}$ such that $\mathcal{M} \approx_X \mathcal{M}'$; and
2. for all $\mathcal{M}$ and $\mathcal{M}'$ in $\mathcal{L}$, if $\mathcal{M} \neq \mathcal{M}'$ then $\mathcal{M} \not\approx_X \mathcal{M}'$.

The two previous properties come from the fact that we are only interested in the truth values of the variables in X. Indeed, the first property means that, for all Top-k model, there is a model in $\mathcal{L}$ equivalent to it with respect to $\approx_X$. Moreover, the second property means that $\mathcal{L}$ does not contain two equivalent Top-k models.

In the following definition, we introduce a particular type of preference relations, called δ-preference relations, that allows us to introduce a general algorithm for computing Top-k models.

Definition 3. *Let Φ be a CNF formula and $\succeq$ a preference relation on the models of Φ. Then $\succeq$ is a δ-preference relation, if there exists a function $\delta^{\phi}_{\succeq}$ from Λ_Φ to the set of CNF formulae, called* bound function*, such that (i) it is polynomially computable in the size of Φ and (ii) for all $\mathcal{M} \in \Lambda_\Phi$, $\mathcal{M}'$ is a model of $\Phi \wedge \delta^{\phi}_{\succeq}(\mathcal{M})$ iff $\mathcal{M}'$ is a model of Φ and $\mathcal{M} \not\succ \mathcal{M}'$, for every Boolean interpretation $\mathcal{M}'$.*

Let us note that, given a model $\mathcal{M}$ of a CNF formula Φ, $\delta^{\Phi}_{\succeq}(\mathcal{M})$ is a CNF formula so that when added (with conjunction) to Φ together with $\overline{\mathcal{M}}$, the models of the resulting formula are different from $\mathcal{M}$ and they correspond to all the models of Φ which are not less preferred than $\mathcal{M}$. Intuitively, this can be seen as a way to introduce a lower bound during the enumeration process. In other words, at each step of the enumeration process, a δ-preference relation allows us to generate a formula from the current found model to force the algorithm to enumerate the models that are not less preferred.

Clearly, if we ignore the condition (i), each preference relation is a δ-preference relation. Indeed, for every preference relation $\succeq$ on Λ_Φ, we only have to define the bound function $\delta^{\phi}_{\succeq}$ as follows:

$$\delta^{\phi}_{\succeq}(\mathcal{M}) = \bigwedge_{\mathcal{M}' \in \Lambda_\Phi, \mathcal{M} \not\succ \mathcal{M}'} \overline{\mathcal{M}'}$$

This definition means that we exclude all the models of Φ that are less preferred than $\mathcal{M}$. One can easily see that in this case $\delta^{\phi}_{\succeq}(\mathcal{M})$ is not polynomially computable in the worst case.

From now, we only consider the δ-preference relations. We also consider that their bound functions are provided.

3.1. Top-k SAT and Partial MAX-SAT

In this section, we show that the Top-k SAT problem generalizes the Partial MAX-SAT problem (e.g. [36]). In Partial MAX-SAT each clause is either relaxable (soft) or non-relaxable (hard). The objective is to find an interpretation that satisfies all the hard clauses together with the maximum number of soft clauses. The MAX-SAT problem is a particular case of Partial MAX-SAT where all the clauses are relaxable.

Let $\Phi = \Phi_h \wedge \Phi_s$ be a partial MAX-SAT instance such that Φ_h is the hard part and Φ_s the soft part. The relation denoted by $\succeq_{\Phi_s}$ corresponds to a preference relation defined as follows: for all $\mathcal{M}$ and $\mathcal{M}'$ models of Φ_h defined over $Var(\Phi_h \wedge \Phi_s)$, $\mathcal{M} \succeq_{\Phi_s} \mathcal{M}'$ if and only if the number of soft clauses satisfied by $\mathcal{M}$ is greater than or equal to the number of soft clauses satisfied by $\mathcal{M}'$, i.e., $|\mathcal{M}(\Phi_s)| \geq |\mathcal{M}'(\Phi_s)|$. Clearly, the set of models of Φ_h over $Var(\Phi_h \wedge \Phi_s)$ is isomorphic to that of $\Phi' = \Phi_h \wedge \bigwedge_{C \in \Phi_s} p_C \leftrightarrow C$, where p_C for $C \in \Phi_s$ are fresh propositional variables. Indeed, for every model $\mathcal{M}$ of Φ', $\mathcal{M}_{|Var(\Phi)}$ is a model of Φ_h. Further, for every model $\mathcal{M}$ of Φ_h, the following Boolean interpretation $\mathcal{M}'$ is a model of Φ':

$$\mathcal{M}'(p) = \begin{cases} \mathcal{M}(p) & if\ p \in Var(\Phi) \\ 1 & if\ p \equiv p_C\ and\ \mathcal{M}(C) = 1 \\ 0 & otherwise \end{cases}$$

We define the preference relation $\succeq$ over the models of Φ' as follows: $\mathcal{M} \succeq \mathcal{M}'$ if and only if $\mathcal{M}_{|Var(\Phi)} \succeq_{\Phi_s} \mathcal{M}'_{|Var(\Phi)}$. It is a δ-preference relation since its bound function $\delta_{\succeq}^{\Phi'}$ can be defined as follows:

$$\delta_{\succeq}^{\Phi'}(\mathcal{M}) = \sum_{C \in \Phi_s} p_C \geq |\mathcal{M}(\Phi_s)|$$

If S is the set of all the Top-1 models of Φ' with respect to $\succeq$ and $Var(\Phi)$, then the $S' = \{\mathcal{M}_{|Var(\Phi)} \mid \mathcal{M} \in S\}$ correspond to the set of all solutions of Φ in Partial Max-SAT. Naturally, the models in S' are the most preferred models with respect to $\succeq_{\Phi_s}$, and that means they satisfy Φ_h and satisfy the maximum number of clauses in Φ_s. Thus, in a sense, the Top-k SAT problem can be seen as a generalization of Partial MAX-SAT.

The formula $\delta_{\succeq}^{\Phi'}(\mathcal{M})$ involves the well-known cardinality constraint (0/1 linear inequality). Several polynomial encodings of this kind of constraints into

CNF formulas have been proposed in the literature. The first linear encoding of general linear inequalities to CNF has been proposed by Warners [37]. Recently, efficient encodings of the cardinality constraint to CNF have been proposed, most of them try to improve the efficiency of constraint propagation (e.g. [38, 39, 40, 41]).

3.2. Top-k SAT and X-minimal Model Generation Problem

Let Φ be a formula, $\mathcal{M}$ and $\mathcal{M}'$ two models of Φ and X a set of propositional variables. Then, $\mathcal{M}$ is said to be smaller than $\mathcal{M}'$ with respect to X, written $\mathcal{M} \leq_X \mathcal{M}'$, if $\mathcal{M} \cap X \subseteq \mathcal{M}' \cap X$. We now define a preference relation $\succeq_X$ as follows: $\mathcal{M} \succeq_X \mathcal{M}'$ if and only if $\mathcal{M} \leq_X \mathcal{M}'$ i.e. $\mathcal{M}$ is at least as preferred as $\mathcal{M}'$.

We now show that $\succeq_X$ is a δ-preference relation. We can define $\delta^{\Phi}_{\succeq_X}$ as follows:

$$\delta^{\Phi}_{\succeq_X}(\mathcal{M}) = (\bigwedge_{p \in \mathcal{M} \cap X} p) \rightarrow \bigwedge_{p' \in X \setminus \mathcal{M} \cap X} \overline{p'}$$

Indeed, $\mathcal{M}'$ is a model of a formula $\Phi \wedge \overline{\mathcal{M}} \wedge \delta^{\Phi}_{\succeq_X}(\mathcal{M})$ if and only if $\mathcal{M}'$ is a model of Φ, $\mathcal{M}' \neq \mathcal{M}$, and either $(\mathcal{M} \cap X) \setminus (\mathcal{M}' \cap X) \neq \emptyset$ or $\mathcal{M}' \cap X \subseteq \mathcal{M} \cap X$. The two previous statements mean that $\mathcal{M} \not\succ_X \mathcal{M}'$. In fact, if $\mathcal{M}'$ satisfies $(\bigwedge_{p \in \mathcal{M} \cap X} p)$ then $\mathcal{M}'$ must satisfy $\bigwedge_{p' \in X \setminus \mathcal{M} \cap X} \overline{p'}$. As a consequence, we deduce that $\mathcal{M}' \cap X \subseteq \mathcal{M} \cap X$. Otherwise, $\mathcal{M}'$ falsify $(\bigwedge_{p \in \mathcal{M} \cap X} p)$ and that means that $(\mathcal{M} \cap X) \setminus (\mathcal{M}' \cap X) \neq \emptyset$. This latter statement expresses that either $\mathcal{M}' \cap X \subset \mathcal{M} \cap X$ or $\mathcal{M}$ and $\mathcal{M}'$ are incomparable with respect to $\succeq_X$.

Let Φ be a propositional formula, X a set of propositional variables and $\mathcal{M}$ a model of Φ. Then $\mathcal{M}$ is said to be an *X-minimal model* of Φ if there is no model strictly smaller than $\mathcal{M}$ with respect to $\succeq_X$. In [42], it was shown that finding an X-minimal model is $P^{NP[O(log(n))]}$-hard, where n is the number of propositional variables.

The set of all X-minimal models corresponds to the set of all top-1 models with respect to $\succeq_X$ and $Var(\Phi)$. Indeed, if $\mathcal{M}$ is a top-1 model, then there is no model $\mathcal{M}'$ such that $\mathcal{M}' \succ_X \mathcal{M}$, and that means that $\mathcal{M}$ is an X-minimal model. In this context, let us note that computing the set of Top-k models for $k \geq 1$ can be seen as a generalization of X-minimal model generation problem.

Algorithm 1: Top-k

Input: a Top-k instance $(\Phi, \succeq, X, k)$
Output: A solution $\mathcal{L}$ of $(\Phi, \succeq, X, k)$

```
1  Φ' ← Φ;
2  L ← ∅;                                              /* Set of all Top-k models */
3  while (M = solve(Φ')) do                            /* M is a model of Φ' */
4      if (∃M' ∈ L s.t. M ≈_X M' & M ≻ M') then
5          replace(M, M', L);
6      else if (∀M' ∈ L s.t. M ≉_X M' & |preferred(M, L)| < k) then
7          S ← min_top(k, L);
8          L ← L ∪ M;
9          removeNotTop-k(k, L);
10         S ← min_top(k, L) \ S;
11         Φ' ← Φ' ∧ ⋀_{M'∈S} δ^Φ_≥(M');
12     else
13         Φ' ← Φ' ∧ δ^Φ_≥(M)
14     Φ' ← Φ' ∧ ¬M;
15 return L;
```

3.3. An algorithm for Top-k SAT

In this section, we describe our algorithm for computing Top-k models in the case of the δ-preference relations (Algorithm 1). The basic idea is simply to use the formula $\delta^{\Phi}_{\succeq}(\mathcal{M})$ associated to a model $\mathcal{M}$ to obtain models that are not less preferred than $\mathcal{M}$. This algorithm takes as input a CNF formula Φ, a preference relation $\succeq$, a strictly positive integer k, and a set X of propositional variables allowing to define the equivalence relation $\approx_X$. It has as output a set $\mathcal{L}$ of Top-k models of Φ satisfying the two properties given in the definition of the Top-k SAT problem.

3.3.1. Algorithm description

In the while-loop, we use lower bounds for finding optimal models. These lower bounds are obtained by using the fact that the preorder relation considered is a δ-preference relation. In each step, the lower bound is integrated by using the formula:

$$\bigwedge_{\mathcal{M}' \in S} \delta^{\Phi}_{\succeq}(\mathcal{M}')$$

- **Lines 3.** Let us first mention that the function $solve(\phi')$ refer to any SAT solver or DPLL procedure that enumerates all the models of a given CNF formula.

- **Lines 4 – 5.** The procedure `replace(`$\mathcal{M}$`,`$\mathcal{M}'$`,`$\mathcal{L}$`)` replaces $\mathcal{M}'$ with $\mathcal{M}$ in $\mathcal{L}$. We apply this replacement because there exists a model $\mathcal{M}'$

in $\mathcal{L}$ which is equivalent to $\mathcal{M}'$ and $\mathcal{M}$ allows to have a better bound.

- **Lines 6 – 11.** In the case where $\mathcal{M}$ is not equivalent to any model in $\mathcal{L}$ and the number of models in $\mathcal{L}$ preferred to it is strictly less than k ($|\texttt{preferred}(\mathcal{M}, \mathcal{L})| < k$), we add $\mathcal{M}$ to $\mathcal{L}$ (line 8). Note that $\mathcal{S}$ contains first the models of $\mathcal{L}$ before adding $\mathcal{M}$ that have exactly $k-1$ models preferred to them in this set. After adding $\mathcal{M}$ to $\mathcal{L}$, we remove from $\mathcal{L}$ the models that are not Top-k, i.e., they have more than $k-1$ models in $\mathcal{L}$ that are strictly preferred to them ($\texttt{removeNotTop-k}(k, \mathcal{L})$). Next, we modify the content of S. Note that the elements of S before adding $\mathcal{M}$ are used as bounds in the previous step. Hence, in order to avoid adding the same bound several times, the new content of S corresponds to the models in $\mathcal{L}$ that have exactly $k-1$ models preferred to them in $\mathcal{L}$ ($\texttt{min\_top}(k, \mathcal{L})$) deprived of the elements of the previous content of S. In line 11, we integrate lower bounds in Φ' by using the elements of S. Indeed, for all model $\mathcal{M}$ of a formula $\Phi' \wedge \bigwedge_{\mathcal{M}' \in S} \delta^{\Phi}_{\geq}(\mathcal{M}')$, $\mathcal{M}' \not\succ \mathcal{M}$ holds, for any $\mathcal{M}' \in S$.

- **Lines 12 – 13.** In the case where $\mathcal{M}$ is not a Top-k model, we integrate its associated lower bound.

- **Line 14.** This instruction enables us to avoid finding the same model in two different steps of the while-loop.

Proposition 1. *Algorithm 1 (Top-k) is correct.*

Proof. The proof of the partial correctness is based on the definition of the δ-preference relation. Indeed, the function $\delta^{\Phi}_{\geq}$ allows us to exploit bounds to systematically improve the preference level of the models. Indeed, using the condition (ii) of the δ-preference relation, adding $\bigwedge_{\mathcal{M}' \in S} \delta^{\Phi}_{\geq}(\mathcal{M}')$ at Line 11 and $\delta^{\Phi}_{\geq}(\mathcal{M})$ at Line 13 ensure that we consider in the next iterations only the models that are not less preferred than $k-1$ found models. Further, as the number of models is bounded, adding the negation of the found model at each iteration leads to an unsatisfiable formula. Consequently the algorithm terminates.

□

3.4. Complete preference relations

Our aim in this section is to show how Top-k SAT problem can benefit from algorithms of standard optimization problems in the case of complete

preference relations. We provide in particular a simple algorithm that allows to find a Top-1 solution from a single Top-1 model, which may be found using an algorithm of an optimization problem.

Let us recall that a preference relation $\succeq$ is complete if, for all models $\mathcal{M}$ and $\mathcal{M}'$, we have $\mathcal{M} \succeq \mathcal{M}'$ or $\mathcal{M}' \succeq \mathcal{M}$. It is worth noting that the size of a solution of a Top-k SAT instance may also be greater than k in the case of complete preference relations, since solutions might be equivalent w.r.t. the equivalence relation induced by the preference relation. However, we get the following interesting property:

Proposition 2. *Let Φ be a propositional formula, $\succeq$ a complete preference relation, X a set of propositional variables and $\mathcal{M}$ a Top-1 model of Φ w.r.t. $\succeq$ and X. Then, for all $\mathcal{M}' \in \Lambda_\Phi$, $\mathcal{M}'$ is a Top-1 model of Φ w.r.t. $\succeq$ and X iff $\mathcal{M}' \approx \mathcal{M}$, where $\approx$ is the equivalence relation induced by $\succeq$.*

Proof. The "if part" is a consequence of the fact that $\mathcal{M}' \succeq \mathcal{M}$ and $\mathcal{M}$ is a Top-1 model. We now consider the "only if part". Let $\mathcal{M}'$ be a Top-1 model. Knowing that $\mathcal{M}$ is a Top-1 model and $\succeq$ is a complete preference relation, we get $\mathcal{M} \succeq \mathcal{M}'$. We also get $\mathcal{M}' \succeq \mathcal{M}$ since $\mathcal{M}$ is a Top-1 model. As a consequence, $\mathcal{M} \approx \mathcal{M}'$ holds. □

Note that the property highlighted in Proposition 2 is not true for any preference relation. Indeed, there exist preference relations in Top-k instances for which two distinct Top-1 models are incomparable.

In a sense, Proposition 2 describes a simple way to compute a solution of a Top-1 instance from a single Top-1 model in the case of complete preference relations. Indeed, such computation process is described in Algorithm 2, where we use an approach widely used for solving model enumeration problems. Usually, the algorithms designed to solve this problem are based on the use of additional clauses, called blocking clauses, to avoid producing repeated models [43]. Similarly, in Algorithm 2, we add to the Top-1 instance a blocking clause $\overline{\mathcal{M}_{|X}}$ (line 1) to avoid models equivalent to $\mathcal{M}$ w.r.t. the equivalence relation $\approx_X$, and a formula $\delta^\Phi_\succeq(\mathcal{M})$ (line 1) to only consider the models that are at least as preferred as the input Top-1 model $\mathcal{M}$, i.e. the models that are equivalent to $\mathcal{M}$ w.r.t. the equivalence relation induced by the preference relation. In this way, at each iteration of the while loop, the next found model $\mathcal{M}'$ (line 3) is a Top-1 model. At each iteration, we also

Algorithm 2: Top-1^C

Input: Top-1 instance $(\Phi, \succeq, X, 1)$ s.t. $\succeq$ is a complete preference relation and $\mathcal{M}$ a Top-1 model
Output: Solution $\mathcal{L}$ of $(\Phi, \succeq, X, 1)$

1 $\Phi' \leftarrow \Phi \wedge \overline{\mathcal{M}_{|X}} \wedge \delta^{\Phi}_{\succeq}(\mathcal{M})$;
2 $\mathcal{L} \leftarrow \{\mathcal{M}\}$;
3 **while** *($\mathcal{M}' = solve(\Phi')$)* **do** `/* M' is a model of Φ' */`
4 $\quad \mathcal{L} \leftarrow \mathcal{L} \cup \{\mathcal{M}'\}$;
5 $\quad \Phi' \leftarrow \Phi' \wedge \overline{\mathcal{M}'_{|X}}$;
6 **return** $\mathcal{L}$;

make use of a blocking clause $\overline{\mathcal{M}'_{|X}}$ (line 5) to avoid equivalent models w.r.t. the equivalence relation $\approx_X$.

As a consequence, for some complete preference relations, methods for computing a solution for a Top-1 instance can be obtained by combining Algorithm 2 with algorithms for standard optimization problems, such as Partial Max-SAT (see Section 3.1). In this approach, the optimization algorithm is used to compute a single Top-1 model, and then Algorithm 2 provides a Top-1 solution using this first Top-1 model. Moreover, a generalization of this approach to Top-k instances can be obtained by using a recursive algorithm. Indeed, consider a Top-k instance $(\Phi, \succeq, X, k)$ with a complete preference relation. We first compute a solution S of the Top-1 instance $(\Phi, \succeq, X, 1)$ using the approach that combines an optimization algorithm with Algorithm 2. Then, if $k - |S| \geq 1$ and $S \neq \emptyset$, a recursive call is made on the instance $(\Phi \wedge \bigwedge_{\mathcal{M} \in S} \overline{\mathcal{M}_{|X}}, \succeq, X, k - |S|)$ to compute a solution S' ($S \cup S'$ is a Top-k solution of $(\Phi, \succeq, X, k)$), otherwise S is returned.

The correctness of the recursive algorithm sketched above is a consequence of the following proposition:

Proposition 3. *Let $I_1 = (\Phi, \succeq, X, k)$ be a Top-k SAT instance and S a Top-1 solution of $(\Phi, \succeq, X, 1)$. If $k - |S| \geq 1$ and $S \neq \emptyset$ then $S \uplus S'$ is a Top-k solution of I_1, where S' is a Top-$(k - |S|)$ solution of the instance $I_2 = (\Phi \wedge \bigwedge_{\mathcal{M} \in S} \overline{\mathcal{M}_{|X}}, \succeq, X, k - |S|)$ and $\uplus$ is the disjoint union operator; otherwise, S is a Top-k solution of I_1.*

Proof. It is worth noting that if $S = \emptyset$, then we know that Φ is unsatisfiable and there is no Top-k model. Moreover, if $k - |S| < 1$, then the Top-1 solution S contains at least k Top-1 models and it means that each Top-k model is a Top-1 model. Assume that $k - |S| \geq 1$ and $S \neq \emptyset$. Using the sub-formula $\bigwedge_{\mathcal{M} \in S} \overline{\mathcal{M}_{|X}}$, we know that (i) there is no model of $\Psi \equiv \Phi \wedge \bigwedge_{\mathcal{M} \in S} \overline{\mathcal{M}_{|X}}$ which

is a Top-1 model of Φ w.r.t. $\succeq$ and X. We also know that (ii) each Top-k model which is not a Top-1 model of Φ w.r.t. $\succeq$ and X is a model of Ψ, since we only exclude in Ψ the Top-1 models of Φ. Let $\mathcal{M}$ be a Top-$(k-|S|)$ model of Ψ w.r.t. $\succeq$ and X. Then, $|[P(\Phi \wedge \bigwedge_{\mathcal{M} \in S} \overline{\mathcal{M}_{|X}}, \mathcal{M}, \succeq)]^X| \leq k-|S|-1$ holds. Using Property (ii), we get $|[P(\Phi, \mathcal{M}, \succeq)]^X| \leq k-1$ and, as a consequence, $\mathcal{M}$ is a Top-k model of Φ w.r.t. $\succeq$ and X. To complete our proof, we have to show that each Top-k model of Φ w.r.t. $\succeq$ and X, which is not a Top-1 model, is a Top-$(k-|S|)$ model of Ψ w.r.t. $\succeq$ and X. Assume that there exists a Top-k model $\mathcal{M}$ of Φ which is not a Top-1 model of Φ and not a Top-$(k-|S|)$ model of Ψ w.r.t. $\succeq$ and X. Using Property (i) and the fact that $\mathcal{M}$ is not a Top-$(k-|S|)$ model of Ψ, we get $|[P(\Phi \wedge \bigwedge_{\mathcal{M} \in S} \overline{\mathcal{M}_{|X}}, \mathcal{M}, \succeq)]^X| > k-|S|-1$ and, as a consequence, $|[P(\Phi, \mathcal{M}, \succeq)]^X| > k-1$ holds. We thus get a contradiction. □

Let us consider, for instance, a generalization of Partial Max-SAT problem, called Top-k Partial Max-SAT. It consists in computing a Top-k solution for an instance $\mathcal{T} = (\Phi_h, \succeq_{\Phi_s}, Var(\Phi_h \wedge \Phi_s), k)$ where $\Phi_h \wedge \Phi_s$ is a Partial Max-SAT instance (Φ_h and Φ_s are the hard part and the soft part respectively) and $\succeq_{\Phi_s}$ is the preference relation defined in Section 3.1. An algorithm for computing a Top-1 solution can be defined as a combination of an algorithm for Partial Max-SAT and Algorithm 2 as described above. Indeed, an algorithm for Partial Max-SAT is first used to find a Top-1 model, and then, Algorithm 2 is used to find a Top-1 solution. Using this approach, we can also derive an algorithm for Top-k Partial Max-SAT using our recursive approach.

In the sequel, we provide our SAT based encodings of itemset and sequence mining problems. We use standard propositional formula instead of CNF formula. Let us recall that every propositional formula can be translated in linear time to a CNF form using the well known extension principle (with fresh propositional variables) [35].

4. An application of Top-k SAT in itemset mining

The problem of mining frequent itemsets is well-known and essential in data mining [1], knowledge discovery and data analysis. Note that several data mining tasks are closely related to the itemset mining problem such as the ones of association rule mining, frequent pattern mining in sequence data, data clustering, etc. Recently, De Raedt et al. in [24, 31] proposed the

first constraint programming (CP) based data mining framework for itemset mining. This new framework offers a declarative and flexible representation model. It allows data mining problems to benefit from several generic and efficient CP solving techniques. This study leads to the first CP approach for itemset mining displaying nice declarative opportunities.

In itemset mining problem, the notion of Top-k frequent itemsets is introduced as an alternative to finding the appropriate value for the minimum support threshold. In this section, we propose a SAT-based encoding for enumerating all closed itemsets. Then we use this encoding in the Top-k SAT problem for computing all Top-k frequent closed itemsets.
In this work, we mainly consider the problem of mining Top-k frequent closed itemsets of minimum length *min*. In this problem, we consider that the minimum support threshold λ is not known.

Definition 4 ($\mathcal{FCI}^k_{min}$)**.** *Let k and min be strictly positive integers. The problem of mining Top-k frequent closed itemsets $\mathcal{FCI}^k_{min}$ consists in computing all closed itemsets of length at least min such that, for each one, there exist no more than $k-1$ closed itemsets of length at least min with supports greater than its support.*

4.1. SAT-based encoding for $\mathcal{FCI}^k_{min}$

We here propose an encoding of $\mathcal{FCI}^k_{min}$ in Top-k SAT problem. Let $\mathcal{I}$ be a set of items, $\mathcal{D} = \{(0, t_i), \ldots, (n-1, t_{n-1})\}$ a transaction database over $\mathcal{I}$, and k and *min* strictly positive integers. In order to define our encoding, we associate to each item a in $\mathcal{I}$ a propositional variable p_a. These propositional variables encode the candidate itemset $I \subseteq \mathcal{I}$, i.e., p_a is *true* iff $a \in I$. Moreover, for all $i \in \{0, \ldots, n-1\}$, we associate to the *i-th* transaction in $\mathcal{D}$ a propositional variable b_i. These propositional variables allow us to reason about the cover of the candidate itemset.

We first propose a constraint allowing to consider only the itemsets of length at least *min*. It corresponds to a cardinality constraint:

$$\sum_{a \in \mathcal{I}} p_a \geq min \tag{1}$$

We now introduce a constraint allowing to capture all the transactions where

the candidate itemset does not appear:

$$\bigwedge_{i=0}^{n-1}(b_i \leftrightarrow \bigvee_{a \in \mathcal{I} \setminus t_i} p_a) \tag{2}$$

This constraint means that b_i is true if and only if the candidate itemset is not in t_i.

By the following constraint, we force the candidate itemset to be closed:

$$\bigwedge_{a \in \mathcal{I}}(\bigwedge_{i=0}^{n-1} \overline{b_i} \rightarrow a \in t_i) \rightarrow p_a \tag{3}$$

This formula means that if $\mathcal{C}(I) = \mathcal{C}(I \cup \{a\})$ where I is the candidate itemset, then $a \in I$ holds. In other words, when the transactions containing the candidate itemset (b_i is $false$) also contain the item a ($a \in t_i$), the candidate itemset can be extended by adding the item a (p_a is $true$).

Proposition 4. *The set of models of* (1) ∧ (2) ∧ (3) *corresponds to the set of closed itemsets of size at least min.*

Proof.
Part ⇒. Let $\mathcal{M}$ be a model of (1)∧(2)∧(3) and I its corresponding itemset, i.e., $I = \{a \in \mathcal{I} \mid p_a \text{ is } true\}$. Then, using (1), we know that the size of I is greater than or equal to min. Moreover, using (2) ∧ (3), we know that, for all $a \in \mathcal{I}$, if $\mathcal{C}(I) = \mathcal{C}(I \cup \{a\})$ then $a \in I$ holds. Thus, we obtain that I is a closed itemset.
Part ⇐. Let I be a closed itemset of size greater than or equal to min. Then, we define its associated Boolean interpretation $\mathcal{M}$ as follows:

- for all $a \in \mathcal{I}$, $\mathcal{M}(p_a) = 1$ if and only if $a \in I$; and
- for all $i \in \{0, \ldots, n-1\}$, $\mathcal{M}(b_i) = 0$ if and only if t_i supports I.

Clearly, $\mathcal{M}$ satisfies the constraint (1), since the size of I is greater or equal to min. Moreover, we know that the set of all the transactions supporting I is $\{t_i \in \mathcal{D} \mid \mathcal{M}(b_i) = 0\}$. Hence, $\mathcal{M}$ satisfies the constraint (2). Using the fact that I is a closed itemset, we have, for all $a \in \mathcal{I}$, if if $\mathcal{C}(I) = \mathcal{C}(I \cup \{a\})$ then $a \in I$ holds. Therefore, $\mathcal{M}$ satisfies the constraint (3). □

In this context, computing the Top-k closed itemsets of length at least min corresponds to computing the Top-k models of $\Phi \equiv (1) \wedge (2) \wedge (3)$ with respect to $\succeq_B$ and $X = \{p_a | a \in \mathcal{I}\}$, where $B = \{b_0, \ldots, b_{n-1}\}$ and $\succeq_B$ is defined as follows: $\mathcal{M} \succeq_B \mathcal{M}'$ if and only if $|\mathcal{M}(B)| \leq |\mathcal{M}'(B)|$. This preference relation means that a model $\mathcal{M}$ is more preferred than a model $\mathcal{M}'$ if the itemset $\{a \in \mathcal{I} \mid \mathcal{M}(p_a) = 1\}$ is more frequent than $\{a \in \mathcal{I} \mid \mathcal{M}'(p_a) = 1\}$. It is a δ-preference relation since one can define the bound function $\delta^{\Phi}_{\succeq_B}$ as follows:

$$\delta^{\Phi}_{\succeq_B}(\mathcal{M}) = (\sum_{i=0}^{n-1} b_i \leq |\mathcal{M}(B)|)$$

Indeed, this formula allows us to have models corresponding to closed itemsets with supports greater than or equal to the support of the closed itemset obtained from $\mathcal{M}$.

4.2. Some Variants of $\mathcal{FCI}^k_{min}$

In this section, our aim is to illustrate the nice declarative aspects of our proposed framework. To this end, we simply consider variations of $\mathcal{FCI}^k_{min}$, and show that their encodings can be obtained by slight modifications of that of $\mathcal{FCI}^k_{min}$.

4.2.1. Variant 1 ($\mathcal{FCI}^k_{(min,max)}$)

In this variant, we consider the problem of mining Top-k closed itemsets of size between min and max. This variant can be used to reduce the size of the output by focusing the attention on a size interval (min, max).

Our encoding in this case is obtained by adding to (1), (2) and (3) the following constraint:

$$\sum_{a \in \mathcal{I}} p_a \leq max \tag{4}$$

In this case, we use the δ-preference relation $\succeq_B$ defined previously.

4.2.2. Variant 2 ($\mathcal{FCI}^k_{\lambda}$)

Let us now propose an encoding of the problem of mining Top-k closed itemsets of supports at least λ (minimal support threshold). In this context, a Top-k closed itemset is a closed itemset such that, for each one, there exist no more than $k-1$ closed itemsets of size greater than its size. In

the same way as maximal frequent itemsets mining, the mining task $\mathcal{FCI}^k_\lambda$ allows us to focus on the largest frequent itemsets. Let us recall that a frequent itemset is maximal if all its supersets are infrequent. Formally, given a transaction database $\mathcal{D}$, a minimal support threshold λ and a frequent itemset I ($\mathcal{S}(I,\mathcal{D}) \geq \lambda$), I is said to be maximal w.r.t. λ if, for all J with $I \subset J$, $\mathcal{S}(J,\mathcal{D}) < \lambda$. $\mathcal{FCI}^k_\lambda$ can be used in cases where the size is related to the amount of information.

Our encoding in this case is obtained by adding to (2) and (3) the following constraint:

$$\sum_{i=0}^{n} \overline{b_i} \geq \lambda \tag{5}$$

The preference relation used in this case is $\succeq_I$ defined as follows: $\mathcal{M} \succeq_I \mathcal{M}'$ if and only if $|\mathcal{M}(I)| \geq |\mathcal{M}'(I)|$. It means that a model $\mathcal{M}$ is more preferred than a model $\mathcal{M}'$ if the size of the itemset $\{a \in \mathcal{I} \mid \mathcal{M}(p_a) = 1\}$ is greater than the size of $\{a \in \mathcal{I} \mid \mathcal{M}'(p_a) = 1\}$. It is a δ-preference relation because the bound function $\delta^{\Phi}_{\succeq_I}$ ($\Phi \equiv (2) \wedge (3) \wedge (5)$) can be defined as follows:

$$\delta^{\Phi}_{\succeq_I}(\mathcal{M}) = \sum_{a \in I} p_a \geq |\mathcal{M}(I)|$$

5. An application of Top-k SAT in sequence mining

5.1. Frequent pattern mining in a sequence of items ($\mathcal{FPS}$)

In this section, we present the datamining problem of enumerating frequent and closed patterns with wildcards in a sequence of items [3, 4, 5].

Sequences of items. Let Σ be a finite set of items, called alphabet. A *sequence of items* s over Σ is a simple sequence of symbols $s_0 \cdots s_{n-1}$ belonging to Σ. We denote by $|s|$ its length and by $\mathcal{P}_s$ the set $\{0, \ldots |s|-1\}$ of all the locations of its symbols. *A wildcard* is a new symbol $\circ$ which is not in Σ. This symbol matches any symbol of the alphabet.

Pattern. A *pattern* over Σ is a sequence $p = p_0 \ldots p_{m-1}$, where $p_0 \in \Sigma$, $p_{m-1} \in \Sigma$ and $p_i \in \Sigma \cup \{\circ\}$ for $i = 1, \ldots, m-2$. We say that p is included in $s = s_0 \ldots s_{n-1}$ at the location $l \in \mathcal{P}_s$, denoted $p \sqsubseteq_l s$, if $\forall i \in \{0 \ldots m-1\}$, $p_i = s_{l+i}$ or $p_i = \circ$. We also say that p is included in s, denoted $p \sqsubseteq s$, if $\exists l \in \mathcal{P}_s$ such that $p \sqsubseteq_l s$. The *cover* of p in s is defined as the set

$\mathcal{L}_s(p) = \{l \in \mathcal{P}_s | p \sqsubseteq_l s\}$. Moreover, The *support* of p in s is defined as the value $|\mathcal{L}_s(p)|$.

$\mathcal{FPS}$ problem. Let s be a sequence, p a pattern and $\lambda \geq 1$ a minimal support threshold, called also a quorum. We say that p is a *frequent pattern in s w.r.t. λ* if $|\mathcal{L}_s(p)| \geq \lambda$. *The frequent pattern mining problem in a sequence of items ($\mathcal{FPS}$)* consists in computing the set $\mathcal{FPS}(s, \lambda)$ of all the frequent patterns w.r.t. λ.

For instance, consider the sequence $s = aaccbcabcba$ and the pattern $p = a \circ c$. We have $\mathcal{L}_s(p) = \{0, 1, 6\}$, since $p \sqsubseteq_0 s$, $p \sqsubseteq_1 s$ and $p \sqsubseteq_6 s$. In this case, if we consider that the minimal support threshold is equal to the value 3, then the pattern p is a frequent pattern of s.

Closed Patterns. A frequent pattern p of a sequence s is said to be *closed* if for any frequent pattern q satisfying $p \sqsubset q$, there is no integer d such that $\mathcal{L}_s(q) = \mathcal{L}_s(p) + d$, where $\mathcal{L}_s(p) + d = \{l + d | l \in \mathcal{L}_s(p)\}$. Clearly, the set of closed frequent patterns is a condensed representation of the set of frequent patterns. Indeed, the frequent patterns can be obtained from the closed ones by replacing items with wildcards.

Note that if p_1 and p_2 are two patterns such that $p_1 \sqsubseteq p_2$, then if $|\mathcal{L}_s(p_2)| \geq \lambda$ then $|\mathcal{L}_s(p_1)| \geq \lambda$. This property is called anti-monotonicity.

Definition 5 ($\mathcal{FCPS}^k_{min}$)**.** *Let k and min be strictly positive integers. The problem of mining Top-k frequent closed patterns in a sequence $\mathcal{FCPS}^k_{min}$ consists in computing all closed patterns with at least min items such that, for each one, there exist no more than $k-1$ closed patterns with at least min items and with supports greater than its support.*

5.2. SAT-based encoding for $\mathcal{FCPS}^k_{min}$

Let $\Sigma = \{a_1, \ldots, a_m\}$ be an alphabet, s a sequence over Σ of length n and λ a minimal support threshold. We associate to each character a appearing in s a set of k_a propositional variables $p_{a,0}, \ldots, p_{a,(k_a-1)}$ where $k_a = max(\mathcal{L}_s(a)) + 1$. The variable $p_{a,i}$ means that a is in the candidate pattern at the location i. In fact, that explains why we associate only $max(\mathcal{L}_s(a)) + 1$ variables to each character a, because $\{0, \ldots, max(\mathcal{L}_s(a))\}$ corresponds to the set of all possible locations of a in the candidate patterns.

We first need to encode that the first symbol must be a solid character (different from the wildcard symbol). This property is expressed by the following simple clause:

$$\bigvee_{a \in \Sigma} p_{a,0} \tag{6}$$

The following constraints allow us to capture all the locations where the candidate pattern appears:

$$\bigwedge_{a \in \Sigma, 0 \leq l \leq n-1, 0 \leq i \leq k_a - 1} (p_{a,i} \wedge s_{l+i} \neq a) \rightarrow b_l \tag{7}$$

$$\bigwedge_{l=0}^{n-1} (b_l \rightarrow \bigvee_{a \in \Sigma, 0 \leq i \leq k_a - 1} (p_{a,i} \wedge s_{l+i} \neq a)) \tag{8}$$

Indeed, the previous constraints allow us to obtain that, if the Boolean interpretation $\mathcal{M}$ is a model of the constraints (6), (7) and (8), then the candidate pattern that corresponds to $\mathcal{M}$ appears only in the locations $\{0 \leq l \leq n-1 | \mathcal{M}(b_l) = 0\}$.

In order to consider the patterns with at least min items (solid characters), we just have to add the following constraint:

$$\sum_{a \in \Sigma, 0 \leq i \leq k_a - 1} p_{a,i} \geq min \tag{9}$$

Now, we introduce a necessary, but not sufficient, constraint, w.r.t. the previous constraints, for obtaining a closed frequent pattern:

$$\bigwedge_{a \in \Sigma, 0 \leq i \leq k_a - 1} (\bigwedge_{l=0}^{n-1} \overline{b_l} \rightarrow s_{l+i} = a) \rightarrow p_{a,i} \tag{10}$$

Intuitively, the previous constraint maximizes the number of symbols different from wildcard on the right side of the symbol represented by the propositional variable having 0 as index.

Then, we introduce a constraint allowing to maximize the number of symbols different from wildcard on the left side of the symbol represented by the

propositional variable having 0 as index:

$$\bigwedge_{a\in\Sigma, 1\leq i\leq k'_a} \neg(\bigwedge_{l=0}^{n-1} \overline{b_l} \rightarrow s_{l-i} = a) \tag{11}$$

where $k'_a = n - min(\mathcal{L}_s(a)) - 1$ for $a \in \Sigma$. for all $a \in \Sigma$, the integers from 1 to k'_a allows us to capture all the possible locations of a on the left side of the candidate pattern.

Proposition 5. *The set of models of* (6)∧(7)∧(8)∧(9)∧(10)∧(11) *correspond to the set of closed patterns with at least min items.*

Proof.
Part ⇒. Let $\mathcal{M}$ be a model of $(6) \wedge (7) \wedge (8) \wedge (9) \wedge (10) \wedge (11)$ and p its corresponding patterns. Then, using the constraints (6), (7), (8) and (9), we know that p contains more than or equal to *min* items and appears only in the locations $\{0 \leq l \leq n-1 | \mathcal{M}(b_l) = 0\}$. Moreover, using (10) and (11), we have, for all p_1, p_2 and a with $p = p_1 \circ p_2$, $\mathcal{L}_s(p) \neq \mathcal{L}_s(p_1 a p_2)$ holds. Thus, we obtain that p is a closed pattern.
Part ⇐. Let p be a closed pattern with at least *min* items. Then, we define its associated Boolean interpretation $\mathcal{M}$ as follows:

- for all $a \in \Sigma$ and $0 \leq i < k_a$, $\mathcal{M}(p_{a,i}) = 1$ if and only if a is in p at the location i; and
- for all $i \in \{0, \ldots, n-1\}$, $\mathcal{M}(b_i) = 0$ if and only if p appears at the location i.

One can easily see that $\mathcal{M}$ satisfies the constraints (6), (7), (8) and (9), since p is a pattern appearing in s that contains a number of items greater than or equal to *min*. Furthermore, using the fact that I is a closed pattern, we obtain that $\mathcal{M}$ satisfies the constraints (10) and (11). Indeed, we have, for all p_1, p_2 and a with $p = p_1 \circ p_2$, $\mathcal{L}_s(p) \neq \mathcal{L}_s(p_1 a p_2)$ holds, and this property implies that $\mathcal{M}$ satisfies $(10) \wedge (11)$. □

The problem $\mathcal{FCPS}^k_{min}$ is encoded as the problem of computing the Top-k models of $(6) \wedge (7) \wedge (8) \wedge (9) \wedge (10) \wedge (11)$ with respect to $\succeq_B$ and $X = \{p_a | a \in \Sigma\}$, where $B = \{b_0, \ldots, b_{n-1}\}$ and $\succeq_B$ is the δ-preference relation defined in the same way as that in the case of $\mathcal{FCI}^k_{min}$, i.e., $\mathcal{M} \succeq_B \mathcal{M}'$ if and only if $|\mathcal{M}(B)| \leq |\mathcal{M}'(B)|$.

5.3. Frequent Pattern Mining in a Sequence of Itemsets ($\mathcal{FPSI}$)

We here define a variant of the problem of discovering patterns with wildcards in a sequence, by considering a sequence of itemsets instead of a sequence of items. The role of wildcard symbol is nicely played by the empty itemset as it match any itemset. This problem admits some similarities and differences with the classical sequential pattern mining problem introduced in [33]. The main difference resides in the definition of the notion of subsequence (inclusion), where empty itemsets are used as wildcards, and in the use or not of a single or several sequences.

Our goal in this section is to illustrate the flexibility of our framework and its nice declarative aspects. As we can see below, a change in the problem specification induces a small change in the model. Considering sequences of itemsets is also interesting from the practical side. In the literature, several data mining papers consider sequences of itemsets (e.g. [44]). Such data can be found in several applications including web logs, trading where one is interested in analyzing the consumer purchases over time. Mining frequent patterns with periodic wildcard gaps can flexibly reflect the sequential behaviors and is often exhibited in many real-world applications. For example, in business, retail companies may want to know what products customers will usually purchase at regular time intervals rather than in continuous time according to time gaps [45]. As mentioned in the introduction, in biology, patterns with wildcards are redeemed as having significant biological and medical values. Minimum and maximum time gaps are also introduced to constrain two items/itemsets to occur neither too close nor too far apart in time.

Sequence of itemset. A *sequence of itemsets* s over an alphabet Σ is defined as a sequence $s_0, \ldots, s_{n-1}$, where $s_i \subseteq \Sigma$ for $i = 0, \ldots, n-1$. Similarly to the sequences of items, we denote by $|s|$ its length ($|s| = n$) and by $\mathcal{P}_s$ the set $\{0, \ldots |s| - 1\}$ of the locations.

Pattern. A *pattern* $p = p_0, \ldots, p_{m-1}$ over Σ is also defined as a sequence of itemsets where the first and the last elements are different from the empty itemset. In this context, let us mention that we do not need the wildcard symbol. Indeed, one can use the empty itemset to match any itemset. Furthermore, we say that p is included in $s = s_0 \ldots s_{n-1}$, denoted $p \sqsubseteq_l s$, at the location $l \in \mathcal{P}_s$ if $\forall i \in \{0 \ldots m-1\}$, $p_i \subseteq s_{l+i}$. The relation $\sqsubseteq$ and the set $\mathcal{L}_s(p)$ are defined in the same way as in the case of the sequences of items.

The *cover* (resp. *support*) of p in s is defined as the set $\mathcal{L}_s(p)$ (resp. as the value $|\mathcal{L}_s(p)|$).

The frequent and closed patterns are also defined in the same way as in the sequences of items. For instance, a frequent pattern p of a sequence s is said to be *closed* if for any frequent pattern q satisfying $p \sqsubset q$, there is no integer d such that $\mathcal{L}_s(q) = \mathcal{L}_s(p) + d$, where $\mathcal{L}_s(p) + d = \{l + d | l \in \mathcal{L}_s(p)\}$. The frequent patterns can be obtained from the closed ones by replacing itemsets with their subsets.

For example, consider the sequence of itemsets $s = \{a, b\}, \{a, b\}, \{c, d\}, \{c, e\}, \{f\}, \{g\}, \{d\}, \{a, b, d\}, \{f\}, \{c\}$ and the pattern $p = \{a, b\}, \{\}, \{c\}$. If we set the minimal support threshold to 3, then p is a frequent pattern in s, since $\mathcal{L}_s(p) = \{0, 1, 7\}$. The pattern p is also a closed frequent pattern, but $p' = \{a\}, \{\}, \{c\}$ is not closed, since $p' \sqsubset p$.

The pattern mining task that we consider in the sequences of itemsets allows to exhibit a high degree of self similarity for better understandings of large volumes of data. For instance, a sequence of itemsets can be seen as a record of the articles bought by a customer over a period of time. In such a case, a frequent pattern could be "the customer bought acetylsalicylic acid two days after buying beer and wine in 20% of the days from 2008 to 2012".

The problem of mining Top-k frequent closed patterns in a sequence of itemsets $\mathcal{FCPSI}^k_{min}$ is defined in the same way as $\mathcal{FCPS}^k_{min}$.

5.4. SAT-based encoding of $\mathcal{FCPSI}^k_{min}$

Our encoding of the problem $\mathcal{FCPSI}^k_{min}$ can be easily obtained from the one of $\mathcal{FCPS}^k_{min}$. We only have to replace the equalities (resp. inequalities) of the form $s_{l\pm i} \neq a$ (resp. $s_{l\pm i} = a$) with $a \notin s_{l\pm i}$ (resp. $a \in s_{l\pm i}$):

$$\bigvee_{a \in \Sigma} p_{a,0} \tag{12}$$

$$\bigwedge_{a \in \Sigma, 0 \leq l \leq n-1, 0 \leq i \leq k_a - 1} (p_{a,i} \wedge a \notin s_{l+i}) \rightarrow b_l \tag{13}$$

$$\bigwedge_{l=0}^{n-1} (b_l \rightarrow \bigvee_{a \in \Sigma, 0 \leq i \leq k_a - 1} (p_{a,i} \wedge a \notin s_{l+i})) \tag{14}$$

$$\sum_{l=0}^{n-1} b_l \leq n - \lambda \tag{15}$$

$$\bigwedge_{a\in\Sigma,0\leq i\leq k_a-1} (\bigwedge_{l=0}^{n-1} \overline{b_l} \rightarrow a \in s_{l+i}) \rightarrow p_{a,i} \tag{16}$$

$$\bigwedge_{a\in\Sigma,1\leq i\leq k'_a} \neg(\bigwedge_{l=0}^{n-1} \overline{b_l} \rightarrow a \in s_{l-i}) \tag{17}$$

Similarly to $\mathcal{FCPS}^k_{min}$, the problem $\mathcal{FCPSI}^k_{min}$ is encoded as the problem of computing the Top-k models of $(12) \wedge (13) \wedge (14) \wedge (15) \wedge (16) \wedge (17)$ with respect to $\succeq_B$ and $X = \{p_a | a \in \Sigma\}$, where $B = \{b_0, \ldots, b_{n-1}\}$.

The slight modification of our encoding in the case of the sequences of items in order to obtain encoding for the sequences of itemsets clearly shows the high flexibility of our proposed framework.

6. Implementation and Experiments

In this section, we carried out an experimental evaluation of the performance of our Algorithm for Top-k SAT empirically. The primary goal is to assess the declarativity and the effectiveness of our proposed framework. For this purpose, we consider the problems $\mathcal{FCIM}^k_{min}$ and $\mathcal{FCPS}^k_{min}$.

For our experiments, we implemented the Algorithm 1 (Top-k) on the top of the state-of-the-art SAT solver MiniSAT 2.2 [1] adapted to efficiently enumerate models of a propositional formula. Compared to our previous version, the new model enumeration algorithm is based on a DPLL based procedure, without adding blocking and learned clauses [46]. We also set the decision literal polarity to false by default as it is clearly the best option in practice. This version is significantly better than our previous model enumerator implemented using a CDCL based solver with blocking clauses [26].

The cardinality constraints involved in our SAT encodings, are managed dynamically during search. In other words, similarly to constraint programming, a propagator is associated to a cardinality constraint, obtained by

[1] MiniSAT: http://minisat.se/

maintaining the sum of its assigned variables. Managing cardinality constraints on the fly outperforms our previous implementation [34] where we used the sorting networks, one of the state-of-the-art encoding of the cardinality constraint to CNF proposed in [39].

For the problem $\mathcal{FCIM}^k_{min}$, we considered a variety of datasets (18 data instances) taken from the `FIMI` repository[2] and `CP4IM`[3]. Regarding the problem $\mathcal{FCPS}^k_{min}$, we used two different datasets:

1. *Bioinformatics*: proteinic data encoded as a sequence of items, where an item is an amino-acid[4];
2. *Computer security:* user data drawn from the command histories of UNIX computer users[5] [47].

All the experiments were done on Intel core i7 machine with 8GB of RAM running at 1.7 Ghz.

Concerning the problem $\mathcal{FCIM}^k_{min}$, Table 1 details the characteristics of the different transaction databases ($\mathcal{D}$). The first column mentions the name of the considered data instance. In the second and third column, we give the size of $\mathcal{D}$ in terms of number of transactions (#trans) and number of items (#items) respectively. The fourth column shows the density (dens) of the transaction database, defined as the percentage of 1's in $\mathcal{D}$. The panel of datasets ranges from sparse (e.g. mushroom) to dense ones (e.g. Hepatitis). Finally, in the two last columns, we give the size of the CNF encoding (#vars, #clauses) of $\mathcal{FCIM}^k_{min}$. As we can see, our proposed encoding leads to CNF formula of reasonable size. The maximum size is obtained for the instance `Connect` (67815 variables and 5877720 clauses).

In order to analyze the behavior of our Top-k algorithm on $\mathcal{FCIM}^k_{min}$, we set the minimum length *min* of the itemsets to 1, while the value of k is varied from 100 to 100000.

In Table 2, we provide the number of Top-k frequent closed itemsets for each data instance according to different values of k. We also provide in parenthesis the total number of models including Top-k and intermediary non Top-k models, found by the model enumerator.

[2]FIMI: http://fimi.ua.ac.be/data/

[3]CP4IM: http://dtai.cs.kuleuven.be/CP4IM/datasets/

[4]http://www.biomedcentral.com/1471-2105/11/175/additional/

[5]http://kdd.ics.uci.edu/databases/UNIX user data/

instance	#trans	#items	$dens(\%)$	#vars	#clauses
Zoo-1	101	36	44	173	2196
Hepatitis	137	68	50	273	4934
Lymph	148	68	40	284	6355
audiology	216	148	45	508	17575
Heart-cleveland	296	95	47	486	15289
Primary-tumor	336	31	48	398	5777
Vote	435	48	33	531	14454
Soybean	650	50	32	730	22153
Australian-credit	653	125	41	901	48573
Anneal	812	93	45	990	39157
Tic-tac-toe	958	27	33	1012	18259
German-credit	1000	112	34	1220	73223
Kr-vs-kp	3196	73	49	3342	121597
Hypothyroid	3247	88	49	3419	143043
Chess	3196	75	49	3346	124797
Splice-1	3190	287	21	3764	727897
Mushroom	8124	119	18	8348	747635
Connect	67558	129	33	67815	5877720

Table 1: Characteristics of the datasets

In general, the number of Top-k frequent closed itemsets is slightly around the value of k. We can also observe that there is only one data instance (`Audiology`), where the number of Top-k frequent closed itemsets is significantly greater than k. Let us note that for a given data instance involving a total of m frequent closed itemsets, if k is greater than m (e.g., `Zoo-1, Soybean, Tic-tac-toe`) then the number of Top-k models remains equal to m. Let us also note that computing the Top-k models can leads to the same number for different values of k (e.g., `Audiology`). To illustrate such a particular case, given a data instance with a number of Top-1 models equal to 100. In this case, the number of top-k models for each value of k less than 100 remains equal to 100. However, the total number of models necessary to the generation of the Top-k models can vary with k. Indeed, the constraints allowing us to cut the non Top-k models is activated when the first k models are generated. Consequently, the search tree might depend on the value of k. As a summary, enumerating Top-k is clearly a method of choice for keeping the size of the output under control.

We compared, our method with a LCM algorithm proposed by Takeaki Uno et al in [48], one of the best specialized algorithm, using the Top-k option. In Table 3, we provide the CPU time in seconds needed by LCM(Top-k option) and $\mathcal{FCIM}_1^k$ respectively (separated by the slash symbol) for different values of k varying from 100 to 100000.

First, the CPU time needed for computing the Top-k models increases, in general, with k. The `Splice-1` dataset with low density, is the most challenging instance for our approach. Our algorithm computes the Top-100000

instance/k	100	1000	5000	10000	50000	75000	100000
Australian-credit	103 (261)	1013 (2717)	5070 (19001)	10072 (39142)	50389 (215691)	76975 (330280)	100569 (435093)
Heart-cleveland	103 (324)	1021 (3909)	5172 (21855)	10507 (43641)	51841 (219591)	78926 (332703)	100678 (448248)
Primary-tumor	101 (188)	1005 (2030)	5007 (10935)	10242 (20873)	50039 (71173)	77336 (85003)	87231 (87231)
Anneal	100 (268)	1004 (2942)	5029 (12650)	10148 (24745)	50403 (115708)	75524 (161776)	100996 (203796)
Hepatitis	121 (335)	1025 (3397)	5409 (20211)	11062 (42950)	51823 (191396)	83047 (274096)	104836 (346937)
Zoo-1	103 (282)	1027 (1991)	4568 (4568)	4568 (4568)	4568 (4568)	4568 (4568)	4568 (4568)
Lymph	100 (246)	1004 (2415)	5226 (12583)	10356 (23549)	51862 (89372)	79185 (115912)	108720 (135816)
Audiology	221 (929)	1583 (10745)	8615 (52307)	38202 (107407)	144016 (496001)	144016 (716473)	144016 (964565)
Soybean	100 (207)	1029 (2043)	5057 (9509)	10041 (17014)	31760 (31760)	31760 (31760)	31760 (31760)
Tic-tac-toe	103 (341)	1011 (3300)	5411 (13413)	10717 (22568)	42712 (42712)	42712 (42712)	42712 (42712)
Chess	100 (152)	1005 (2716)	5019 (17283)	10018 (35613)	50174 (182140)	75251 (285942)	100051 (377538)
Vote	105 (416)	1062 (4738)	5147 (21932)	10324 (39959)	50855 (126138)	75016 (159608)	101717 (182252)
German-credit	100 (302)	1013 (3889)	5031 (21814)	10126 (46044)	50460 (253203)	75145 (379607)	101001 (504992)
Kr-vs-kp	100 (150)	1005 (3589)	5019 (19929)	10018 (43726)	50174 (222536)	75251 (331336)	100051 (436830)
Mushroom	100 (202)	1000 (2316)	5016 (11632,)	10127 (22014)	50914 (94294)	75364 (133102)	108114 (160138)
Connect	100 (181)	1000 (2084)	5001 (11127)	10001 (22300)	50001 (141083)	75002 (223922)	100004 (325512)
Splice-1	102 (450)	1005 (5450)	5041 (30682)	10217 (64958)	50700 (90941)	75670 (145505)	100424 (190976)
Hypotyroid	100 (251)	1004 (1881)	5035 (11364)	10031 (22921)	50188 (110745)	75239 (161237)	100013 (215279)

Table 2: Number of Top-k Models and Total number of Enumerated Models

in 557.2 seconds. This is not the case for `Mushroom` which is even more sparse (density of 18%). Our experimental evaluation clearly shows that finding the Top-k models (the most interesting ones) can be computed efficiently for small values of k. For example, on most datasets the top-1000 models are computed in less than 1 second of CPU time, except for `Mushroom, Connect` and `Splice-1` dataset, where the Top-1000 are computed in less than 6 seconds. As expected, on all the datasets and all tested values of k, LCM is able to enumerate the Top-k frequent closed itemsets in less than 10 seconds. It is important to note, that our new implementation is able to compute the Top-k on most of the tested instances in less than 100 seconds except for `Splice-1`. Let us give some elements of explanation. It is first important to note that in our SAT based itemset mining encoding, the propositional variables associated to items form a strong backdoor set [49]. Any assignment of the variables from these set leads to a tractable sub-formula that can be decided by unit-propagation. Indeed, the propositional variables associated to transactions are dependent on the variables associated to items. As

a consequence, the size of the search tree depends on the number of items. The excessive CPU time obtained on the `Splice-1` data instance can be explained by its high number of items. Indeed, `Splice-1` involves 287 items, the highest number of items among all the data instances. For the `Mushroom` and `Connect` data instances, the number of items is lower, but the higher number of transactions increase the cost of unit propagation.

As a summary, comparatively, to our previously published results the gap with specialized approaches is significantly reduced. The results depicted in Table 3, demonstrate the competitiveness of our declarative approach with state-of-the-art LCM specialized algorithm.

instance/k	100	1000	5000	10000	50000	75000	100000
Chess	0.04/0.04	0.04/0.09	0.05 /0.23	0.07 /0.4	0.25/1.95	0.31/2.96	0.44/3.87
Heart-cleveland	0.01/0.01	0.03/0.05	0.05/0.16	0.13/0.29	0.53/1.2	0.8 /1.73	1.04/2.32
Primary-tumor	0.01/0.03	0.01/0.01	0.01/0.03	0.03/0.05	0.15 /0.14	0.18/0.17	0.28/0.16
Australian-credit	0.01/0.02	0.04/0.09	0.15/0.32	0.23/0.62	0.98/2.54	1.25/3.62	1.62/4.74
Anneal	0.07/2.57	0.02/0.04	0.04/0.11	0.07/0.18	0.28/0.67	0.37/0.91	0.55/1.18
Hepatitis	0.001/001	0.01/002	0.06/007	0.10/0.14	0.43/0.51	0.61/0.84	1.2/0.84
Lymph	0.001/0.005	0.01/0.01	0.03/0.05	0.08/0.08	0.38/0.3	0.59/0.41	0.8/0.48
Vote	0.01/0.03	0.03/0.15	0.1/0.44	0.16/0.66	0.47/1.41	0.59/1.60	0.71/1.77
Soybean	0.001/0.01	0.03/0.03	0.03/0.12	0.12/0.20	0.12/0.38	0.12/0.38	0.12/0.38
German-credit	0.01/0.07	0.04/0.17	0.11/0.55	0.17/1.10	0.76/3.94	1.1/5.48	1.43/7.01
Kr-vs-kp	0.01/0.07	0.04/0.15	0.11/0.37	0.12/0.66	0.25/3.02	0.43/4.3	0.61/5.67
Tic-tac-toe	0.001/0.03	0.01/0.14	0.05/0.39	0.06/0.57	0.17/0.77	0.17/0.76	0.17/0.78
Zoo-1	0.001/0.003	0.01/0.007	0.01/0.01	0.01/0.01	0.01/0.01	0.01/0.01	0.01/0.01
Audiology	0.01/0.07	0.01/0.06	0.15/0.26	0.43/0.52	1.71/2.32	2.10/3.20	2.69/4.23
Mushroom	0.01/0.73	0.08/2.3	0.17/5.8	0.26/9.1	0.74/43.48	1.01/59.47	1.03/72.99
Connect	0.81/3.81	1.18/5.6	1.55/10.59	2.64/15.13	2.21/55.61	2.56/83.09	2.8/102.32
Splice-1	0.07/2.57	0.32 /6.07	0.93 /24.77	1.75 /70.00	6.28 /332.67	8.32/451.09	10.27/557.92
Hypotyroid	0.03/0.08	0.05/0.10	0.07/0.24	0.1/0.39	0.34/1.45	0.39/2.03	0.57/2.65

Table 3: CPU time (seconds) to compute Top-k itemsets: LCM(Top-k option) / $\mathcal{FCIM}_1^k$

Concerning sequence mining, we are not aware of any available tool that enumerates the Top-k frequent closed patterns with wildcards in a sequence ($\mathcal{FCPS}_{min}^k$) as described in this paper. Consequently, our last experimental evaluation (see Table 4) aims to simply show the flexibility of our proposed declarative approach and its feasibility. However, to provide an idea on the performance gap between our Top-k sequence mining approach and specialized algorithms, we compare with MaxMotif the state-of-the-art sequence mining tool [5][6]. As MaxMotif does not provide a Top-k option, we proceed in two steps. In the first step, we generate all the frequent closed patterns and in a second step we generate the Top-k by sorting the patterns according to their frequencies.

Similarly to $\mathcal{FCIM}_{min}^k$, we set the minimum length min of the patterns

[6]maxMotif: http://research.nii.ac.jp/ uno/code/maxmotif.html

instance/k (#vars, #clauses)	100	250	500	1000	2500	5000	10000
User_200_ (3491, 685562)	1.97 (106, 348)	2.1 (392, 802)	2.24 (944, 1285)	2.19 (2084, 2084)	2.19 (2084, 2084)	2.19 (2084, 2084)	2.19 (2084, 2084)
User_400_ (12049, 4070356)	55.95 (113, 412)	61.32 (265, 1045)	64.60 (552, 2059)	73.36 (1832, 4130)	75.25 (3842, 9036)	77.34 (8803, 15893)	77.81 (21772, 25891)
Bioinfo_300_ (5582, 1640349)	8.66 (109, 490)	8.74 (491, 1059)	8.92 (1352, 1924)	9.57 (1352, 2806)	8.57 (3848, 3977)	9.51 (4105, 4105)	9.51 (4105, 4105)
Bioinfo_500_ (9712, 4749779)	66.00 (118, 440)	69.11 (297, 1127)	72.13 (852, 2161)	74.08 (1628, 3937)	79.27 (3436, 8809)	79.12 (9779, 14907)	79.39 (24174, 24323)
Bioinfo_650_ (12456, 7909598)	174.54 (109, 452)	185.08 (344, 1171)	196.69 (530, 2271)	200.19 (1280, 4290)	210.06 (4001, 10163)	222.96 (8759, 19013)	227.40 (26396 , 35137)

Table 4: Top-k frequent closed patterns in a sequence ($\mathcal{FCPS}_1^k$)

User_200_ (3491, 685562)	User_400_ (12049, 4070356)	Bioinfo_300_ (5582, 1640349)	Bioinfo_500_ (9712, 4749779)	Bioinfo_650_ (12456 , 7909598)
0.3 (2084)	5.16 (33670)	0.41 (4105)	4.41 (24623)	12.80 (59382)

Table 5: Top-10000 frequent closed patterns in a sequence (MaxMotif in two steps)

to 1, while the value of k is varied from 100 to 10000.

Let us first mention that on the considered data sets, the encoding of $\mathcal{FCPS}_{min}^k$ leads to large CNF formula comparatively to those considered in $\mathcal{FCIM}_{min}^k$ encoding. Indeed, for the sequence `User-400` containing 400 solid characters, the set of clauses required for its encoding exceeds 4 millions of clauses. The same observation can be made on the number of variables.

Similarly to $\mathcal{FCIM}_{min}^k$, the CPU time needed for computing the Top-k models increases, in general, with k. Although the model enumeration process using DPLL leads to an important improvement with respect to our first version of CDCL-based model enumerator [26]. In Table 4, for each k and for each instance, the CPU time in seconds needed to compute Top-k frequent closed patterns in a sequence is given. We also provide a couple (x, y) where x represents the number of Top-k models and y the total number of models including the intermediary models necessary to the generation of the Top-k models. We also mention for each data instance the number of variables (#vars) and clauses (#clauses) of its associated CNF encoding.

As expected, the numbers of Top-k patterns is close to k. We can also remark that we have to mine about a factor of 4 of non Top-k models to find the final Top-k models. Overall, less than 120 seconds is needed to enumerate all top-k patterns for different values of k.

Table 5 illustrates the performances of MaxMotif using two steps as explained above. We tested the five sequence data instances with $k = 10000$. For each data instance, we provide the cumulated CPU time of the two steps in seconds and the total number of frequent closed patterns found in the

first step (in parenthesis). As expected, to compute the Top-k, on all considered data instances, MaxMotif (in two steps) requires less than 13 seconds to enumerate the set of Top-10000 frequent closed patterns in a sequence.

Let us also note that mining Top-k frequent closed patterns in a sequence using Top-k SAT approach suffers from the size of the encoding. However, the advantage of our declarative approach is its ability to not enumerate all the Top-k frequent closed patterns through a dynamic use of constraints to cut the search tree.

As a summary, the experiments show clearly the feasibility of our proposed framework. It is clearly competitive on Top-k itemsets mining problem. However, on Top-k sequence mining, MaxMotif (in two steps) is several orders of magnitude better.

7. Conclusion and Perspectives

In this paper, we introduce a new problem, called Top-k SAT, defined as the problem of enumerating the Top-k models of a propositional formula. A Top-k model is a model having no more than $k-1$ models preferred to it with respect to the considered preference relation. We also show that Top-k SAT generalizes the two well-known problems: the partial Max-SAT problem and the problem of computing minimal models. A general algorithm for this problem is proposed and evaluated on the problem of enumerating Top-k patterns in data mining, namely, the problem of mining Top-k motifs in the transaction databases and in the sequences. In the case of mining sequence data, we introduce a natural extension of the problem to deal with the sequences of itemsets. Interestingly, its encoding to SAT is obtained with a slight modification of the SAT encoding of the problem dealing with the sequences of items.

While our new problem of computing the Top-k preferred models in Boolean satisfiability is flexible and declarative, there are a number of questions that deserve further research efforts. One direction is the study of (preferred/Top-k) model enumeration algorithm so as to achieve a further speedup of the runtime. This fundamental problem has not received a lot of attention in the SAT community, except some interesting works on enumerating minimal/preferred models. We also plan to investigate other variants of the considered data mining problems such as sequences of sequences of

items or itemsets. It would be interesting to extend our encodings with constraints on the form of the enumerated patterns (restriction on the number of consecutive wildcards, regular expressions, etc). Finally, on the Boolean satisfiability side, the design of efficient model generation procedures is an important issue for SAT-based datamining framework in general and to other important application domains. Finding a better approximation of the initial set of Top-k patterns to reduce the intermediary non Top-k patterns deserves to be investigated.

References

[1] R. Agrawal, T. Imielinski, A. N. Swami, Mining association rules between sets of items in large databases, in: ACM SIGMOD International Conference on Management of Data, ACM Press, Baltimore, 1993, pp. 207–216.

[2] A. Tiwari, R. Gupta, D. Agrawal, A survey on frequent pattern mining: Current status and challenging issues, Inform. Technol. J 9 (2010) 1278–1293.

[3] L. Parida, I. Rigoutsos, A. Floratos, D. Platt, Y. Gao, Pattern discovery on character sets and real-valued data: Linear bound on irredundant motifs and an efficient polynomial time algorithm, in: ACM-SIAM Symposium on Discrete Algorithms, 2000, pp. 297–308.

[4] N. Pisanti, M. Crochemore, R. Grossi, M. france Sagot, Bases of motifs for generating repeated patterns with wild cards, IEEE/ACM TCBB'2003 2 (2003) 2005.

[5] H. Arimura, T. Uno, An efficient polynomial space and polynomial delay algorithm for enumeration of maximal motifs in a sequence, Journal of Combinatorial Optimization 13 (2007) 243–262.

[6] A. W.-C. Fu, R. W. w. Kwong, J. Tang, Mining n-most interesting itemsets, in: Proceedings of the 12th International Symposium on MÈthodologies for Intelligent Systems (ISMIS 2000), Lecture Notes in Computer Science, Springer, 2000, pp. 59–67.

[7] J. Han, J. Wang, Y. Lu, P. Tzvetkov, Mining top-k frequent closed patterns without minimum support, in: Proceedings of the 2002 IEEE

International Conference on Data Mining (ICDM 2002), IEEE Computer Society, 2002, pp. 211–218.

[8] Y. Ke, J. Cheng, J. X. Yu, Top-k correlative graph mining, in: Proceedings of the SIAM International Conference on Data Mining (SDM 2009), 2009, pp. 1038–1049.

[9] E. Valari, M. Kontaki, A. N. Papadopoulos, Discovery of top-k dense subgraphs in dynamic graph collections, in: Proceedings of the 24th International Conference on Scientific and Statistical Database Management (SSDBM 2012), 2012, pp. 213–230.

[10] H. T. Lam, T. Calders, Mining top-k frequent items in a data stream with flexible sliding windows, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010), 2010, pp. 283–292.

[11] H. T. Lam, T. Calders, N. Pham, Online discovery of top-k similar motifs in time series data, in: Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011 (SDM 2011), 2011, pp. 1004–1015.

[12] Y. Shoham, Reasoning about change: time and causation from the standpoint of artificial intelligence, MIT Press, Cambridge, MA, USA, 1988.

[13] P. Meseguer, F. Rossi, T. Schiex, Soft constraints, in: P. v. B. F. Rossi, T. Walsh (Eds.), Handbook of Constraint Programming, Elsevier, 2006.

[14] C. Boutilier, R. I. Brafman, C. Domshlak, D. L. Poole, H. H. Hoos, CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements, Journal of Artificial Intelligence Research (JAIR) 21 (2004) 135–191.

[15] T. Walsh, Representing and reasoning with preferences, AI Magazine 28 (2007) 59–70.

[16] R. I. Brafman, C. Domshlak, Preference Handling - An Introductory Tutorial, AI Magazine 30 (2009) 58–86.

[17] C. Domshlak, E. Hüllermeier, S. Kaci, H. Prade, Preferences in AI: An overview, Artificial Intelligence 175 (2011) 1037–1052.

[18] S. Bistarelli, F. Bonchi, Soft constraint based pattern mining, Data & Knowledge Engineering 62 (2007) 118 – 137.

[19] S. de Amo, M. S. Diallo, C. T. Diop, A. Giacometti, H. D. Li, A. Soulet, Mining contextual preference rules for building user profiles, in: Proceedings of the 14th International Conference on Data Warehousing and Knowledge Discovery, DaWaK'12, 2012, pp. 229–242.

[20] W. Ugarte Rojas, P. Boizumault, S. Loudni, B. Crémilleux, A. Lepailleur, Mining (soft-) skypatterns using dynamic csp, in: Integration of AI and OR Techniques in Constraint Programming (CPAIOR'14), 2014, pp. 71–87.

[21] B. Négrevergne, A. Dries, T. Guns, S. Nijssen, Dominance programming for itemset mining, in: 2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013, 2013, pp. 557–566.

[22] E. D. Rosa, E. Giunchiglia, M. Maratea, Solving satisfiability problems with preferences, Constraints 15 (2010) 485–515.

[23] T. Castell, C. Cayrol, M. Cayrol, D. L. Berre, Using the davis and putnam procedure for an efficient computation of preferred models, in: ECAI, 1996, pp. 350–354.

[24] L. D. Raedt, T. Guns, S. Nijssen, Constraint programming for itemset mining, in: ACM SIGKDD, 2008, pp. 204–212.

[25] T. Guns, S. Nijssen, L. D. Raedt, Itemset mining: A constraint programming perspective, Artif. Intell. 175 (2011) 1951–1983.

[26] S. Jabbour, L. Sais, Y. Salhi, Boolean satisfiability for sequence mining, in: CIKM, 2013, pp. 649–658.

[27] H. Cambazard, T. Hadzic, B. O'Sullivan, Knowledge compilation for itemset mining, in: ECAI'10, 2010, pp. 1109–1110.

[28] M. Khiari, P. Boizumault, B. Crémilleux, Combining csp and constraint-based mining for pattern discovery, in: Computational Science and Its Applications – ICCSA 2010, 2010, pp. 432–447.

[29] J.-P. Metivier, P. Boizumault, B. Crémilleux, M. Khiari, S. Loudni, A Constraint-based Language for Declarative Pattern Discovery, in: Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, Vancouver, Canada, 2011, pp. 1112–1119.

[30] T. Guns, A. Dries, G. Tack, S. Nijssen, L. De Raedt, Miningzinc: A modeling language for constraint-based mining, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI'13, 2013, pp. 1365–1372.

[31] T. Guns, S. Nijssen, L. De Raedt, Itemset mining: A constraint programming perspective, Artificial Intelligence 175 (2011) 1951–1983.

[32] J. Wang, J. Han, Y. Lu, P. Tzvetkov, TFP: An efficient algorithm for mining top-k frequent closed itemsets., IEEE Transactions on Knowledge Data Engineering 17 (2005) 652–664.

[33] R. Agrawal, R. Srikant, Mining sequential patterns, in: A. L. P. C. Philip S. Yu (Ed.), Proceedings of the Eleventh International Conference on Data Engineering (ICDE'1995), IEEE Computer Society, 1995, pp. 3–14.

[34] S. Jabbour, L. Sais, Y. Salhi, The top-k frequent closed itemset mining using top-k sat problem, in: ECML/PKDD (3), 2013, pp. 403–418.

[35] G. Tseitin, On the complexity of derivations in the propositional calculus, in: Structures in Constructives Mathematics and Mathematical Logic, Part II, 1968, pp. 115–125.

[36] Z. Fu, S. Malik, On Solving the Partial MAX-SAT Problem, in: Proceedings of the Ninth International Conference on Theory and Applications of Satisfiability Testing (SAT'06), 2006, pp. 252–265.

[37] J. P. Warners, A linear-time transformation of linear inequalities into conjunctive normal form, Information Processing Letters (1996).

[38] C. Sinz, Towards an optimal cnf encoding of boolean cardinality constraints, in: 11th International Conference on Principles and Practice of Constraint Programming - CP 2005, 2005, pp. 827–831.

[39] N. Eén, N. Sörensson, Translating pseudo-boolean constraints into SAT, JSAT 2 (2006) 1–26.

[40] R. Asin, R. Nieuwenhuis, A. Oliveras, E. Rodriguez-Carbonell, Cardinality networks: a theoretical and empirical study, Constraints 16 (2011) 195–221.

[41] S. Jabbour, L. Saïs, Y. Salhi, A pigeon-hole based encoding of cardinality constraints, in: International Symposium on Artificial Intelligence and Mathematics, ISAIM 2014, Fort Lauderdale, FL, USA, January 6-8, 2014, 2014.

[42] M. Cadoli, On the complexity of model finding for nonmonotonic propositional logics, in: 4th Italian conference on theoretical computer science, 1992, pp. 125–139.

[43] A. R. Morgado, J. a. P. Marques-Silva, Good Learning and Implicit Model Enumeration, in: International Conference on Tools with Artificial Intelligence (ICTAI'2005), IEEE, 2005, pp. 131–136.

[44] E. Egho, C. Raïssi, T. Calders, N. Jay, A. Napoli, On measuring similarity for sequences of itemsets, Data Mining and Knowledge Discovery 29 (2015) 732–764.

[45] Y. Wu, L. Wang, J. Ren, W. Ding, X. Wu, Mining sequential patterns with periodic wildcard gaps, Applied Intelligence 41 (2014) 99–116.

[46] S. Jabbour, L. Sais, Y. Salhi, On SAT models enumeration in itemset mining, CoRR abs/1506.02561 (2015).

[47] T. Lane, Filtering techniques for rapid user classification, in: AAAI-98/ICML-98 Joint Workshop on AI Approaches to Time-series Analysis, 1998, pp. 58–63.

[48] T. Uno, M. Kiyomi, H. Arimura, LCM ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets, in: FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Brighton, UK, November 1, 2004, 2004.

[49] R. Williams, C. P. Gomes, B. Selman, Backdoors to typical case complexity., in: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, 2003, pp. 1173–1178.

A SAT-Based Approach for Mining Association Rules

A. Boudane, S. Jabbour, L. Sais, Y. Salhi. A SAT-Based Approach for Mining Association Rules. 25th International Joint Conference on Artificial Intelligence, IJCAI 2016 : IJCAI/AAAI Press, 2472-2478.

Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)

A SAT-Based Approach for Mining Association Rules

Abdelhamid Boudane, Said Jabbour, Lakhdar Sais, Yakoub Salhi
CRIL - CNRS, Université d'Artois
Rue Jean Souvraz, SP-18 62307, Lens Cedex 3
{boudane, jabbour, sais, salhi}@cril.fr

Abstract

Discovering association rules from transaction databases is one of the most studied data mining task. Many effective techniques have been proposed over the years. All these algorithms share the same two steps methodology: frequent itemsets enumeration followed by effective association rules generation step. In this paper, we propose a new propositional satisfiability based approach to mine association rules in a single step. The task is modeled as a Boolean formula whose models correspond to the rules to be mined. To highlight the flexibility of our proposed framework, we also address two other variants, namely the closed and indirect association rules mining tasks. Experiments on many datasets show that on both closed and indirect association rules mining tasks, our declarative approach achieves better performance than the state-of-the-art specialized techniques.

1 Introduction

Association analysis is one of the fundamental data mining task. It aims to discover interesting relationships hidden in large data sets. Such relationships between sets of items are presented in the form of implications, called association rules, along with metrics to quantify the rule's relevance. Since the first well known application [Agrawal and Srikant, 1994], usually referred to as market basket data analysis, several new application domains have been identified, including among others, bioinformatics, medical diagnosis, networks intrusion detection, web mining, and scientific data analysis. This broad spectrum of applications enabled association analysis to be applied to a variety of data sets, including sequential, spatial, and graph-based data.

There has been considerable work developing a nice theory and fast algorithms for mining association rules. Among the existing techniques, Apriori [Agrawal and Srikant, 1994] and FP-Growth [Han *et al.*, 2004] are some of the most known algorithms. All these algorithms share the same two steps methodology. The first step is to find all itemsets with adequate supports and the second step is to generate association rules with high confidence by combining these frequent or large itemsets. Support and confidence are two important statistical measures. For a given rule $r : X \rightarrow Y$, the support is defined as the percentage of transactions containing $X \cup Y$, while the confidence provides an estimate of the conditional probability $p(X/Y)$ of Y given X usually defined as the ratio between the supports of $X \cup Y$ and X. Association rules mining aims to identify all rules meeting user specified constraints such as minimum support and minimum confidence. Supports is used to eliminate uninteresting rules, while confidence measures the reliability of the inference made by the rule. The higher the confidence, the more likely it is for Y to be present in transactions that contain X.

From this brief overview, two observations can be made. First, one can easily guess why association rules mining techniques follow a two steps based approach. Secondly, we can also observe that the relevance of the rules to be mined are expressed using constraints. As pointed out in [Raedt *et al.*, 2011], on many data mining tasks, constraints are often part of the problem specification. This observation led to a new active and multidisciplinary research field, initiated by Luc De Reardt et al. [Raedt *et al.*, 2008], focussing on cross fertilization between data mining and artificial intelligence (AI). Two well-known AI representation and solving models, namely constraint programming (CP) and propositional satisfiability (SAT) have been used to model and solve several data mining tasks, including pattern mining [Guns *et al.*, 2011; Négrevergne and Guns, 2015; Jabbour *et al.*, 2015a; 2015b] and clustering [Davidson *et al.*, 2010; Métivier *et al.*, 2012; Dao *et al.*, 2013]. This new framework offers a declarative and flexible representation model. Indeed, in data mining, new constraints often require new implementations, while they can be easily integrated in such declarative models.

Following this research trend, in this paper, we propose a new propositional satisfiability based approach to mine association rules in a single step. The task is modeled as a propositional formula whose models correspond to the rules to be mined. As the number of association rules can grow rapidly, especially as we lower the frequency requirements, limiting the number of rules produced without information loss has be recognized as an important issue. It has also been noted that some of the infrequent patterns, such as indirect associations, provide useful insight into the data. In our second contribution, we consider two well-known variants designed to overcome these two main limitations, namely closed [Taouil

et al., 2000] and indirect [Tan *et al.*, 2000] association rules mining tasks. Our goal is to highlight the flexibility and the nice declarative features of our proposed framework.

The paper is organized as follows. After some preliminaries about propositional satisfiability and association rules mining, we present in Section 3 our SAT-based encoding of the association rules mining task. Section 4 presents the two variants mentioned above, namely closed and indirect associations rules. In Section 5, our proposed approches are extensively evaluated on many data sets, demonstrating that declarative approaches can achieve better performances with respect to specialized techniques particularly on closed and indirect association rules.

2 Preliminaries

2.1 Propositional Logic and SAT Problem

In this section, we define the syntax and the semantics of propositional logic. Let Prop be a countable set of propositional variables. We use the letters p, q, r, etc to range over Prop. The set of *propositional formulas*, is defined inductively started from Prop, the constant $\bot$ denoting $false$, the constant $\top$ denoting $true$, and using the usual logical connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$. We use $\mathcal{P}(A)$ to denote the set of propositional variables appearing in the formula A. A *Boolean interpretation* $\mathcal{I}$ of a formula A is defined as a function from $\mathcal{P}(A)$ to $\{0,1\}$ (0 corresponds to $false$ and 1 to $true$). A *model* of a formula A is a Boolean interpretation $\mathcal{I}$ that satisfies A, i.e. $\mathcal{I}(A) = 1$. A formula A is satisfiable if there exists a model of A. A formula in *conjunctive normal form* (CNF) is a conjunction ($\wedge$) of clauses, where a *clause* is a disjunction ($\vee$) of literals. A *literal* is a propositional variable (p) or a negated propositional variable ($\neg p$). The two literals p and $\neg p$ are called *complementary*. Let us mention that any propositional formula can be translated to a CNF formula equivalent w.r.t. satisfiability, using linear Tseitin's encoding [Tseitin, 1968]. The *SAT problem* consists in deciding wether a given CNF formula admits a model or not.

2.2 Association Rules

Let Ω be a finite non empty set of symbols, called *items*. From now on, we assume that this set is fixed. We use the letters a, b, c, etc to range over the elements of Ω. An *itemset* I over Ω is defined as a subset of Ω, i.e., $I \subseteq \Omega$. We use 2^Ω to denote the set of itemsets over Ω and we use the capital letters I, J, K, etc to range over the elements of 2^Ω.

A *transaction* is an ordered pair (i, I) where i is a natural number, called *transaction identifier*, and I an itemset, i.e., $(i, I) \in \mathbb{N} \times 2^\Omega$. A *transaction database* $\mathcal{D}$ is defined as a finite non empty set of transactions ($\mathcal{D} \subseteq \mathbb{N} \times 2^\Omega$) where each transaction identifier refers to a unique itemset.

Given a transaction database $\mathcal{D}$ and an itemset I, the *cover* of I in $\mathcal{D}$, denoted $\mathcal{C}(I, \mathcal{D})$, is defined as follows: $\{i \in \mathbb{N} \mid (i, J) \in \mathcal{D} \; and \; I \subseteq J\}$. The *support* of I in $\mathcal{D}$, denoted $\mathcal{S}(I, \mathcal{D})$, corresponds to the cardinality of $\mathcal{C}(I, \mathcal{D})$, i.e., $\mathcal{S}(I, \mathcal{D}) = |\mathcal{C}(I, \mathcal{D})|$. An itemset $I \subseteq \Omega$ such that $\mathcal{S}(I, \mathcal{D}) \geq 1$ is a *closed itemset* if, for all itemsets J with $I \subset J$, $\mathcal{S}(J, \mathcal{D}) < \mathcal{S}(I, \mathcal{D})$.

For instance, consider the transaction database $\mathcal{D}$ in Table 1. In this case, we have $\mathcal{C}(\{a, b\}, \mathcal{D}) = \{1, 2, 3\}$ and $\mathcal{S}(\{a, b\}, \mathcal{D}) = 3$ while $\mathcal{S}(\{f\}, \mathcal{D}) = 2$. The itemset $\{a, b\}$ is closed, while $\{f\}$ is not closed.

Tid	Itemset
1	a b c d
2	a b e f
3	a b c
4	a c d f
5	g
6	d
7	d g

Table 1: A Transaction Database $\mathcal{D}$.

In this work, we are interested in the problem of mining association rules. An *association rule* is a pattern of the form $X \rightarrow Y$ where X (called the antecedent) and Y (called the consequent) are two disjoint itemsets. In association rules mining, the interestingness predicate is defined using the notions of support and confidence. The *support of an association rule* $X \rightarrow Y$ in a transaction database $\mathcal{D}$, defined as $\mathcal{S}(X \rightarrow Y, \mathcal{D}) = \frac{\mathcal{S}(X \cup Y, \mathcal{D})}{|\mathcal{D}|}$, determines how often a rule is applicable to a given data set, i.e., the occurrence frequency of the rule. The *confidence* of $X \rightarrow Y$ in $\mathcal{D}$, defined as $\mathcal{C}onf(X \rightarrow Y, \mathcal{D}) = \frac{\mathcal{S}(X \cup Y, \mathcal{D})}{\mathcal{S}(X, \mathcal{D})}$, provides an estimate of the conditional probability of Y given X. A *valid association rule* is an association rule with support and confidence greater or equal to the minimum support (α) and minimum confidence (β) thresholds. More precisely, given a transaction database $\mathcal{D}$, a minimum support threshold α and a minimum confidence threshold β, the problem of mining association rules consists in computing the following set:
$\mathcal{MAR}(\mathcal{D}, \alpha) = \{X \rightarrow Y \mid X, Y \subseteq \Omega \wedge \mathcal{S}(X \rightarrow Y, \mathcal{D}) \geq \alpha \wedge \mathcal{C}onf(X \rightarrow Y, \mathcal{D}) \geq \beta\}$

From Table 1, we get $\mathcal{S}(\{a\} \rightarrow \{b\}, \mathcal{D}) = 3/7$ and $\mathcal{C}onf(\{a\} \rightarrow \{b\}, \mathcal{D}) = 3/4$.

3 SAT-Based Association Rules Mining

In this section, we describe a SAT encoding for the problem of mining association rules. The basic idea consists in the use of propositional variables to represent the covers of the itemsets X and $X \cup Y$ for each candidate rule $X \rightarrow Y$. These variables are used in 0/1 linear inequalities to determine wether the support and the confidence of the candidate rule are greater than the specified minimum thresholds for the support and the confidence.

Let $\mathcal{D} = \{(1, I_1), \ldots, (m, I_m)\}$ be a transaction database, α a minimum support threshold and β a minimum confidence threshold. To represent the two itemsets of each candidate rule $X \rightarrow Y$, we associate two propositional variables x_a and y_a to each item a. The variables of the form x_a (resp. y_a) are used to represent the antecedent (resp. consequent) of each candidate rule. Then, to represent the cover of X and $X \cup Y$, we associate to each transaction identifier $i \in \{1 \ldots m\}$ two propositional variables p_i and q_i. The variables of the form p_i (resp. q_i) are used to

represent the cover of X (resp. $X \cup Y$). More precisely, given a Boolean interpretation $\mathcal{I}$, the candidate rule is $X = \{a \in \Omega \mid \mathcal{I}(x_a) = 1\} \rightarrow Y = \{b \in \Omega \mid \mathcal{I}(y_b) = 1\}$, the cover of X is $\{i \in \mathbb{N} \mid \mathcal{I}(p_i) = 1\}$, and the cover of $X \cup Y$ is $\{i \in \mathbb{N} \mid \mathcal{I}(q_i) = 1\}$.

We now describe our SAT-based encoding using the propositional variables described previously. The first propositional formula allows us to express the constraint $X \cap Y = \emptyset$:

$$\bigwedge_{a \in \Omega} (\neg x_a \vee \neg y_a) \tag{1}$$

To obtain the cover of the itemset X, we use the following propositional formula:

$$\bigwedge_{i=1}^{m} (\neg p_i \leftrightarrow \bigvee_{a \in \Omega \setminus I_i} x_a) \tag{2}$$

In this formula, p_i is $false$ if and only if X contains an item that does not belong to the transaction i. As a consequence, the cover of X is $\{i \in \mathbb{N} \mid \mathcal{I}(p_i) = 1\}$.

In the same way as the previous formula, we use the following formula to capture the cover of $X \cup Y$:

$$\bigwedge_{i=1}^{m} (\neg q_i \leftrightarrow \neg p_i \vee (\bigvee_{a \in \Omega \setminus I_i} y_a)) \tag{3}$$

It is worth noticing that we use the propositional variables p_i to prevent the reuse of the variables x_a. This allows us to obtain a more compact formula.

Let us now introduce the formula expressing that the support of the candidate rule has to be greater than or equal to the specified threshold $m \times \alpha$ (in percentage):

$$\sum_{i=1}^{m} q_i \geq m \times \alpha \tag{4}$$

This formula means that the number of variables q_i ($i \in \{1 \dots m\}$) assigned to 1 has to be greater than or equal to α, which is equivalent to the fact that the support of $X \cup Y$ is greater than or equal to $m \times \alpha$.

Finally, we describe the formula expressing the fact that β is a minimum confidence threshold:

$$\frac{\sum_{i=1}^{m} q_i}{\sum_{i=1}^{m} p_i} \geq \beta$$

We here consider that β is given in percentage format $\beta\%$ where β is a positive integer. Thus, the previous formula can be rewritten as follows:

$$100 * \sum_{i=1}^{m} q_i - \beta * \sum_{i=1}^{m} p_i \geq 0 \tag{5}$$

The two propositional formulas (4), (5) corresponds to 0/1 linear inequalities, usually called cardinality (resp. Pseudo Boolean) constraints. The first linear encoding of general 0/1 linear inequalities to CNF has been proposed by J. P. Warners in [Warners, 1998]. Several authors have addressed the issue of finding an efficient encoding of cardinality (e.g. [Asin *et al.*, 2011; Jabbour *et al.*, 2013a; Warners, 1998]) or Pseudo Boolean (e.g. [Abío *et al.*, 2012; Eén and Sörensson, 2006; Warners, 1998]) constraints as a CNF formula. Efficiency refers to both the compactness of the representation (size of the CNF formula) and to the ability to achieve the same level of constraint propagation (generalized arc consistency) on the CNF formula.

We use $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ to denote the encoding corresponding to the conjunction of formulas (1), (2), (3), (4), and (5).

4 Closed and Indirect Association Rules

In this section, we highlight the nice declarative and flexible aspects of the proposed SAT framework for mining association rules. To this end, we consider two well-known association rules variants, namely closed [Taouil *et al.*, 2000] and indirect association rules [Tan *et al.*, 2000]. The first has been proposed to avoid redundant rules using condensed representation, while the second aims to find indirect relations in data. Indirect association, extensively used to build web recommandation systems [Kazienko, 2009], refers to a pair of items that rarely occur together but highly depend on the presence of a mediator itemset.

Definition 1 (Closed Association Rule) *An association rule $r : X \rightarrow Y$ is a Closed association rule iff $r : X \rightarrow Y$ is a valid association rule and $X \cup Y$ is closed.*

Intuitively, we obtain a closed association rule by maximizing either its antecedent or its consequent while decreasing neither the support nor the confidence.

A SAT encoding of the problem of mining closed association rules, noted $\mathcal{E}_{CAR}(\mathcal{D}, \alpha, \beta)$, can be simply obtained by extending the encoding described previously ($\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$) with the following formula:

$$\bigwedge_{a \in \Omega} (\bigwedge_{i=1}^{m} (q_i \rightarrow a \in I_i) \wedge \neg x_a \rightarrow y_a) \tag{6}$$

This formula means that if we have $\mathcal{C}(X \cup Y, \mathcal{D}) = \mathcal{C}(X \cup Y \cup \{a\}, \mathcal{D})$ then the item a has to belong to Y, i.e., $a \in Y$. As a consequence, if $X \rightarrow Y$ and $X \rightarrow Y \uplus \{a\}$ are two valid association rules ($\uplus$ stands for disjoint union), then the Boolean interpretation that corresponds to the rule $X \rightarrow Y$ is a counter-model of (6). Moreover, if there is no item a such that $X \rightarrow Y \uplus \{a\}$ is a valid association rule, then the Boolean interpretation corresponding to $X \rightarrow Y$ is a model of (6). Furthermore, it is worth noticing that the formula (6) encodes that $X \cup Y$ is closed. Indeed, we do not need to add a formula to maximize the antecedent of the rule as it is implicitly encoded in the formula (6). More precisely, the formula remains the same if we substitute $\neg x_a$ (resp. y_a) by $\neg y_a$ (resp. x_a). Thus, the formula (6) describes a necessary and sufficient requirement for mining the closed association rules. We can also note that the number of clauses added by the formula (6) is equal to the number of items.

We now consider a second mining task related to the problem of mining association rules. It consists in mining *indirect rules*, which allow to discover items that rarely occur together but frequently occur with other items [Tan *et al.*, 2000].

Definition 2 *Let $\mathcal{D}$ be a transaction database. Two items a_0 and b_0 are indirectly associated via an itemset M, called mediator, w.r.t. a maximum support threshold λ, a minimum support threshold α and a mediator dependence threshold β iff the following conditions hold:*

- $\mathcal{S}(\{a_0\} \rightarrow \{b_0\}, \mathcal{D}) \leq \lambda$ *(Itempair Support Condition)*[1].
- *There exists a non empty itemset M such that:*
  1. $\mathcal{S}(\{a_0\} \rightarrow M, \mathcal{D}) \geq \alpha$ *and* $\mathcal{S}(\{b_0\} \rightarrow M, \mathcal{D}) \geq \alpha$ *(Mediator Support Condition)*
  2. $\mathcal{D}ep(\{a_0\}, M, \mathcal{D}) \geq \beta$ *and* $\mathcal{D}ep(\{b_0\}, M, \mathcal{D}) \geq \beta$ *where $\mathcal{D}ep(p, Q, \mathcal{D})$ is a dependence measure between p and Q w.r.t. $\mathcal{D}$ (Dependence Condition)*

In other words, in the problem of mining indirect rules, we look for pairs of items that are infrequent (or rare) but separately involved in interesting association rules with the same consequent.

Over the year, several measures of dependence between two itemsets X and Y have been proposed, including IS measure [Tan *et al.*, 2002] and classical confidence. The relevance of such measures depends on the target application. In this paper, as a dependency measure, we simply use the confidence of $X \rightarrow Y$. This last dependence measure is for example employed in [Kazienko, 2005] for a web recommendation application. Using confidence as a dependency measure and minimum confidence threshold instead of mediator dependance threshold the two conditions (mediator support and dependence) in Definition 2 can be simply stated as: $\{a_0\} \rightarrow M$ and $\{b_0\} \rightarrow M$ are two valid association rules w.r.t. the minimum support α and minimum confidence β thresholds.

In order to define a SAT encoding for the problem of mining indirect association rules, we have to use propositional variables that allow us to capture the cover of the association rule $\{a_0\} \rightarrow \{b_0\}$ and we adapt the encoding $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ to constrain the antecedent of the two association rules $\{a_0\} \rightarrow M$ and $\{b_0\} \rightarrow M$ to contain only a single item. We use the propositional variables $x_c^{a_0}$ and $x_c^{b_0}$ for each item c to represent a_0 and b_0 respectively. Similarly, we use the same set of variables y_a for each item a as in $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ to capture the elements of the mediator. Moreover, we introduce variables of the form $p_i^{a_0}$ (resp. $p_i^{b_0}$) to express the cover of the item a_0 (resp. b_0) in the same way as in $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ and variables of the form $q_i^{a_0}$ and $q_i^{b_0}$ to express the covers of $M \cup \{a_0\}$ and $M \cup \{b_0\}$ respectively. Finally, we also introduce variables of the form r_i to capture the cover of $\{a_0, b_0\}$.

The following formula allows us to capture the association rule $\{a_0\} \rightarrow M$:

$$\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta) \wedge (\sum_{a \in \Omega} x_a^{a_0} = 1) \tag{7}$$

where the variables of the form x_a, p_i and q_i are replaced in $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ with $x_a^{a_0}$, $p_i^{a_0}$ and $q_i^{a_0}$ respectively. The cardinality constraint $\sum_{a \in \Omega} x_a^{a_0} = 1$ is used to require that antecedent of the rules contains a single item.

[1] We can equivalently write $\frac{\mathcal{S}(\{a_0, b_0\}, \mathcal{D})}{|\mathcal{D}|} \leq \lambda$

In the same way as (7), we use the following formula to capture the association rule $\{b_0\} \rightarrow M$:

$$\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta) \wedge (\sum_{a \in \Omega} x_a^{b_0} = 1) \tag{8}$$

where the variables of the form x_a, p_i and q_i are replaced in $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ with $x_a^{b_0}$, $p_i^{b_0}$ and $q_i^{b_0}$ respectively.

We now describe the formula that allows us to capture the cover of $\{a_0, b_0\}$:

$$r_i \leftrightarrow (p_i^{a_0} \wedge p_i^{b_0}) \tag{9}$$

Finally, we introduce the formula expressing that $\{a_0\} \rightarrow \{b_0\}$ is infrequent w.r.t. the maximum support threshold λ:

$$\sum_{i=1}^{m} r_i \leq m \times \lambda \tag{10}$$

In Definition 2, the items a_0 and b_0 are interchangeable leading to symmetrical indirect association rules. To avoid enumerating such redundant indirect association rules, we break symmetries between a_0 and b_0 by adding the constraints $a_0 < b_0$ over the set of items Ω expressed as:

$$\bigwedge_{a, a' \in \Omega, a' \leq a} \neg x_a^{a_0} \vee \neg x_{a'}^{b_0} \tag{11}$$

We use $\mathcal{E}_{IR}(\mathcal{D}, \lambda, \alpha, \beta)$ to denote the encoding of the problem of mining indirect rules $(7) \wedge (8) \wedge (9) \wedge (10) \wedge (11)$.

5 Experiments

In this section, we present a comparative experimental evaluation of our proposed approaches with specialized association rules mining algorithms. We consider, three mining tasks, namely classical (pure), closed, and indirect association rules.

For our SAT based association rules mining, to enumerate all the models of a given propositional CNF formula, we use an adaptation of modern SAT solvers proposed in [Jabbour *et al.*, 2014]. For cardinality and pseudo Boolean constraints, similarly to constraint programming, a propagator is associated to each constraint, obtained by maintaining the sum of its assigned variables. Managing such constraints on the fly outperforms our previous implementation based on the state-of-the-art SAT encodings [Jabbour *et al.*, 2013b].

Another advantage, is that for each association rules mining instance, as the constraints (1), (2) and (3) does not depend on the specified thresholds, the propositional formula is generated only once. On all the considered data, the encoding phase does not exceed 5 seconds CPU time.

Let us note that our approach can be easily encoded using MiningZinc, a general framework for constraint-based pattern mining [Guns *et al.*, 2013]. MiningZinc is a nice declarative framework, it offers a high level modeling language with a toolchain component for finding solutions.

In the experiments, we indicates by `SFAR_R` with $R \in \{pure, closed, indirect\}$, our SAT based approach for mining the corresponding (R) association rules. We compare our approaches to two specialized association rules mining

	SFAR_Pure		ZART_Pure		SFAR_Closed		ZART_Closed		SFAR_Indirect		SPMF_Indirect	
data (#items, #trans, density)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)
Audiology (148, 216, 45%)	20	855.00	20	855.01	20	855.00	20	855.01	124	453.74	61	680.45
Zoo-1 (36, 101, 44%)	400	19.12	400	6.37	400	0.52	400	11.28	250	0.15	250	9.12
Tic-tac-toe (27, 958, 33%)	400	0.09	400	0.24	400	0.09	400	0.23	250	0.09	250	0.20
Anneal (93, 812, 45%)	101	709.50	101	678.41	147	604.09	103	679.31	171	309.69	55	702.04
Australian-credit (125, 653, 41%)	245	370.17	264	321.62	268	323.29	226	403.72	232	121.06	156	339.56
German-credit (112, 1000, 34%)	306	246.88	322	192.52	329	198.02	304	238.79	244	49.07	210	154.49
Heart-cleveland (95, 296, 47%)	284	286.38	301	252.27	304	251.05	262	340.15	235	64.97	203	300.48
Hepatitis (68, 137, 50)	305	241.41	304	228.00	324	206.02	266	312.26	245	32.98	205	187.92
Hypothyroid (88, 3247, 49%)	85	732.12	121	665.41	107	686.95	64	761.59	163	336.40	81	621.29
Kr-vs-kp (73, 3196, 49%)	172	552.92	203	487.73	192	523.66	146	590.89	204	206.47	114	499.33
Lymph (68, 148, 40%)	336	181.64	338	170.37	387	63.22	291	281.35	250	6.10	211	170.19
Mushroom (119, 8124, 18%)	366	109.12	387	46.00	400	30.32	390	42.84	250	8.89	250	29.62
Primary-tumor (31, 336, 48%)	400	3.68	400	1.17	400	2.03	400	18.82	250	0.15	250	2.63
Soybean (50, 650, 32%)	400	2.90	400	1.50	400	0.17	400	7.94	250	0.05	250	0.76
Splice-1 (287, 3190, 21%)	380	53.44	400	3.52	380	54.04	400	3.25	250	61.73	250	0.50
Vote (48, 435, 33%)	380	66.74	400	1.46	400	32.40	398	30.22	250	0.84	250	1.48
Total	4560	279.76	**4741**	**247.29**	**4838**	**242.24**	4470	286.10	**3618**	**103.27**	3046	231.25

Table 2: Pure, Closed, and Indirects Associations Rules: SFAR vs ZART and SFAR vs SPMF

algorithms *Coron* [2] and *SPMF* [3][Fournier-Viger *et al.*, 2014]. *Coron* and *SPMF* are two multi-purpose data mining toolkits, implemented in Java, which incorporate a rich collection of data mining algorithms. For *pure* and *closed* association rules, we compare our approach to the *ZART* algorithm implemented in the *Coron* toolkit, which is one of the recent state-of-the-art algorithms for enumerating closed association rules [Szathmary *et al.*, 2007]. For *indirect* association rules, we compare our solver to the *SPMF* implementation.

To give an idea on the size of our encodings, for classical association rule mining, the smallest (resp. the biggest) formula corresponds to the encoding of zoo-1 (resp. mushroom) data and contains 274 variables and 4379 clauses (resp. 16486 variables and 1616795 clauses).

To compare the performances of the different mining approaches, for each data we proceed as follows:

- For pure and closed association rules, the support is varied from 5% to 100% with an interval of size 5%. The confidence is varied in the same way. Then, for each data, a set of 400 configurations is generated.
- For indirect association rules, there are an additional parameter λ. The frequency and confidence are varied from 20% to 100% with an interval of size 20%. λ is varied from 10% to 100% with an interval of size 10%. This leads to 250 configurations for each data.

All the experiments were done on Intel Xeon quad-core machines with 32GB of RAM running at 2.66 Ghz. For each instance, we fix the timeout to 15 minutes of CPU time.

Table 2 describes our comparative results. We report in column 1 the name of the data and its characteristics in parenthesis: number of items (#items), number of transactions (#trans) and density. For each algorithm, we report the number of solved configurations ($\#S$), and the average solving time ($avg.time$ in seconds). For each unsolved configuration, the time is set to 900 seconds (time out). In the last row of Table 2, we provide the total number of solved configurations and the global average CPU time in seconds.

[2]Coron: http://coron.loria.fr/site/system.php

[3]SPMF: http://www.philippe-fournier-viger.com/spmf/

Pure rules: The performances of ZART algorithm are better than SFAR. It solves 181 configurations more and it is better on all the considered data. ZART performs the enumeration of pure association rules in two steps. We observed that ZART performs the first step efficiently. Its CPU time does not exceed few seconds on the majority of the considered configurations. For the pure rules, the second step remains easy enough to perform. On classical association rules, the specialized algorithm ZART is better than SFAR.

Closed association rules: On this category, our SAT based approach outperform ZART. It solves 368 configurations more than ZART. Except for Splice-1 data, SFAR is the best on all the data in terms of the number of solved configurations and average CPU time. Let us remark that for Splice-1 data, the number of closed association rules is very limited (less than 4000). This explains why SFAR is worse than ZART on this data.

Let us recall that ZART finds the closed association rules in two steps. In the first step, the set of all frequent closed itemsets are efficiently enumerated (in few seconds), while in the second step, the extraction of association rules from the closed itemsets already generated is more time consuming. For instance, on Lymph data, SFAR is remarkably efficient. It solves about 100 configurations more than ZART. More generally, the higher the density of the data, the better are the performances of SFAR.

Indirect association rules: The performances of SFAR are very impressive. SFAR approach solves 572 instances more than SPMF. Here again, the approach is better on all the considered data. As we can remark the time needed by SFAR to obtain all indirect association rules is relatively stable and very low compared to SPMF. The number of indirect associations is very small compared to classical or closed association rules. However, SPMF takes a lot of time to find them. For example, if we take the Hepatitis data with $frequency = 40\%$, $confidence = 40\%$ and $\lambda = 20\%$, SPMF takes 122.56 seconds to find just 359 indirect association rules, while SFAR does not exceed 1 second. We also

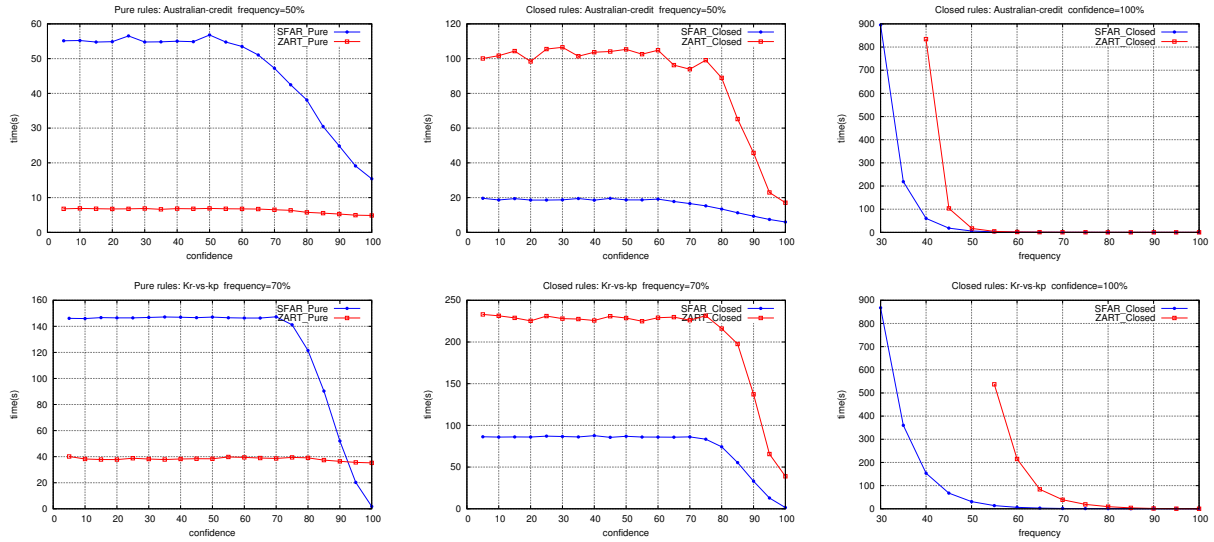


Figure 1: Highlights: `Australian-credit` and `Kr-vs-kp`

frequency (%)	40	45	50	55	60	65	70
Kr-vs-kp	7.67	5.68	3.64	2.99	2.46	1.95	1.67
Australian-credit	12.38	8.13	5.61	4.29	3.23	2.62	2.01

Table 3: Pure vs Closed: $\#Purerules/\#Closed\ rules$

noticed that for some configurations, SPMF takes excessive CPU time without finding any indirect association rule under the time limit. As a summary on indirect association rules, SFAR outperforms SPMF.

In Figure 1, the behavior of the considered approaches are highlighted on two representative data, $\texttt{Australian} - \texttt{credit}$ and $\texttt{Kr} - \texttt{vs} - \texttt{kp}$. We varied one parameter, while maintaining the others fixed. For pure association rules, ZART and SFAR present similar behavior. When the frequency decreases, the time needed to find all rules increases. Let us remark that for some particular parameters values, our approach can outperforms the one of ZART on pure rules as is the case for Kr-vs-kp. Similar behavior is also observed for frequent closed association rules. However, we can note that when the confidence goes from 100% to 80%, the CPU time dramatically increases. Such a gap is more visible with SFAR on classical association rules and with ZART on closed association rules. Indeed, for Kr-vs-kp instance, using SFAR, we vary between 0 to 70 seconds while with ZART approach, the variation range is from 40 to 240 seconds.

Throughout this experimental study, we noticed that the specialized algorithms like ZART performs the first frequent (resp. closed) itemsets enumeration step efficiently. However, they take excessive CPU time in the second rules extraction step. Additionally, the extraction step is more time consuming for closed rules than classical rules even if the number of closed rules is lower in general. In Table 3, we provide the variation of the ratio between the number of classical (pure) rules and the number of closed rules. As we can observe, as the frequency decreases, the number of classical rules increases rapidly compared to the number of the closed rules. This last observation explains why SFAR is more efficient on the enumeration of closed rules than on the enumeration of classical ones. Overall, ZART solves more configurations for pure rules than for closed ones, while SFAR is more efficient in mining closed rules and indirect rules than classical ones.

As a summary of our experiments, we can say that for mining tasks combining several constraints, our declarative and flexible approach is better than specialized mining tools.

6 Acknowledgments

This work has been supported by the CNRS project QuDoSSI - Défi Mastodons 2016.

7 Conclusion and perspectives

In this paper we developed a novel association rules mining approach that accurately discovers association rules efficiently. Our declarative approach contrasts with all the previous techniques as the mining of association rules is performed in a single step, thanks to our SAT based encoding. As a second contribution, we have shown that our proposed framework is flexible and declarative, as one can easily model other important variants, such as closed and indirect association rules mining. The experiments particularly show that on closed and indirect association rules mining, our proposed approaches achieves better performance with respect to specialized mining techniques.

Our work opens several perspectives. First, our results on closed and indirect association rules provide new research directions for association rules mining. Indeed, several other variants can be addressed more efficiently by extending our proposed framework. Such rules include among others top-k association rules, weighted association rules [Tao *et al.*, 2003] and disjunctive association rules [Nanavati *et al.*, 2001].

References

[Abío *et al.*, 2012] Ignasi Abío, Robert Nieuwenhuis, Albert Oliveras, Enric Rodríguez-Carbonell, and Valentin Mayer-Eichberger. A new look at bdds for pseudo-boolean constraints. *J. Artif. Intell. Res. (JAIR)*, 45:443–480, 2012.

[Agrawal and Srikant, 1994] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of VLDB'94*, pages 487–499, 1994.

[Asin *et al.*, 2011] Roberto Asin, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodriguez-Carbonell. Cardinality networks: a theoretical and empirical study. *Constraints*, 16(2):195–221, 2011.

[Dao *et al.*, 2013] Thi-Bich-Hanh Dao, Khanh-Chuong Duong, and Christel Vrain. A declarative framework for constrained clustering. In *Proceedings of ECML PKDD'13*, pages 419–434, 2013.

[Davidson *et al.*, 2010] Ian Davidson, S. S. Ravi, and Leonid Shamis. A SAT-based framework for efficient constrained clustering. In *Proceedings of SDM'10*, pages 94–105, 2010.

[Eén and Sörensson, 2006] Niklas Eén and Niklas Sörensson. Translating pseudo-boolean constraints into SAT. *JSAT*, 2(1-4):1–26, 2006.

[Fournier-Viger *et al.*, 2014] Philippe Fournier-Viger, Antonio Gomaric, Ted Gueniche, Azadeh Soltani, Cheng-Wei Wu, and Vincent S Tseng. Spmf: a java open-source pattern mining library. *The Journal of Machine Learning Research*, 15(1):3389–3393, 2014.

[Guns *et al.*, 2011] Tias Guns, Siegfried Nijssen, and Luc De Raedt. Itemset mining: A constraint programming perspective. *Artif. Intell.*, 175(12-13):1951–1983, 2011.

[Guns *et al.*, 2013] Tias Guns, Anton Dries, Guido Tack, Siegfried Nijssen, and Luc De Raedt. Miningzinc: A modeling language for constraint-based mining. In *Proceedings of IJCAI'13*, pages 1365–1372, 2013.

[Han *et al.*, 2004] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004.

[Jabbour *et al.*, 2013a] Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi. A pigeon-hole based encoding of cardinality constraints. *TPLP*, 13, 2013.

[Jabbour *et al.*, 2013b] Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi. The top-k frequent closed itemset mining using top-k sat problem. In *Proceedings of ECML/PKDD'13*, pages 403–418, 2013.

[Jabbour *et al.*, 2014] Saïd Jabbour, Jerry Lonlac, Lakhdar Sais, and Yakoub Salhi. Extending modern SAT solvers for models enumeration. In *Proceedings of IEEE-IRI'14*, pages 803–810, 2014.

[Jabbour *et al.*, 2015a] Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi. Decomposition based SAT encodings for itemset mining problems. In *Proceedings of PAKDD'15*, pages 662–674, 2015.

[Jabbour *et al.*, 2015b] Said Jabbour, Lakhdar Sais, and Yakoub Salhi. Mining top-k motifs with a sat-based framework. *Artificial Intelligence*, pages –, 2015.

[Kazienko, 2005] Przemysław Kazienko. *Intelligent Information Processing and Web Mining: in Proceedings of IIPWM'05*, chapter IDARM — Mining of Indirect Association Rules, pages 77–86. Springer, 2005.

[Kazienko, 2009] Przemyslaw Kazienko. Mining indirect association rules for web recommendation. *Applied Mathematics and Computer Science*, 19:165–186, 2009.

[Métivier *et al.*, 2012] Jean-Philippe Métivier, Patrice Boizumault, Bruno Crémilleux, Mehdi Khiari, and Samir Loudni. Constrained clustering using SAT. In *Proceedings of IDA'12*, pages 207–218, 2012.

[Nanavati *et al.*, 2001] Amit Anil Nanavati, Krishna Prasad Chitrapura, Sachindra Joshi, and Raghu Krishnapuram. Mining generalised disjunctive association rules. In *Proceedings of CIKM'01*, pages 482–489, 2001.

[Négrevergne and Guns, 2015] Benjamin Négrevergne and Tias Guns. Constraint-based sequence mining using constraint programming. In *Proceedings of CPAIOR'15*, pages 288–305, 2015.

[Raedt *et al.*, 2008] Luc De Raedt, Tias Guns, and Siegfried Nijssen. Constraint programming for itemset mining. In *Proceedings of SIGKDD'08*, pages 204–212, 2008.

[Raedt *et al.*, 2011] Luc De Raedt, Siegfried Nijssen, Barry O'Sullivan, and Pascal Van Hentenryck. Constraint programming meets machine learning and data mining. *Dagstuhl Reports*, 1(5):61–83, 2011.

[Szathmary *et al.*, 2007] Laszlo Szathmary, Amedeo Napoli, and Sergei O. Kuznetsov. ZART: A multifunctional itemset mining algorithm. In *Proceedings of ICCLTA'07*, 2007.

[Tan *et al.*, 2000] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Indirect association: Mining higher order dependencies in data. In *Proceedings of PKDD'00*, pages 632–637, 2000.

[Tan *et al.*, 2002] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of SIGKDD'02*, pages 32–41, 2002.

[Tao *et al.*, 2003] Feng Tao, Fionn Murtagh, and Mohsen Farid. Weighted association rule mining using weighted support and significance framework. In *Proceedings of SIGKDD'03*, pages 661–666, 2003.

[Taouil *et al.*, 2000] Rafik Taouil, Nicolas Pasquier, Yves Bastide, and Lotfi Lakhal. Mining bases for association rules using closed sets. In *Proceedings of ICDE'00*, page 307, 2000.

[Tseitin, 1968] G.S. Tseitin. On the complexity of derivations in the propositional calculus. In *Structures in Constructives Mathematics and Mathematical Logic, Part II*, pages 115–125, 1968.

[Warners, 1998] Joost P. Warners. A linear-time transformation of linear inequalities into conjunctive normal form. *Inf. Process. Lett.*, 68(2):63–69, 1998.

Boolean satisfiability for sequence mining

S. Jabbour, L. Sais, Y. Salhi : Boolean satisfiability for sequence mining. ACM International Conference on Information and Knowledge Management, CIKM 2013, ACM, 649-658.

Boolean Satisfiability for Sequence Mining

Said Jabbour
jabbour@cril.fr

Lakhdar Sais
sais@cril.fr

Yakoub Salhi
salhi@cril.fr

CRIL - CNRS, University of Artois
F-62307 Lens Cedex
France

ABSTRACT

In this paper, we propose a SAT-based encoding for the problem of discovering frequent, closed and maximal patterns in a sequence of items and a sequence of itemsets. Our encoding can be seen as an improvement of the approach proposed in [8] for the sequences of items. In this case, we show experimentally on real world data that our encoding is significantly better. Then we introduce a new extension of the problem to enumerate patterns in a sequence of itemsets. Thanks to the flexibility and to the declarative aspects of our SAT-based approach, an encoding for the sequences of itemsets is obtained by a very slight modification of that for the sequences of items.

Categories and Subject Descriptors

F.4.1 [**Mathematical logic and formal languages**]: Mathematical Logic—*Logic and constraint programming*; H.2.8 [**Database management**]: Database applications—*Data mining*

Keywords

Data mining; Propositional satisfiability and modeling

1. INTRODUCTION

Frequent sequence data mining is the problem of discovering frequent patterns shared across time among an input data-sequence. Sequence mining is a central task in computational biology, temporal sequence analysis and text mining.

In this paper, we consider the pattern discovery problem for a specific class of patterns with wildcards in a sequence. The data-sequence can be seen as a sequence of items, while the pattern can be seen as a subsequence that might contains wildcards or jokers in the sense that they match any item [18, 20, 2]. At the first sight, allowing wildcards to occur in a pattern can be seen as an even more restrictive type of patterns in general. However as argued in [18] "*studying patterns with wildcards has the merit of capturing one important aspect of biological features that often concerns isolated positions inside a motif that are not part of the biological feature being captured*". The enumeration problem for maximal and closed motifs with wildcards has been investigated recently by several authors [19, 20, 2, 8]. One of the major problem is that the number of motifs can be of exponential size. This combinatorial explosion is tackled using different approaches. For example, in Parida et al. [18], the number of patterns is reduced by introducing the maximal non redundant q-patterns (patterns occurring at least q times in a sequence). Arimura and Uno [2] proposed a polynomial space and polynomial delay algorithm MaxMotif for maximal pattern discovery of the class of motifs with wildcards.

In this work, we follow the constraint programming (CP) based data mining framework proposed recently by Luc De Raedt et al. in [10] for itemset mining. This new framework offers a declarative and flexible representation model. New constraints often require new implementations in specialized approaches, while they can be easily integrated in such a CP framework. It allows data mining problems to benefit from several generic and efficient CP solving techniques. The authors show how some typical constraints (e.g. frequency, maximality, monotonicity) used in itemset mining can be formulated for use in CP [14]. This first study leads to the first CP approach for itemset mining displaying nice declarative opportunities without neglecting efficiency. More recently, Coquery et al. [8] have proposed a SAT-Based approach for Discovering for enumerating frequent, closed and maximal patterns with wildcards in a sequence of items. In this paper, we first propose a new SAT encoding of the problem of enumerating frequent, closed and maximal patterns with wildcards in a sequence of items. Our contribution can be seen as an improvement of the approach proposed in [8]. Indeed, the experimental results clearly show that the new encoding is significantly better than the original SAT encoding proposed in [8].

Encouraged by these promising results, we propose in our second contribution a new variant of the problem of discovering patterns with wildcards in a sequence, by considering a sequence of itemsets instead of a sequence of items. In this extension the emptyset will simply play the same role as the wildcard symbol. Indeed, one can use the emptyset to match any itemset. This new problem admits some similarities and differences with the classical sequential pattern

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

http://dx.doi.org/10.1145/2505515.2505577 .

mining problem introduced in [1]. Indeed, given an alphabet or a set of items Σ, in both problems we consider a sequence s as an ordered list of itemsets $s_0, \ldots, s_n$ where $s_i \subseteq \Sigma$ for $i = 0, \ldots, n$. However, the first difference resides in the definition of a subsequence. Indeed, in the sequential patterns, we say that s' is a subsequence of s if there exists a one-to-one order-preserving function f that maps (inclusion relation) itemsets in s' with itemsets in s. In our new setting, the notion of subsequence is defined w.r.t. to a given location and by using empty itemsets as wildcards. The other difference is that in the sequential pattern mining we consider a database of sequences of itemsets, while in our setting, we consider only a single sequence of itemsets. These differences leads also to different definitions of the notions of support, closeness and maximality. In summary, our new problem of discovering patterns in a sequence of itemsets can be seen as a simple and natural extension of the same problem in a sequence of items. As we can show later, the SAT encoding can be derived from the one used for the sequences of items with a very slight modification demonstrating its flexibility.

The paper is organized as follows. In the next section, we give a short overview of necessary definitions and notations about Boolean Satisfiability (SAT) and Frequent Pattern mining in a Sequence of items (FPS). The extension to the Frequent Pattern mining in a Sequence of Itemsets (FPSI) is presented in Section 3, followed by a discussion of some related works. Then our new SAT-based approach for FPS is described in Section 4, while the closeness and maximality constraints are discussed in Section 5. In Section 6, we show how the SAT encoding of FPSI, can be obtained by a slight modification of the SAT encoding of FPS. Finally, experimental results are conducted and discussed before concluding.

2. PRELIMINARIES

2.1 Boolean Satisfiability

In this section, we introduce the Boolean satisfiability problem, called SAT. It corresponds to the problem of deciding if a formula of propositional classical logic is consistent or not. It is one of the most studied NP-complete decision problem. In this work, we consider the associated problem of Boolean model enumeration.

We consider the conjunctive normal form (CNF) representation for the propositional formulas. A *CNF formula* Φ is a conjunction of clauses, where a *clause* is a disjunction of literals. A *literal* is a positive (p) or negated ($\neg p$) propositional variable. The two literals p and $\neg p$ are called *complementary*.

A CNF formula can also be seen as a set of clauses, and a clause as a set of literals. Let us recall that any propositional formula can be translated to CNF using linear Tseitin's encoding [22]. We denote by $Var(\Phi)$ the set of propositional variables occurring in Φ.

An *interpretation* $\mathcal{B}$ of a propositional formula Φ is a function which associates a value $\mathcal{B}(p) \in \{0, 1\}$ (0 corresponds to $false$ and 1 to $true$) to the variables $p \in Var(\Phi)$. A *model* of a formula Φ is an interpretation $\mathcal{B}$ that satisfies the formula. *SAT problem* consists in deciding if a given formula admits a model or not.

We denote by $\bar{l}$ the complementary literal of l, i.e., if $l = p$ then $\bar{l} = \neg p$ and if $l = \neg p$ then $\bar{l} = p$. For a set of literals L, $\bar{L}$ is defined as $\{\bar{l} \mid l \in L\}$. Moreover, $\overline{\mathcal{B}}$ ($\mathcal{B}$ is an interpretation over $Var(\Phi)$) corresponds to the clause $\bigvee_{p \in Var(\Phi)} f(p)$, where if $\mathcal{B}(p) = 0$ then $f(p) = p$, otherwise $f(p) = \neg p$.

Let us informally describe the most important components of modern SAT solvers. They are based on a reincarnation of the historical Davis, Putnam, Logemann and Loveland procedure, commonly called DPLL [9]. It performs a backtrack search; selecting at each level of the search tree, a decision variable which is set to a Boolean value. This assignment is followed by an inference step that deduces and propagates some forced unit literal assignments. This is recorded in the implication graph, a central data-structure, which encodes the decision literals together with there implications. This branching process is repeated until finding a model or a conflict. In the first case, the formula is answered satisfiable, and the model is reported, whereas in the second case, a conflict clause (called learnt clause) is generated by resolution following a bottom-up traversal of the implication graph [17, 24]. The learning or conflict analysis process stops when a conflict clause containing only one literal from the current decision level is generated. Such a conflict clause asserts that the unique literal with the current level (called asserting literal) is implied at a previous level, called assertion level, identified as the maximum level of the other literals of the clause. The solver backtracks to the assertion level and assigns that asserting literal to *true*. When an empty conflict clause is generated, the literal is implied at level 0, and the original formula can be reported unsatisfiable.

In addition to this basic scheme, modern SAT solvers use other components such as activity based heuristics and restart policies. An extensive overview about propositional Satisfiability can be found in [6, 15].

2.2 Frequent Pattern Mining in a Sequence of Items (FPS)

In this section, we present the frequent pattern mining problem of enumerating frequent, closed and maximal patterns with wildcards in a sequence of items [18, 20, 2]. Let us first give some preliminary definitions and notations.

Sequences of items.

Let Σ be a finite set of items, called alphabet. A *sequence of items* s over Σ is a simple sequence of symbols $s_0 \cdots s_{n-1}$ belonging to Σ. We denote by $|s|$ its length and by $\mathcal{P}_s$ the set $\{0, \ldots |s| - 1\}$ of all the locations of its symbols. *A wildcard* is a new symbol $\circ$ which is not in Σ. This symbol matches any symbol of the alphabet.

Pattern.

A *pattern* over Σ is a sequence $p = p_0 \ldots p_{m-1}$, where $p_0 \in \Sigma$, $p_{m-1} \in \Sigma$ and $p_i \in \Sigma \cup \{\circ\}$ for $i = 1, \ldots, m-2$. We say that p is included in $s = s_0 \ldots s_{n-1}$ at the location $l \in \mathcal{P}_s$, denoted $p \preceq_l s$, if $\forall i \in \{0 \ldots m-1\}$, $p_i = s_{l+i}$ or $p_i = \circ$. We also say that p is included in s, denoted $p \preceq s$, if $\exists l \in \mathcal{P}_s$ such that $p \preceq_l s$. The *cover* of p in s is defined as the set $\mathcal{L}_s(p) = \{l \in \mathcal{P}_s | p \preceq_l s\}$. Moreover, The *support* of p in s is defined as the value $|\mathcal{L}_s(p)|$.

FPS problem.

Let s be a sequence, p a pattern and $\lambda \geq 1$ a minimal support threshold, called also a quorum. We say that p is a *frequent pattern in s w.r.t. λ* if $|\mathcal{L}_s(p)| \geq \lambda$. *The frequent pattern mining problem in a sequence of items (FPS)* consists in computing the set $\mathcal{M}_s^\lambda$ of all the frequent patterns w.r.t. λ. For instance, let us consider the sequence $s = aaccbcabcba$ and then pattern $p = a \circ c$. We have $\mathcal{L}_s(p) = \{0, 1, 6\}$, since $p \preceq_0 s$, $p \preceq_1 s$ and $p \preceq_6 s$. In this case, if we consider that the minimal support threshold is equal to the value 3, then the pattern p is a frequent pattern of s.

Closed and Maximal Patterns.

A frequent pattern p of a sequence s is said to be *closed* if for any frequent pattern q satisfying $q \succ p$, there is no integer d such that $\mathcal{L}_s(q) = \mathcal{L}_s(p) + d$, where $\mathcal{L}_s(p) + d = \{l + d | l \in \mathcal{L}_s(p)\}$. Moreover, it is said to be *maximal* if for any frequent pattern q, $q \not\succ p$. Clearly, the set of closed frequent patterns (resp. maximal frequent patterns) is a condensed representation of the set of frequent patterns. Indeed, the frequent patterns can be obtained from the closed (resp. maximal) ones by replacing items with wildcards.

Note that if p_1 and p_2 are two patterns such that $p_1 \preceq p_2$, then if $|\mathcal{L}_s(p_2)| \geq \lambda$ then $|\mathcal{L}_s(p_1)| \geq \lambda$. This property is called anti-monotonicity.

3. FREQUENT PATTERN MINING IN A SEQUENCE OF ITEMSETS (FPSI)

In this section, we define a new variant of the problem of discovering patterns with wildcards in a sequence, by considering a sequence of itemsets instead of a sequence of items. The role of wildcard symbol is nicely played by the empty itemset as it match any itemset. As mentioned in the introduction, this new problem admits some similarities and differences with the classical sequential pattern mining problem introduced in [1]. The main difference resides in the definition of the notion of subsequence (inclusion), where empty itemsets are used as wildcards, and in the use or not of a single or several sequences

A *sequence of itemsets* s over an alphabet Σ is defined as a sequence $s_0, \ldots, s_{n-1}$, where $s_i \subseteq \Sigma$ for $i = 0, \ldots, n-1$. Similarly to the sequences of items, we denote by $|s|$ its length ($|s| = n$) and by $\mathcal{P}_s$ the set $\{0, \ldots |s| - 1\}$ of the locations.

A *pattern* $p = p_0, \ldots, p_{m-1}$ over Σ is also defined as a sequence of itemsets where the first and the last elements are different from the empty itemset. In this context, let us mention that we do not need the wildcard symbol. Indeed, one can use the empty itemset to match any itemset. Furthermore, we say that p is included in $s = s_0 \ldots s_{n-1}$, denoted $p \preceq_l s$, at the location $l \in \mathcal{P}_s$ if $\forall i \in \{0 \ldots m-1\}$, $p_i \subseteq s_{l+i}$. The relation $\preceq$ and the set $\mathcal{L}_s(p)$ are defined in the same way as in the case of the sequences of items. The *cover* (resp. *support*) of p in s is defined as the set $\mathcal{L}_s(p)$ (resp. as the value $|\mathcal{L}_s(p)|$).

The frequent, closed and maximal patterns are also defined in the same way. For instance, a frequent pattern p of a sequence s is said to be *closed* if for any frequent pattern q satisfying $q \succ p$, there is no integer d such that $\mathcal{L}_s(q) = \mathcal{L}_s(p) + d$, where $\mathcal{L}_s(p) + d = \{l + d | l \in \mathcal{L}_s(p)\}$. The frequent patterns can be obtained from the closed (resp. maximal) ones by replacing itemsets with their subsets.

For example, let us consider the sequence of itemsets $s = \{a, b\}, \{a, b\}, \{c, d\}, \{c, e\}, \{f\}, \{g\}, \{d\}, \{a, b, d\}, \{f\}, \{c\}$ and the pattern $p = \{a, b\}, \{\}, \{c\}$. If we set the minimal support threshold to 3, then p is a frequent pattern in s, since $\mathcal{L}_s(p) = \{0, 1, 7\}$. The pattern p is also a closed frequent pattern, but $p' = \{a\}, \{\}, \{c\}$ is not closed, since $p \prec p'$.

The pattern mining task that we consider in the sequences of itemsets allows to exhibit a high degree of self similarity for better understandings of large volumes of data. For instance, a sequence of itemsets can be seen as a record of the articles bought by a customer over a period of time. In such a case, a frequent pattern could be "the customer bought acetylsalicylic acid two days after buying beer and wine in 20% of the days from 2008 to 2012".

3.1 Related Works and Motivations

SAT-based encodings for enumerating frequent, closed and maximal patterns in the sequences of items have been proposed in [8]. They follows the constraint programing (CP) based approach proposed recently by Luc De Raedt et al. in [10] for itemset mining. The SAT and CP based approaches in data mining are proposed in order to offer declarative and flexible frameworks. Indeed, new constraints require often new implementations in specialized approaches, while they can be easily integrated in such frameworks.

In this paper, we propose a SAT-based encodings for enumerating frequent, closed and maximal patterns in the sequences of items and the sequences of itemsets. The choice of SAT comes from our desire to exploit the efficiency of modern SAT solvers [6]. In this context, our encodings can be seen as an improvement and an extension of the encodings proposed in [8]. Indeed, we show experimentally on real world data that our encodings are better than those in [8]. Furthermore, we show that encodings in the case of the sequences of itemsets are obtained by a very slight modification of that for the sequences of items.

4. A NEW SAT-BASED APPROACH FOR FPS

We describe here our new Boolean encoding for the problem of enumerating the frequent patterns in a sequence of items FPS. The base idea consists in using a propositional variable to represent the location of an element of the alphabet in the candidate pattern. Moreover, we use the well-known cardinality constraint to reason about the support of the candidate pattern.

Let $\Sigma = \{a_1, \ldots, a_m\}$ be an alphabet, s a sequence over Σ of length n and λ a minimal support threshold. We associate to each character a appearing in s a set of k_a propositional variables $p_{a,0}, \ldots, p_{a,(k_a-1)}$ such that $k_a = min(max(\mathcal{L}_s(a)) + 1, n - \lambda + 1)$. The variable $p_{a,i}$ means that a is in the candidate pattern at the location i. In fact, that explains why we associate only $min(max(\mathcal{L}_s(a)) + 1, n - \lambda + 1)$ variables to each character a, because $\{0, \ldots, min(max(\mathcal{L}_s(a)), n - \lambda)\}$ corresponds to the set of all possible locations of a in the candidate patterns.

We first need to encode that the first symbol must be a solid character (different from the wildcard symbol). This property is expressed by the following simple clause:

$$\bigvee_{a \in \Sigma} p_{a,0} \tag{1}$$

The following constraint composed of binary clauses allows us to capture the locations where the candidate pattern does not appear:

$$\bigwedge_{a \in \Sigma, 0 \leq l \leq n-1, 0 \leq i \leq k_a - 1} (p_{a,i} \wedge s_{l+i} \neq a) \rightarrow b_l \tag{2}$$

where $b_0, \ldots, b_{n-1}$ are n new propositional variables. In the previous formula $b_j = 1$ if the candidate pattern does not appear in s at the location j. Let us recall that, in classical propositional logic, we have $A \rightarrow B := \neg A \vee B$, and that explains why the previous formula can be seen as a set of binary clauses (the expressions of the for $s_{l+i} \neq a$ are constants, i.e. $s_{l+i} \neq a \in \{0, 1\}$).

In the problem of enumerating all the frequent patterns in s w.r.t. λ, we need to express that the candidate pattern occurs at least λ times. This property is obtained by the following *cardinality constraint*:

$$\sum_{l=0}^{n-1} b_l \leq n - \lambda \tag{3}$$

Indeed, if this constraint is not satisfied, then we know that there exist at least $n - \lambda + 1$ locations where the candidate pattern does not appear. This is equivalent to say that there exist at most $\lambda - 1$ locations where the candidate pattern appears, i.e. it is not frequent. Otherwise, there exist at least λ locations of the candidate pattern , i.e. it is frequent. Hence this constraint allows us to reason about the support of the considered candidate pattern an to decide whether it is greater or equal to the minimal support threshold or not.

The previous constraint involves the well known cardinality constraint (0/1 linear inequality). Several polynomial encoding of this kind of constraints into a CNF formula have been proposed in the literature. The first linear encoding of general linear inequalities to CNF have been proposed by J. P. Warners [23]. Recently, efficient encodings of the cardinality constraint to CNF have been proposed, most of them try to improve the efficiency of constraint propagation (e.g. [4, 21, 3, 16]).

PROPOSITION 1. *The problem of enumerating all frequent patterns in a given sequence s is expressed by the constraints (1), (2) and (3).*

PROOF. We first prove that if $p = a_0, \ldots, a_{k-1}$ is a frequent pattern, then there exists an extension of its corresponding Boolean interpretation $\mathcal{B}_p$ which is a model of (1), (2) and (3). Note that $\mathcal{B}_p$ is defined as follows: for all $a \in \Sigma$ and for all $i \in \{0, \ldots, k_a\}$, if $a_i = a$ then $\mathcal{B}_p(p_{a,i}) = 1$. One can easily see that the constraint (1) is satisfied by $\mathcal{B}_p$, since $\mathcal{B}_p(p_{a_0,0}) = 1$. Let us now extend the Boolean Interpretation $\mathcal{B}_p$ to the variables $b_0, \ldots, b_{n-1}$. This extension is obtained as follows: for all $0 \leq i \leq n-1$, if $p \not\preceq_i s$ then $\mathcal{B}_p(b_i) = 1$. Clearly this extension corresponds to a Boolean interpretation that satisfies (2). Finally, it also satisfies (3), since p is a frequent pattern, i.e. its support is greater or equal to the minimal support threshold λ.

Conversely, we have to prove that if a Boolean interpretation $\mathcal{B}$ is a model of (1), (2) and (3), then there exists a unique pattern $p_{\mathcal{B}}$ corresponding to $\mathcal{B}$ which is frequent. Note that, using the constraints (2) and (3), we have, for all $a, a' \in \Sigma$ and for all $i \in \{0, \ldots, k_a - 1\}$, if $\mathcal{B}(p_{a,i}) = \mathcal{B}(p_{a',i}) = 1$, then $a = a'$. Indeed, if there exists $a \neq a'$ such that $\mathcal{B}(p_{a,i}) = \mathcal{B}(p_{a',i}) = 1$, then we get $\sum_{l=0}^{n-1} b_l = n$ and this is in contradiction with $\sum_{l=0}^{n-1} b_l \leq n - \lambda$ ($\lambda \neq 0$). Furthermore, using the constraint (1), we know that the first symbol of p is different from $\circ$. Therefore, we deduce that there exists a unique pattern associated to $\mathcal{B}$. This pattern corresponds to $p_{\mathcal{B}} = a_0 \cdots a_{k-1}$ such that, for all $i \in \{0, \ldots, k-1\}$ with $a_i \neq \circ$, $\mathcal{B}(p_{a_i,i}) = 1$, and $\mathcal{B}(p_{a,i}) = 0$ for all $a \in \Sigma$ with $a \neq a_i$. Moreover, using the constraint (2), we know that if $\mathcal{B}(b_i) = 1$, then $p \not\preceq_i s$. Hence, using the cardinality constraint (3), we deduce that the support of p is greater or equal to the support threshold λ. □

Example. Consider the frequent pattern mining problem in the case of the sequence *aabb* with 2 as minimal support threshold. Our encoding corresponds to the following formulae:

$p_{a,0} \vee p_{b_0}$
$p_{a,0} \rightarrow (b_2 \wedge b_3)$
$p_{a,1} \rightarrow (b_1 \wedge b_2 \wedge b_3)$
$p_{a,2} \rightarrow (b_0 \wedge b_1 \wedge b_2 \wedge b_3)$
$p_{b,0} \rightarrow (b_0 \wedge b_1)$
$p_{b,1} \rightarrow (b_0 \wedge b_3)$
$p_{b,2} \rightarrow (b_2 \wedge b_3)$
$b_0 + b_1 + b_2 + b_3 \leq 2$

Note that, for all Boolean interpretation $\mathcal{B}$, if $\mathcal{B}(p_{a,i}) = \mathcal{B}(p_{b,i})$, then $b_0 + b_1 + b_2 + b_3 = 4$. Hence we cannot have different solid characters at the same position. Moreover, using the last constraint, for all Boolean interpretation $\mathcal{B}$ which is a model of the encoding, we must have $\mathcal{B}(p_{a,1}) = \mathcal{B}(p_{a,2}) = 0$ and $\mathcal{B}(p_{a,0}) \neq \mathcal{B}(p_{b,1})$. If we describe each Boolean model of the formula by a subset of $\{p_{a,0}, p_{a,1}, p_{a,2}, p_{b,0}, p_{b,1}, p_{b,2}\}$, then we obtain as models $\{p_{a,0}\}$, $\{p_{b,0}\}$ and $\{p_{a,0}, p_{b,2}\}$. These Boolean models correspond to the patterns a, b and $a \circ b$.

Note that in order to consider the frequent patterns with at least *min* solid characters, we just have to add the following constraint:

$$\sum_{a \in \Sigma, 0 \leq i \leq k_a - 1} p_{a,i} \geq min \tag{4}$$

Conversely, in order to only consider the frequent patterns with at most *max* solid characters, we add:

$$\sum_{a \in \Sigma, 0 \leq i \leq k_a - 1} p_{a,i} \geq max \tag{5}$$

Moreover, the combination of the two previous constraints allows us to only consider with the number of solid characters between *min* and *max*. Let us mention that such extensions show that our approach is flexible.

5. ENUMERATING CLOSED AND MAXIMAL MOTIFS

5.1 Enumerating Closed Motifs (CPS)

In order to provide constraints allowing to enumerate the closed frequent patterns, we associate to each symbol a a set of $k_a + (n - min(\mathcal{L}_s(a)) - 1)$ propositional variables:

$$p_{a,-k'_a}, \ldots, p_{a,k_a-1}$$

where $k'_a = n - min(\mathcal{L}_s(a)) - 1$. Similarly to our previous encoding, the propositional variables $p_{a,0}, \ldots, p_{a,k_a-1}$ allow us to reason about the possible locations of a in the candidate pattern. The variables with negative indices are used to force the candidate pattern to be closed. Our encoding of the problem of enumerating the closed frequent patterns in a sequence of items CPS is obtained by extending the previous one with new constraints.

We first have to capture all the locations where the candidates pattern appears. This is obtained by the following constraint:

$$\bigwedge_{l=0}^{n-1} (b_l \rightarrow \bigvee_{a\in\Sigma, 0\leq i\leq k_a-1} (p_{a,i} \wedge s_{l+i} \neq a)) \quad (6)$$

Indeed, the previous constraint combined to the constraint (2) allows us to obtain that, if the Boolean interpretation $\mathcal{B}$ is a model of the constraints (1), (2) and (6), then the candidate pattern that corresponds to $\mathcal{B}$ appears only in the locations $\{0 \leq l \leq n-1 | \mathcal{B}(b_l) = 0\}$.

Now, we introduce a necessary, but not sufficient, constraint, w.r.t. the previous constraints, for obtaining a closed frequent pattern:

$$\bigwedge_{a\in\Sigma, 0\leq i\leq k_a-1} (\bigwedge_{l=0}^{n-1} \overline{b_l} \rightarrow s_{l+i} = a) \rightarrow p_{a,i} \quad (7)$$

Intuitively, the previous constraint maximizes the number of the symbols different from wildcard on the right side of the symbol represented by the propositional variable having 0 as index.

We now define a constraint with the propositional variables having the negative indices. Conversely to the previous constraint, the following constraint allows us to to maximize the number of the symbols different from wildcard on the left side:

$$\bigwedge_{a\in\Sigma, 1\leq i\leq k'_a} (\bigwedge_{l=0}^{n-1} \overline{b_l} \rightarrow s_{l-i} = a) \leftrightarrow p_{a,-i} \quad (8)$$

Let us note that if the Boolean interpretation $\mathcal{B}$ is a model of $(1) \wedge (2) \wedge (3) \wedge (6) \wedge (7) \wedge (8)$ and $\mathcal{B}(p_{a,i}) = 1$, then, for all $b \in \Sigma$ such that $a \neq b$ and $p_{b,i}$ exists, $\mathcal{B}(p_{b,i}) = 0$ holds. This property is mainly obtained from the constraints (2) and (8) (see arguments used in the proof of Proposition 1). A closed motif is obtained from a model by using the propositional variables associated to the elements of Σ and evaluated to 1 by this model. Let $p_{a_0,i_0}, p_{a_1,i_1}, \ldots, p_{a_{k-1},i_{k-1}}$ be these variables. In this case, the closed motif is:

$$a_0 \overbrace{\circ \cdots \circ}^{i_1-i_0-1} a_1 \cdots a_{k-1}$$

We now provide another encoding without using propositional variables with negative indices. The idea consists in excluding each interpretation whenever its corresponding pattern is not closed. This encoding is obtained by replacing the constraint (8) with the following constraint:

$$\bigwedge_{a\in\Sigma, 1\leq i\leq k'_a} \neg(\bigwedge_{l=0}^{n-1} \overline{b_l} \rightarrow s_{l-i} = a) \quad (9)$$

By the previous constraint, we simply force the candidate pattern to be closed.

PROPOSITION 2. *The problem of enumerating all the closed frequent patterns in a given sequence s is expressed by the constraints (1), (2), (3), (6), (7) and (9).*

PROOF. Let $p = a_0, \ldots, a_{k-1}$ be a closed frequent pattern. We define its corresponding Boolean interpretation $\mathcal{B}_p$ as follows: for all $a \in \Sigma$ and for all $i \in \{0, \ldots, k_a\}$, if $a_i = a$ then $\mathcal{B}_p(p_{a,i}) = 1$. We extend $\mathcal{B}_p$ to the propositional variables $\{b_0, \ldots, b_{n-1}\}$ as follows: $p \not\preceq_i s$ iff $\mathcal{B}_p(b_i) = 1$, for $i = 0, \ldots, n-1$. By using similar arguments as in our proof of Proposition 1, we know that $\mathcal{B}_p$ satisfies the constraints (1), (2) and (3). It also satisfies (6), since if $\mathcal{B}_p(b_i) = 1$ then $p \not\preceq_i s$. In this context, the support of p is equal to $|\{b_i \mid \mathcal{B}_p(b_i) = 0\}|$. This allows us to deduce that $\mathcal{B}_p$ satisfies (7) and (9), since p is closed. Indeed, if (7) or (9) are not satisfied by $\mathcal{B}_p$, then there exists a pattern p' such that $p \prec p'$ with a support greater or equal to that of p and we get a contradiction because that means that p is not a closed pattern.

Conversely, consider $\mathcal{B}$ a model of (1), (2), (3), (6), (7) and (9). Using the constraints (2) and (3), we have, for all $a, a' \in \Sigma$ and for all $i \in \{0, \ldots, k_a - 1\}$, if $\mathcal{B}(p_{a,i}) = \mathcal{B}(p_{a',i}) = 1$, then $a = a'$. Hence, there exists a unique pattern associated to $\mathcal{B}$ that corresponds to $p_{\mathcal{B}} = a_0 \cdots a_{k-1}$ such that, for all $i \in \{0, \ldots, k-1\}$ with $a_i \neq \circ$, $\mathcal{B}(p_{a_i,i}) = 1$, and $\mathcal{B}(p_{a,i}) = 0$ for all $a \in \Sigma$ with $a \neq a_i$. Using Prposition 1, we know that the pattern $p_{\mathcal{B}}$ is frequent. The constraint (6) allows us to obtain that the support of p is equal to $|\{b_i \mid \mathcal{B}(b_i) = 0\}|$. Using the constraints (7) and (9), we now that there is no frequent pattern q having the same support as p such that $p \prec q$. Therefore, we deduce that p is a closed frequent pattern. □

Note that in order to enumerate all frequent patterns without any condition on their supports, we only have to remove the constraint 3.

5.2 Enumerating Maximal Motifs (MPS)

In our encoding of the problem of enumerating the maximal frequent patterns in a sequence of items MPS, we only use the propositional variables associated to the elements of Σ with positive indices, i.e. we associate to each symbol a a set of k_a propositional variables $p_{a,0}, \ldots, p_{a,(k_a-1)}$. Our encoding of MPS is obtained by extending the one of FPS in a similar way as our encoding of CPS.

In order to enumerate the maximal frequent patterns, we

need to capture all the locations where the candidates pattern appears. To this end, similarly to our encodings of CPS, we use the constraint (6). Indeed, the combination of the constraints (2) and (6) allows us to obtain, if $\mathcal{B}$ is a Boolean model of these two constraints, then $\{0 \leq l \leq n-1 | \mathcal{B}(b_l) = 0\}$ corresponds to the set of the locations where the candidate pattern appears.

We now provide the constraint allowing to maximize the number of symbols different from wildcard on the right side of the symbol represented by the propositional variable having 0 as index:

$$\bigwedge_{a \in \Sigma, 1 \leq i \leq k_a - 1} (\sum_{l=0}^{n-1} \overline{b_l} \wedge s_{l+i} = a \geq \lambda) \rightarrow p_{a,i} \quad (10)$$

Intuitively, the constraint means that if $p = a_0 \cdots a_{k-1}$ is the patter candidate and there exists $a \in \Sigma$ such that then pattern $a_0 \cdots a_{k-1} \circ \cdots \circ a$ have the same support as p, then p is not a maximal frequent pattern.

Coversely to the previous constraint, we finally introduce the constraint allowing to maximize the number of symbols different from wildcard on the left side of the symbol represented by the propositional variable having 0 as index:

$$\bigwedge_{a \in \Sigma, 1 \leq i \leq k'_a} \neg(\sum_{l=0}^{n-1} \overline{b_l} \wedge s_{l-i} = a \geq \lambda) \quad (11)$$

One can easily see that it is equivalent to the following constraint:

$$\bigwedge_{a \in \Sigma, 1 \leq i \leq k'_a} \sum_{l=0}^{n-1} \overline{b_l} \wedge s_{l-i} = a \leq \lambda - 1 \quad (12)$$

Indeed, the constraint $\neg(\sum_{l=0}^{n-1} x \geq \lambda)$ is equivalent to cardinality constraint $\sum_{l=0}^{n-1} x \leq \lambda - 1$.

PROPOSITION 3. *The problem of enumerating all the maximal frequent patterns in a given sequence s is expressed by the constraints (1), (2), (3), (6), (10) and (12).*

PROOF. Similar to our proof of Proposition 2. □

Let us mention that we can also use the constraints (4) and (5) to reason about the number of the solid characters in the considered patterns in the cases of CPS and MPS. Furthermore, we can use a constraint in order to only consider the closed and maximal patterns with support between λ and λ'. This constraint is the following:

$$\sum_{l=0}^{n-1} b_l \geq n - \lambda' \quad (13)$$

6. SAT-BASED ENCODINGS AND SEQUENCE OF ITEMSETS

In this section, we extend our SAT-based approach for discovering frequent, closed and maximal patterns in a sequence of itemsets. We will show that our encodings in this case can be obtained from the previous ones with a very slight modification.

Our encoding of the problem of enumerating the frequent patterns in a sequence of itemsets FPSI can be easily obtained from the one of FPS. We only have to replace the equalities of the form $s_{l+i} \neq a$ with $a \notin s_{l+i}$:

$$\bigvee_{a \in \Sigma} p_{a,0} \quad (14)$$

$$\bigwedge_{a \in \Sigma, 0 \leq l \leq n-1, 0 \leq i \leq k_a - 1} (p_{a,i} \wedge a \notin s_{l+i}) \rightarrow b_l \quad (15)$$

$$\sum_{l=0}^{n-1} b_l \leq n - \lambda \quad (16)$$

In this case, the variable $p_{a,i}$ means that the symbol a is in the candidate pattern in the itemset at the location i. Let us recall that we use the empty itemset as wildcard.

We denote by CPSI (resp. MPSI) the problem of enumerating the closed (resp. maximal) frequent patterns in a sequence of itemsets. Similarly to FPSI, Boolean encodings of CPSI and MPSI can be directly obtained from the ones of respectively CPS and MPS by replacing the expressions of the form $s_{l+i} \neq a$ (resp. $s_{l+i} = a$) with $a \notin s_{l+i}$ (resp. $a \in s_{l+i}$).

Constraints of closeness:

$$\bigwedge_{l=0}^{n-1} (b_l \rightarrow \bigvee_{a \in \Sigma, 0 \leq i \leq k_a - 1} (p_{a,i} \wedge a \notin s_{l+i})) \quad (17)$$

$$\bigwedge_{a \in \Sigma, 0 \leq i \leq k_a - 1} (\bigwedge_{l=0}^{n-1} \overline{b_l} \rightarrow a \in s_{l+i}) \rightarrow p_{a,i} \quad (18)$$

$$\bigwedge_{a \in \Sigma, 1 \leq i \leq k'_a} \neg(\bigwedge_{l=0}^{n-1} \overline{b_l} \rightarrow a \in s_{l-i}) \quad (19)$$

Constraints of maximality: to express the maximality, we add the following two constraints to (17):

$$\bigwedge_{a \in \Sigma, 1 \leq i \leq k_a - 1} (\sum_{l=0}^{n-1} \overline{b_l} \wedge a \in s_{l+i} \geq \lambda) \rightarrow p_{a,i} \quad (20)$$

$$\bigwedge_{a \in \Sigma, 1 \leq i \leq k'_a} \sum_{l=0}^{n-1} \overline{b_l} \wedge a \in s_{l-i} \leq \lambda - 1 \quad (21)$$

The slight modification of our encodings in the case of the sequences of items in order to obtain encodings for the sequences of itemsets clearly shows the high flexibility of our proposed framework.

7. IMPLEMENTATION AND EXPERIMENTS

In our study, we carried out a preliminary experimental evaluation of our proposed approaches using two different datasets.

1. *Bioinformatics*: proteinic data encoded as a sequence of items, where an item is an amino-acid [1].

[1] http://www.biomedcentral.com/1471-2105/11/175/additional/

2. *Synthetic datasets:* we use the well-known IBM itemset data generator[2] to derive datasets with different features. We used this generator to get sequences of itemsets, that can be seen as an ordered set of transactions.

To make fair our comparison with the approach proposed in [8], we adopted the similar choices in our implementations. First, as we deal with the problem of enumerating all the models of a given CNF formula encoding our sequence mining problem, we implemented a model enumeration solver based on the CDCL-based solver MiniSAT 2.2 [3]. To enumerate all the models, each time a model is found, we add only the negation of the sub-model restricted to the literals encoding the pattern to the formula and we restart the search. Secondly, as our SAT encodings include cardinality constraints, we also use the BDD encoding [5] using BoolVar/PB open source java library [4].

In the first experiment, we compare our new SAT encoding (noted CPS1) against the SAT encoding proposed in [8] (noted CPS2), on bioinformatics datasets (sequences of items). This first evaluation concerns the enumeration of frequent closed patterns with wild cards in a sequence of items. We consider a sequence of fixed length and we measure the evolution of computation time with respect to the minimal support threshold (quorum) λ. The quorum evolves linearly ($\lambda_0 = 5$ and $\lambda_i = \lambda_{i-1} + 5$). Several datasets have been considered, their evaluation shows similar behavior. The results obtained on a representative dataset are depicted in Figure 1. This experiment confirms that in the case of sequences of items, our new SAT-based sequence mining approach outperforms (in terms of CPU time) the approach proposed recently in [8]. We also obtain significant improvement w.r.t. the size of the encoding. For instance, if we consider the most difficult dataset of the Figure 1 (quorum $\lambda = 5$), the obtained CNF formula using our encoding contains about 19 millions of clauses and 3 millions of variables, whereas with the encoding proposed in [8] the formula contains about 30 millions of clauses and 7.5 millions of variables.

The second experiment concerns the new extension of the problem to the case of sequence of itemsets. Our goal is show the feasibility of our proposed extension and its associated encodings. For our evaluation, we considered synthetic datasets, generated using the approach outlined in [1] and also used by several authors (e.g. [12]). In our context, we only consider a single sequence of itemsets with different features (size of the sequence, number of items, average size of the itemsets). In Figure 2, we illustrate the results obtained on two datasets: dataset1 (size of the sequence = 100, avg. size of itemsets = 15, number of items = 40) and dataset2 obtained from dataset1 by cutting the sequence at 50th position. The quorum is also varied linearly as in the first experiment. The main observation that can be made, is that the hardness of the enumeration problem increase as the quorum decrease. Indeed, for smaller values of λ, the number of frequent closed patterns is huge, leading to even harder problems. However for higher values of λ, the enumeration problem becomes easy as the number of interesting patterns decreases.

[2]http://sourceforge.net/projects/ibmquestdatagen/
[3]MiniSAT: http://minisat.se/
[4]BoolVAR/PB : http://boolvar.sourceforge.net/

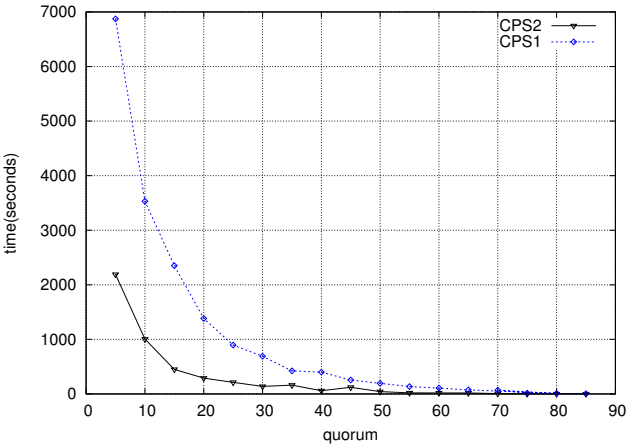


Figure 1: Bioinfo: time Vs quorum (Sequence of Items - Frequent Closed Patterns)

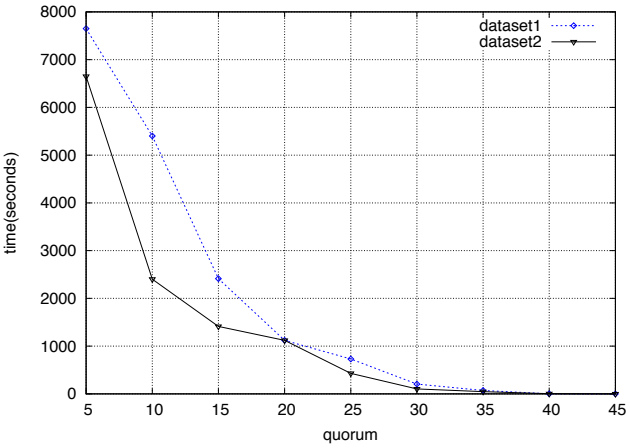


Figure 2: Synthetic datasets: time Vs quorum (Sequence of Itemsets - Frequent Closed Patterns)

As a summary, in the above experiments, we have shown that our encoding significantly improve the one proposed in [8, 7]. For comparison purposes, we used the same encoding of the cardinality constraint and also the same algorithm for enumerating all models of a CNF formula. We think that several room for future improvements can be obtained by using the state-of-the-art encoding of the cardinality [3], and by using more efficient model enumeration algorithm [13, 11, 25]. Obviously, our SAT based approach is less efficient than dedicated approaches such as the state-of-the-art algorithm proposed by Arimura et al. in [2]. However, our SAT model is declarative and highly flexible. Indeed, the SAT encoding for a sequence of itemsets is obtained with a very slight modification of the SAT encoding of a sequence of items. Also, one can easily combine several kind of constraints. Finally, SAT-based data mining benefits from the continuous progress of SAT community.

8. AKNOWLEDGMENTS

We thank the reviewers for their helpful comments. This work has been supported in part by the CNRS and the French ANR project *"DAG: Declarative Approaches for Enumerating Interesting Patterns"* under the Défis program 2009.

9. CONCLUSION AND PERSPECTIVES

The contributions of this paper are twofolds. First, we proposed an interesting improvement of the SAT-based encodings introduced in [8] for enumerating frequent, closed and maximal patterns with wildcards in a sequence of items. Secondly, we introduced a new and natural extension of the problem to deal with the sequences of itemsets. Interestingly, its encoding to SAT is obtained with a slight modification of the SAT encoding of the problem dealing with the sequences of items. This clearly shows the high flexibility of our proposed framework and opens several issues for future research. We first plan to investigate other variants of the problem such as sequences of sequences of items or itemsets. It would be interesting to extend our encoding with constraints on the form of the enumerated patterns (restriction on the number of consecutive wildcards, regular expressions, etc). Finally, on the Boolean satisfiability side, the design of efficient model generation procedures is an important issue for SAT-based data mining framework in general and to other important application domains.

10. REFERENCES

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In A. L. P. C. Philip S. Yu, editor, *Proceedings of the Eleventh International Conference on Data Engineering (ICDE'1995)*, pages 3–14. IEEE Computer Society, 1995.

[2] H. Arimura and T. Uno. An efficient polynomial space and polynomial delay algorithm for enumeration of maximal motifs in a sequence. *Journal of Combinatorial Optimization*, 13, 2007.

[3] R. Asin, R. Nieuwenhuis, A. Oliveras, and E. Rodriguez-Carbonell. Cardinality networks: a theoretical and empirical study. *Constraints*, 16(2):195–221, 2011.

[4] O. Bailleux and Y. Boufkhad. Efficient CNF Encoding of Boolean Cardinality Constraints. In *9th International Conference on Principles and Practice of Constraint Programming - CP 2003*, pages 108–122, 2003.

[5] O. Bailleux, Y. Boufkhad, and O. Roussel. A translation of pseudo boolean constraints to sat. *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)*, 2(1-4), 2006.

[6] A. Biere, M. J. H. Heule, H. van Maaren, and T. Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in AI and Applications*. IOS Press, 2009.

[7] E. Coquery, S. Jabbour, and L. Sais. A constraint programming approach for enumerating motifs in a sequence. In M. Spiliopoulou, H. Wang, D. J. Cook, J. Pei, W. Wang, O. R. Zaïane, and X. Wu, editors, *ICDM Workshops*, pages 1091–1097. IEEE, 2011.

[8] E. Coquery, S. Jabbour, L. Saïs, and Y. Salhi. A SAT-Based Approach for Discovering Frequent, Closed and Maximal Patterns in a Sequence. In *20th European Conference on Artificial Intelligence ECAI*, pages 258–263, 2012.

[9] M. Davis, G. Logemann, and D. W. Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.

[10] L. De Raedt, T. Guns, and S. Nijssen. Constraint Programming for Itemset Mining. In *Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD'2008)*, pages 204–212, Las Vegas, Nevada, USA, August 24-27 2008.

[11] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. *clasp* : A conflict-driven answer set solver. In *9th International Conference Logic Programming and Nonmonotonic Reasoning (LPNMR'2007)*, volume 4483 of *Lecture Notes in Computer Science*, pages 260–265. Springer, 2007.

[12] K. Gouda, M. Hassaan, and M. J. Zaki. Prism: A primal-encoding approach for frequent sequence mining. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007)*, pages 487–492, Omaha, Nebraska, USA, October 28-31 2007. IEEE Computer Society.

[13] O. Grumberg, A. Schuster, and A. Yadgar. Memory efficient all-solutions sat solver and its application for reachability analysis. In *In Proceedings of the 5th International Conference on Formal Methods in Computer-Aided Design (FMCAD)*, pages 275–289. Springer, 2004.

[14] T. Guns, S. Nijssen, and L. D. Raedt. Itemset mining: A constraint programming perspective. *Artificial Intelligence*, 175(12-13):1951–1983, 2011.

[15] Y. Hamadi, S. Jabbour, and L. Sais. Learning from conflicts in propositional satisfiability. *4OR*, 10(1):15–32, 2012.

[16] S. Jabbour, L. Sais, and Y. Salhi. A pigeon-hole based encoding of cardinality constraints. In *In proceedings of the 29th International Conference on Logic Programming (ICLP'2013)*, August 24-29 2013.

[17] J. P. Marques-Silva and K. A. Sakallah. GRASP - A New Search Algorithm for Satisfiability. In *Proceedings of IEEE/ACM CAD*, pages 220–227, 1996.

[18] L. Parida, I. Rigoutsos, A. Floratos, D. Platt, and Y. Gao. Pattern Discovery on Character Sets and Real-valued Data: Linear Bound on Irredundant Motifs and an Efficient Polynomial Time Algorithm. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 297–308, 2000.

[19] L. Parida, I. Rigoutsos, and D. Platt. An output-sensitive flexible pattern discovery algorithm. In *In proceedings of the 12th Annual Symposium on Combinatorial Pattern Matching (CPM'2001)*, volume 2089 of *Lecture Notes in Computer Science*, pages 131–142. Springer, 2001.

[20] N. Pisanti, M. Crochemore, R. Grossi, and M.-F. Sagot. Bases of motifs for generating repeated patterns with wild cards. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB'2005)*, 2(1):40–50, 2005.

[21] C. Sinz. Towards an Optimal CNF Encoding of Boolean Cardinality Constraints. In *11th International*

Conference on Principles and Practice of Constraint Programming - CP 2005, pages 827–831, 2005.

[22] G. Tseitin. On the complexity of derivations in the propositional calculus. In H. Slesenko, editor, *Structures in Constructives Mathematics and Mathematical Logic, Part II*, pages 115–125, 1968.

[23] J. P. Warners. A linear-time transformation of linear inequalities into conjunctive normal form. *Information Processing Letters*, 68(2):63–69, 1998.

[24] L. Zhang, C. F. Madigan, M. W. Moskewicz, and S. Malik. Efficient conflict driven learning in Boolean satisfiability solver. In *IEEE/ACM CAD'2001*, pages 279–285, 2001.

[25] W. Zhao and W. Wu. Asig: An all-solution sat solver for cnf formulas. In *11th International Conference on Computer-Aided Design and Computer Graphics, CAD/Graphics 2009, Huangshan, China, August 19-21 (CAD/Graphics)*, pages 508–513. IEEE, 2009.

Decomposition Based SAT Encodings for Itemset Mining Problems

S. Jabbour, L. Sais, Y. Salhi. Decomposition Based SAT Encodings for Itemset Mining Problems. Advances in Knowledge Discovery and Data Mining - 19th Pacific-Asia Conference, PAKDD (2) 2015 : Lecture Notes in Computer Science 9078, 662-674.

Decomposition Based SAT Encodings for Itemset Mining Problems

Said Jabbour, Lakhdar Sais, and Yakoub Salhi

CRIL - CNRS, Université d'Artois
Rue Jean Souvraz, SP-18 62307, Lens Cedex 3
{jabbour,sais,salhi}@cril.fr

Abstract. Recently, several constraint programming (CP)/propositional satisfiability (SAT) based encodings have been proposed to deal with various data mining problems including itemset and sequence mining problems. This research issue allows to model data mining problems in a declarative way, while exploiting efficient and generic solving techniques. In practice, for large datasets, they usually lead to constraints network/Boolean formulas of huge size. Space complexity is clearly identified as the main bottleneck behind the competitiveness of these new declarative and flexible models w.r.t. specialized data mining approaches. In this paper, we address this issue by considering SAT based encodings of itemset mining problems. By partitioning the transaction database, we propose a new encoding framework for SAT based itemset mining problems. Experimental results on several known datasets show significant improvements, up to several orders of magnitude.

Keywords: Declarative Data Mining, Itemset Mining

1 Introduction

Recently, a constraint programming (CP) based data mining (DM) framework was proposed by Luc De Raedt et al. in [10] for itemset mining (CP4IM). This new framework offers a declarative and flexible representation model. New constraints often require new implementations in specialized approaches, while they can be easily integrated in such a CP framework. It allows data mining problems to benefit from several generic and efficient CP solving techniques. The authors show how some typical constraints (e.g. frequency, maximality, monotonicity) used in itemset mining can be formulated for use in CP [5]. This study leads to the first CP approach for itemset mining displaying nice declarative opportunities. Encouraged by these promising results, several contributions addressed other data mining problems using the two well-known AI models: constraint programming and propositional satisfiability. For example, in [7], the authors proposed a SAT based formulation of the problem of enumerating the Top-k frequent closed itemsets. In [2], the authors solve the frequent itemset mining problem by compiling the set of all itemset into a binary decision diagram (BDD)

(augmented with counts). Frequent itemset are then extracted by querying the BDD. By considering the relationship between local constraint-based mining and constraint satisfaction problems, Khiari et al. [8] proposed a model for mining patterns combining several local constraints, i.e., patterns defined by n-ary constraints. Also, several constraint-based language for modeling and solving data mining problems have been designed. Let us mention, the constraint-based language, defined in [9], which enables the user to define queries in a declarative way addressing pattern sets and global patterns. All primitive constraints of the language are modeled and solved using the SAT framework. More recently, Guns at al. [3], introduced a general-purpose, declarative mining framework called MiningZinc. Compared to CP4IM framework [4], MiningZinc supports a wide variety of different solvers (including DM algorithms and general purpose solvers) and employe a significantly more high-level modeling language.

The above non exhaustive description of some recent works on constraint programming based data mining shows an important research activity in this new research trend. In most of these contributions, particularly for SAT based data mining framework, despite the nice declarative aspects, the authors pointed out that the main challenge concern their competitiveness against specialized data mining algorithms. As mentioned in [3], "this is non-trivial because data mining algorithms are highly optimized for specific tasks and large datasets, while generic constraint solvers may struggle in particular with the size of the problems". The work presented in this paper fit into this framework. Our goal is to provide a step in that direction by pushing forward the effectiveness of SAT-based data mining approaches.

In this paper, we consider the SAT-based encodings of itemset mining problems proposed in [7]. An enhancement of the encoding is provided. We propose an original partition based approach allowing us to partition the whole problem into sub-problems of reasonable size while maintaining solving incrementality. It takes as input a transaction database and a partition of the set of items, then it incrementally generates and solves a sequence of sub-problems while ensuring completeness. This partition based SAT encoding improves Jabbour et al. [7] SAT-based itemset mining approach by several orders of magnitude on several well-known datasets.

2 Preliminaries

Let Ω be a finite non empty set of symbols, called *items*. From now on, we assume that this set is fixed. We use the letters a, b, c, etc to range over the elements of Ω. An *itemset* I over Ω is defined as a subset of Ω, i.e., $I \subseteq \Omega$. We use 2^{Ω} to denote the set of itemsets over Ω and we use the capital letters I, J, K, etc to range over the elements of 2^{Ω}.

A *transaction* is an ordered pair (i, I) where i is a natural number, called *transaction identifier*, and I an itemset, i.e., $(i, I) \in \mathbb{N} \times 2^{\Omega}$. A *transaction database* $\mathcal{D}$ is defined as a finite non empty set of transactions ($\mathcal{D} \subseteq \mathbb{N} \times 2^{\Omega}$) where each transaction identifier refers to a unique itemset.

Let $\mathcal{D}$ be a transaction database and I an itemset. The *cover* of I in $\mathcal{D}$, denoted $\mathcal{C}(I, \mathcal{D})$, is the following set of transaction identifiers:

$$\{i \in \mathbb{N} \mid (i, J) \in \mathcal{D} \text{ and } I \subseteq J\}$$

Moreover, the *support* of I in $\mathcal{D}$, denoted $\mathcal{S}(I, \mathcal{D})$, corresponds to the cardinality of $\mathcal{C}(I, \mathcal{D})$, i.e., $\mathcal{S}(I, \mathcal{D}) = |\mathcal{C}(I, \mathcal{D})|$.

Tid	Itemset
1	a, b, c, d
2	a, b, e, f
3	a, b, c
4	a, c, d, f
5	g
6	d
7	d, g

Table 1. A Transaction Database $\mathcal{D}$.

For instance, consider the transaction database $\mathcal{D}$ in Table 1. In this case, we have $\mathcal{C}(\{a, b\}, \mathcal{D}) = \{1, 2, 3\}$ and $\mathcal{S}(\{a, b\}, \mathcal{D}) = 3$ while $\mathcal{S}(\{f\}, \mathcal{D}) = 2$.

In this work, we are mainly interested in the problem of finding frequent itemsets (FIM problem). Given a transaction database $\mathcal{D}$ and a natural number n greater than 0, solving this problem consists in computing the following set of itemsets: $FIM(\mathcal{D}, n) = \{I \subseteq \Omega \mid \mathcal{S}(I, \mathcal{D}) \geqslant n\}$. In this context, we call n a *minimal support threshold*.

The number of frequent itemsets in a transaction database can be significant. Indeed, this problem is $\#P$-hard [15]. The complexity class $\#P$ corresponds to the counting problems associated with a decision problem in NP. In order to partially face this problem, condensed representations have been proposed:

Definition 1 (Closed Frequent Itemsets). *Let $\mathcal{D}$ be a transaction database and n a minimal support threshold and I an itemset in $FIM(\mathcal{D}, n)$. The itemset I is closed iff, for all $J \supset I$, $\mathcal{S}(I, \mathcal{D}) > \mathcal{S}(J, \mathcal{D})$.*

Definition 2 (Maximal Frequent Itemsets). *Let $\mathcal{D}$ be a transaction database and n a minimal support threshold and I an itemset in $FIM(\mathcal{D}, n)$. The itemset I is maximal iff, for all $J \supset I$, $\mathcal{S}(J, \mathcal{D}) < n$.*

For instance, in the previous example, for $n = 2$, the sets of closed and maximal frequent itemsets are respectively $\{\{a, b\}, \{a, b, c\}, \{d\}, \{a, c, d\}, \{a, f\}, \{g\}\}$ and $\{\{a, b, c\}, \{a, c, d\}, \{a, f\}, \{g\}\}$. We use $CFIM(\mathcal{D}, n)$ (resp. $MFIM(\mathcal{D}, n)$) to denote the set of closed (resp. maximal) frequent itemsets.

3 Propositional Logic and SAT Problem

In this section, we define the syntax and the semantics of propositional logic. Let Prop be a countably set of propositional variables. We use the letters p,

q, r, etc to range over Prop. The set of *propositional formulæ*, denoted Form, is defined inductively started from Prop, the constant $\bot$ denoting false, the constant $\top$ denoting true, and using the logical connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$. We use $\mathcal{P}(A)$ to denote the set of propositional variables appearing in the formula A. The equivalence connective $\leftrightarrow$ is defined by $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$.

A *Boolean interpretation* $\mathcal{I}$ of a formula A is defined as a function from $\mathcal{P}(A)$ to $\{0,1\}$ (0 corresponds to $false$ and 1 to $true$). It is inductively extended to propositional formulæ as usual:

$$\mathcal{I}(\bot) = 0 \quad \mathcal{I}(\top) = 1 \quad \mathcal{I}(A \wedge B) = min(\mathcal{I}(A), \mathcal{I}(B))$$
$$\mathcal{I}(\neg A) = 1 - \mathcal{I}(A) \quad \mathcal{I}(A \vee B) = max(\mathcal{I}(A), \mathcal{I}(B))$$
$$\mathcal{I}(A \rightarrow B) = max(1 - \mathcal{I}(A), \mathcal{I}(B))$$

A *model* of a formula A is a Boolean interpretation $\mathcal{I}$ that satisfies A, i.e. $\mathcal{I}(A) = 1$. A formula A is satisfiable if there exists a model of A. A is *valid* or *a theorem*, if every Boolean interpretation is a model of A. We use $Mod(A)$ to denote the set of models of A.

Let us now define the *conjunctive normal form* (CNF) representation of propositional formulæ. A CNF formula is a conjunction ($\wedge$) of clauses, where a *clause* is a disjunction ($\vee$) of literals. A *literal* is a propositional variable (p) or a negated propositional variable ($\neg p$). The two literals p and $\neg p$ are called *complementary*. A CNF formula can also be seen as a set of clauses, and a clause as a set of literals. The size of the CNF formula A corresponds to the value $\sum_{c \in A} |c|$ where $|c|$ is the number of literals in the clause c. Let us mention that any propositional formula can be translated to a CNF formula equivalent w.r.t. satisfiability, using linear Tseitin's encoding [13]. The *SAT problem* consists in deciding if a given CNF formula admits a model or not.

4 SAT Encoding of Itemset Mining

In this section, we describe SAT encodings for itemset mining which are mainly based on the encodings proposed in [7]. In order to do this, we fix, without loss of generality, a transaction database $\mathcal{D} = \{(1, I_1), \ldots, (m, I_m)\}$ and a minimal support threshold n.

The SAT encoding of itemset mining that we consider is based on the use of propositional variables representing the items and the transaction identifiers in $\mathcal{D}$. More precisely, for each item a (resp. transaction identifier i), we associate a propositional variable, denoted p_a (resp. q_i). These propositional variables are used to capture all possible itemsets and their covers. Formally, given a model $\mathcal{I}$ of the considered encoding, the candidate itemset is $\{a \in \Omega \mid \mathcal{I}(p_a) = 1\}$ and its cover is $\{i \in \mathbb{N} \mid \mathcal{I}(q_i) = 1\}$.

The first propositional formula that we describe allows us to obtain the cover of the candidate itemset:

$$\bigwedge_{i=1}^{m} (\neg q_i \leftrightarrow \bigvee_{a \in \Omega \setminus I_i} p_a) \tag{1}$$

This formula expresses that q_i is true if and only if the candidate itemset is supported by the i^{th} transaction. In other words, the candidate itemset is not supported by the i^{th} transaction (q_i is false), when there exists an item a (p_a is true) that does not belong to the transaction ($a \in \Omega \setminus I_i$).

The following propositional formula allows us to consider the itemsets having a support greater than or equal to the minimal support threshold:

$$\sum_{i=1}^{m} q_i \geqslant n \tag{2}$$

This formula corresponds to 0/1 linear inequalities, usually called cardinality constraints. The first linear encoding of general 0/1 linear inequalities to CNF have been proposed by J. P. Warners in [14]. Several authors have addressed the issue of finding an efficient encoding of cardinality (e.g. [12, 11, 1]) as a CNF formula. Efficiency refers to both the compactness of the representation (size of the CNF formula) and to the ability to achieve the same level of constraint propagation (generalized arc consistency) on the CNF formula.

We use $\mathcal{E}_{FIM}(\mathcal{D}, n)$ to denote the encoding corresponding to the conjunction of the two formulæ (1) and (2).

Proposition 1 ([7]). *Let $\mathcal{D}$ be a transaction database and n a minimal support threshold. $\mathcal{I}$ is a model of $\mathcal{E}_{FIM}(\mathcal{D}, n)$ iff $I = \{a \in \Omega \mid \mathcal{I}(p_a) = 1\}$ is a frequent itemset where $\mathcal{C}(I, \mathcal{D}) = \{i \in \mathbb{N} \mid \mathcal{I}(q_i) = 1\}$.*

We now describe the propositional formula allowing to force the candidate itemset to be closed:

$$\bigwedge_{a \in \Omega} (\bigwedge_{i=1}^{m} q_i \rightarrow a \in I_i) \rightarrow p_a \tag{3}$$

This formula means that if we have $\mathcal{S}(I, \mathcal{D}) = \mathcal{S}(I \cup \{a\}, \mathcal{D})$ then $a \in I$ holds. This condition is necessary and sufficient to force the candidate itemset to be closed. Let us note that the expressions of the form $a \in I_i$ correspond to constants, i.e., $a \in I_i$ corresponds to $\top$ if the item a is in I_i, to $\bot$ otherwise.

Note that the formula (3) can be simply reformulated as a conjunction of clauses as follows:

$$\bigwedge_{a \in \Omega} ((\bigvee_{1 \leqslant i \leqslant m, a \notin I_i,} q_i) \vee p_a) \tag{4}$$

This reformulation is obtained using the equivalence $A \rightarrow B \equiv \neg A \vee B$.

For illustration purposes, and to show the generality of our proposed framework, we give below the encoding of the maximality constraint. Indeed, the following formula corresponds to a constraint forcing the candidate itemset to be maximal:

$$\bigwedge_{a \in \Omega} (\sum_{i=1}^{m} (q_i \wedge a \in I_i) \geqslant n) \rightarrow p_a \tag{5}$$

Since the expressions of the form $a \in I_i$ correspond to constants, the formulæ of the form $\sum_{i=1}^{m} (q_i \wedge a \in I_i) \geqslant n$ correspond to cardinality constraints. Indeed,

for every propositional variable q_i, we have $q_i \wedge \bot = \bot$ and $q_i \wedge \top = q_i$. In the same way as in the case of (3), the formula (5) provide a necessary and sufficient condition to force the candidate itemset to be maximal, since we have, for every $a \in \Omega$, $\mathcal{S}(I \cup \{a\}, \mathcal{D}) \geqslant n$ implies $a \in I$. It is worth noticing that this formula is not provided in [7]. However, an equivalent constraint is provided in De Raedt et al's encoding proposed in [10].

We use $\mathcal{E}_{CFIM}(\mathcal{D}, n)$ (resp. $\mathcal{E}_{MFIM}(\mathcal{D}, n)$) to denote the encoding corresponding to the conjunction of the formulæ (1), (2) and (4) (resp. (1), (2) and (5)).

Proposition 2. *Let $\mathcal{D}$ be a transaction database and n a minimal support threshold. $\mathcal{I}$ is a model of $\mathcal{E}_{CFIM}(\mathcal{D}, n)$ iff $I = \{a \in \Omega \mid \mathcal{I}(p_a) = 1\}$ is a closed frequent itemset where $\mathcal{C}(I, \mathcal{D}) = \{i \in \mathbb{N} \mid \mathcal{I}(q_i) = 1\}$.*

Proposition 3. *Let $\mathcal{D}$ be a transaction database and n a minimal support threshold. $\mathcal{I}$ is a model of $\mathcal{E}_{MFIM}(\mathcal{D}, n)$ iff $I = \{a \in \Omega \mid \mathcal{I}(p_a) = 1\}$ is a maximal frequent itemset where $\mathcal{C}(I, \mathcal{D}) = \{i \in \mathbb{N} \mid \mathcal{I}(q_i) = 1\}$.*

5 A Partition Based Method

One of the significant problems of the declarative approaches in data mining is the large size of the encodings. Indeed, even if the sizes of the encodings is polynomial in the size of the input, this does not mean that these sizes are reasonable. In order to tackle this problem, we propose an approach which allows us to decompose the encodings described previously.

Let $\mathcal{D}$ be a transaction database and S an itemset. We use $\mathcal{D}_{|S}$ to denote the transaction database $\{(i, I) \in \mathcal{D} \mid I \cap S \neq \emptyset\}$. Let k be a natural number smaller than or equal to $|\Omega|$. A *k-partition* of Ω is a structure of the form $(\{S_1, \ldots, S_k\}, \prec)$ where $\{S_1, \ldots, S_k\}$ is a partition of Ω into k subsets and $\prec$ is an ordering on $\{S_1, \ldots, S_k\}$.

Let $\mathcal{D}$ be a transaction database, $\mathcal{P} = (W, \prec)$ a k-partition of Ω, $S \in W$ and n a minimal support threshold. We use $\Sigma_{DM}(\mathcal{D}, n, S, \mathcal{P})$ to denote the following propositional formula:

$$\mathcal{E}_{DM}(\mathcal{D}_{|S}, n) \wedge (\bigvee_{a \in S} p_a) \wedge (\bigwedge_{b \in \bigcup_{S' \prec S} S'} \neg p_b) \tag{6}$$

where $DM \in \{FIM, CFIM, MFIM\}$.

In formula 6, the subformula $(\bigvee_{a \in S} p_a)$ expresses that the itemsets must contains at least one item from S. The second subformula $(\bigwedge_{b \in \bigcup_{S' \prec S} S'} \neg p_b)$ avoid generating itemsets previously generated using the previous elements of the partition sequence.

Proposition 4. *Let $\mathcal{D}$ be a database, $\mathcal{P} = (W, \prec)$ a k-partition of Ω, $S \in W$ and n a minimal support threshold. $\mathcal{I}$ is a model of $\Sigma_{DM}(\mathcal{D}, n, S, \mathcal{P})$ iff the following properties are satisfied:*

1: **procedure** $SAT^{\mathcal{P}}_{DM}(\mathcal{D},n,\mathcal{P})$ ▷ $\mathcal{P} = (\{S_1, \ldots, S_k\}, \prec)$
2: $R \leftarrow \emptyset$
3: **for** $i \leftarrow 1, k$ **do** ▷ $S_1 \prec \cdots \prec S_k$
4: $R \leftarrow R \cup enumModels(\Sigma_{DM}(\mathcal{D}, n, S_i, \mathcal{P}))$
5: **end for**
6: **return** $itemsets(R)$
7: **end procedure**

Fig. 1. Algorithm $SAT^{\mathcal{P}}_{DM}$

(i) $I = \{a \in \Omega \mid \mathcal{I}(p_a) = 1\} \in DM(\mathcal{D}, n)$;
(ii) $\mathcal{C}(I, \mathcal{D}) = \{i \in \mathbb{N} \mid \mathcal{I}(q_i) = 1\}$;
(iii) $I \cap S \neq \emptyset$; *and*
(iv) $I \cap \bigcup_{S' \prec S} S' = \emptyset$.

Proof. Part $\Rightarrow$. Using Propositions 1, 2 and 3, we obtain the properties (i) and (ii). The properties (iii) and (iv) are directly obtained from the formulæ $\bigvee_{a \in S} p_a$ and $\bigwedge_{b \in \bigcup_{S' \prec S} S'} \neg p_b$ respectively.

Part $\Leftarrow$. Using the properties (i) and (ii), we obtain that $\mathcal{I}$ is a model of $\mathcal{E}_{DM}(\mathcal{D}_{|S}, n)$. This is a consequence of Propositions 1, 2 and 3. Since $I \cap S \neq \emptyset$ (resp. $I \cap \bigcup_{S' \prec S} S' = \emptyset$), we get that $\mathcal{I}$ is a model of $\bigvee_{a \in S} p_a$ (resp. $\bigwedge_{b \in \bigcup_{S' \prec S} S'} \neg p_b$).

Proposition 5. *Let* $\mathcal{D}$ *be a transaction database,* $\mathcal{P} = (W, \prec)$ *a* k*-partition of* Ω*,* $S, S' \in W$ *such that* $S \neq S'$ *and* n *a minimal support threshold. Then* $Mod(\Sigma_{DM}(\mathcal{D}, n, S, \mathcal{P})) \cap Mod(\Sigma_{DM}(\mathcal{D}, n, S', \mathcal{P})) = \emptyset$ *holds.*

Proof. Since $S \neq S'$, we have either $S \prec S'$ or $S' \prec S$. We here consider, without loss of generality, that $S \prec S'$. Since all models of $\Sigma_{DM}(\mathcal{D}, n, S', \mathcal{P})$ satisfy $\bigwedge_{a \in S} \neg p_a$ because of $S' \prec S$ and all models of $\Sigma_{DM}(\mathcal{D}, n, S, \mathcal{P})$ satisfy $\bigvee_{a \in S} p_a$, we deduce that $Mod(\Sigma_{DM}(\mathcal{D}, n, S, \mathcal{P})) \cap Mod(\Sigma_{DM}(\mathcal{D}, n, S', \mathcal{P})) = \emptyset$.

We provide in Figure 1 a simple algorithm using our partition based method for enumerating all (closed/maximal) frequent itemsets. This algorithm takes as input a transaction database $\mathcal{D}$, a minimal support threshold n and a k-partition $\mathcal{P} = (\{S_1, \ldots, S_k\}, \prec)$ such that $S_1 \prec \cdots \prec S_k$. The procedure $enumModels(A)$ enumerate all Boolean models of the propositional formula A. The procedure $itemsets(R)$ returns the itemsets corresponding to the Boolean models in the set R.

Theorem 1 (Soundness). *The algorithm* $SAT^{\mathcal{P}}_{DM}$ *is sound w.r.t. the data mining task* DM.

Proof. The soundness of $SAT^{\mathcal{P}}_{DM}$ is a direct consequence of Proposition 4.

Theorem 2 (Completeness). *The algorithm* $SAT^{\mathcal{P}}_{DM}$ *is complete w.r.t. the data mining task* DM.

Proof. The completeness of $SAT^{\mathcal{P}}_{DM}$ comes from the fact that $\Omega = \bigcup_{1 \leqslant i \leqslant k} S_i$ and Proposition 4.

It is worth noticing that, using Proposition 5, we know that there is no itemset computed in two different steps in the for-loop.

In the propositional formula (6), the main difficulty is in the sub-formula $\mathcal{E}_{DM}(\mathcal{D}_{|S}, n)$. Indeed, if the number of transactions of $\mathcal{D}_{|S}$ is close to that of $\mathcal{D}$, then the gain is minimal. For this reason, it is more interesting to consider orderings $\prec$ in k-partitions that are based on the number of transactions in the sense that $S \prec S'$ if the number of transactions of $\mathcal{D}_{|S}$ is smaller than that of $\mathcal{D}_{|S'}$. Let us note that the number of transactions of $\mathcal{D}_{|S}$ depends directly on the size of S: if S is included in S', then the number of transactions of $\mathcal{D}_{|S}$ is smaller than that of $\mathcal{D}_{|S'}$.

6 Experiments

In this section, we carried out an experimental evaluation of the performance of our partition based approach for enumerating frequent closed itemsets. We considered a variety of datasets taken from the `FIMI`[1] and `CP4IM`[2] repositories.

All the experiments were done on Intel Xeon quad-core machines with 32GB of RAM running at 2.66 Ghz. For each instance, we used a timeout of 2 hours of CPU time. As described in the previous section, the Algorithm 1 takes a

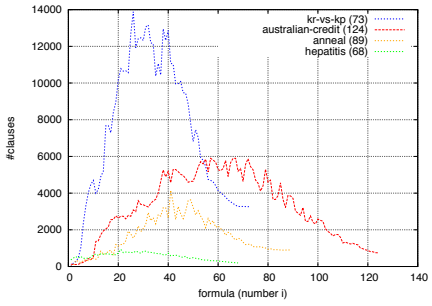


Fig. 2. $SAT^{\mathcal{P}}_{CFIM}$ Formulas: Evolution of the Number of Clauses

transaction database and a k-partition of the set of items as inputs, and returns the set of patterns of interest (frequent, closed or maximal). In our experiments, we consider a partition $\mathcal{P}$ where each $S_i \in \mathcal{P}$ contains a single item and $S_i \prec S_{i+1}$ if S_i contains a less frequent item than S_{i+1}. Obviously, in this case k is equal to $|\Omega|$. This ordering allows us to first generate encodings of smaller size. Our goal is to show the feasibility of our proposed approach even when a basic partitioning

[1] FIMI: http://fimi.ua.ac.be/data/
[2] CP4IM: http://dtai.cs.kuleuven.be/CP4IM/datasets/

scheme is considered. Finding a k-partition of the set of items that leads to better improvements is an interesting optimisation problem. This issue is out of the scope of this paper. We compare our partition based approach noted $SAT^{\mathcal{P}}_{CFIM}$ ($SAT\text{-}P\text{-}CFIM$ in the figures) with SAT_{CFIM}, the encoding of frequent closed itemsets mining problem without partition (named $SAT\text{-}CFIM$ in the figures).

We implemented the Algorithm 1 in C. For the enumeration of all models of the Boolean formula encoding the corresponding itemset mining problem, we use the extension of modern SAT solvers described in [6]. The implementation is based on an extension of MiniSAT 2.2 [3].

Note that, the time needed to generate the partition does not exceed 1 seconds on the majority of the instances, except for `splice-1` and `connect`, which takes 17 and 60 seconds respectively. First, we present in Table 2, the charac-

instance	#trans	#items	#vars	#clauses	min	max
zoo-1	101	36	173	2196	4	387
Hepatitis	137	68	273	4934	193	934
Lymph	148	68	284	6355	28	724
audiology	216	148	508	17575	4	657
Heart-cleveland	296	95	486	15289	192	2534
Primary-tumor	336	31	398	5777	117	801
Vote	435	48	531	14454	225	2591
Soybean	650	50	730	22153	4	1847
Australian-credit	653	125	901	48573	108	5896
Anneal	812	93	990	39157	4	4149
Tic-tac-toe	958	27	1012	18259	485	3619
german-credit	1000	112	1220	73223	319	6957
Kr-vs-kp	3196	73	3342	121597	4	13879
Hypothyroid	3247	88	3419	143043	4	14410
chess	3196	75	3346	124797	4	14853
splice-1	3190	287	3764	727897	4	105540
mushroom	8124	119	8348	747635	23	34695
connect	67558	129	67815	5877720	297	291139

Table 2. Characteristics of the instances & Encoding size

teristics of the dataset and the size of the CNF formula encoding the whole transaction database for the closed itemsets mining problem. For each instance, we mention the number of transactions ($\#trans$), the number of items ($\#items$), the size of the formula encoding the whole problem in terms of number of variables ($\#vars$) and clauses ($\#clauses$). The last two columns give the number of clauses of the smallest (min) respectively the largest formula (max) generated using our partition based encoding. On all instances, the number of clauses of the largest formula (max) generated with our partition based encoding is significantly smaller than those generated without partition ($\#clauses$).

In Figure 2, we represent the evolution of the size of the resulting CNFs during the partition process for a sample of representative instances. For each instance, we mention its name and the number of generated Boolean formulas (in parenthesis) corresponding to the number of elements in the partition (k). As we can observe, our approach allows to generate small CNFs compared to the approach encoding the whole transaction database (see Table 2). For example, if we consider the `Kr-vs-kp` instance, the largest generated CNF formula does

[3] MiniSAT: http://minisat.se/

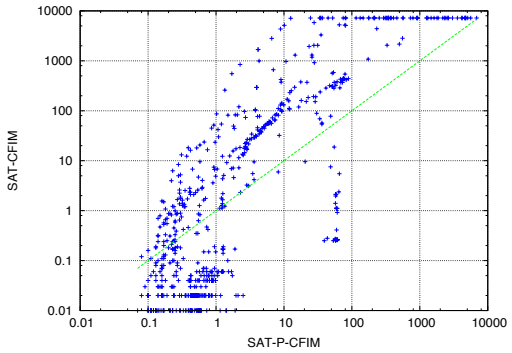


Fig. 3. $SAT^{\mathcal{P}}_{CFIM}$ vs SAT_{CFIM}

not exceed 14 000 clauses while it reaches 121 597 clauses without partitioning. Moreover, the size of the different formulas evolves as follows. During the first steps of the partitioning process, the size of the obtained CNFs is smaller, as the first elements of the partition contains less frequent items. For these formulas, the enumeration process takes relatively short time. The same observation can be made for the latest steps of the partitioning process (last elements of the partition). We recall that at iteration i, the transactions containing the items from S_j $(j < i)$ are removed. A peak in the size of the generated formulas can be observed in the middle of the partitioning process. Indeed, for the first generated formulas, the considered items appears in less transactions leading to an encoding of a smaller transaction database. For higher elements of the partition, even if they correspond to frequent items, some of their occurrences are avoided by the previous removed items. The peak in size corresponds to the steps of the partitioning process, where items occurs a reasonable number of times.

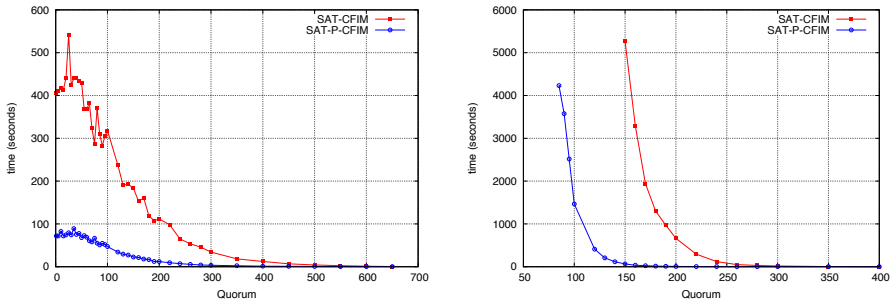


Fig. 4. $SAT^{\mathcal{P}}_{CFIM}$: `anneal` (left) and `australian` (right) instances

To evaluate the performances of our proposed approach, we compare the time needed for our partition based approach $SAT^{\mathcal{P}}_{CFIM}$ with SAT_{CFIM} (with-

out partitioning) to enumerate all models corresponding to all closed frequent itemsets. The comparison is depicted by the the scatter plot of Figure 3. Each dots (x, y) represents an instance (see Table 2) with a fixed minimal support threshold n. The scatterplot represent all the instances described in Table 2 (18 instances), where for each instance, we tested 70 different values of n. The total number of instances is 1260. The x-axis (respectively y-axis) represents the CPU time (seconds) needed for the enumeration of all closed frequent itemsets using $SAT^{\mathcal{P}}_{CFIM}$ (respectively SAT_{CFIM}). Clearly, the partition based approach outperforms those without partition on the majority of instances. For several instances and values of n, our partition based approach allows to enumerate all models while the classical method without partitioning is not able to enumerate them under the time limit. For instance, for `connect` data, SAT_{CFIM} is able to enumerate all closed frequent itemsets for only $n = 50000$, while our partition based approach enumerates all models for 8 different minimal support threshold (for all $n \geqslant 5500$).

Figures 4 and 5, highlight the results obtained by $SAT^{\mathcal{P}}_{CFIM}$ and SAT_{FCIM} on `anneal`, `australian`, `hepatitis` and `mushroom` instances while varying the minimum support threshold (quorum). These figures highlights the great potential of our partitioning based approach.

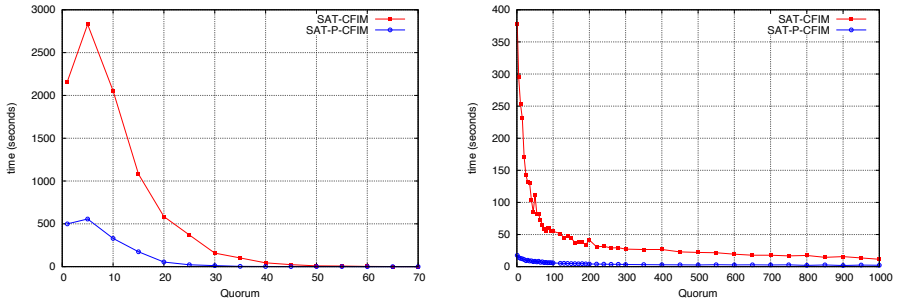


Fig. 5. $SAT^{\mathcal{P}}_{CFIM}$: `hepatitis` (left) and `mushroom` (right) instance

7 Acknowledgements

This work is supported by the French ANR Agency under theTUPLES project. We thank the reviewers for their comments that helped us to improve the paper.

8 Conclusion and perspectives

In this paper, a partition based SAT encoding of itemset mining problems is proposed. It aims to decompose the original problem into subproblems of reasonable size generated by partitioning the set of items. Our proposed approach is incremental and complete. The experimental evaluation on several known datasets shows significant improvements, up to several orders of magnitude. The results obtained in this paper, open several interesting paths for future works, including

the design of parallel based approaches. Handling the different subproblems in parallel, will leads to substantial additional improvements. Finding the "best" partition of the set of items is another interesting issue.

References

1. Asin, R., Nieuwenhuis, R., Oliveras, A., Rodriguez-Carbonell, E.: Cardinality networks: a theoretical and empirical study. Constraints 16(2), 195–221 (2011)
2. Cambazard, H., Hadzic, T., O'Sullivan, B.: Knowledge compilation for itemset mining. In: ECAI'10. pp. 1109–1110 (2010)
3. Guns, T., Dries, A., Tack, G., Nijssen, S., De Raedt, L.: Miningzinc: A modeling language for constraint-based mining. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. pp. 1365–1372. IJCAI'13 (2013)
4. Guns, T., Nijssen, S., De Raedt, L.: Itemset mining: A constraint programming perspective. Artificial Intelligence 175(12-13), 1951–1983 (Aug 2011)
5. Guns, T., Nijssen, S., Raedt, L.D.: Itemset mining: A constraint programming perspective. Artif. Intell. 175(12-13), 1951–1983 (2011)
6. Jabbour, S., Lonlac, J., Sais, L., Salhi, Y.: Extending modern sat solvers for models enumeration. In: Proceedings of the 11th IEEE International Conference on Information Reuse and Integration (IEEE-IRI'14), (To appear). San Francisco, September 13-15th (2014), published in CoRR 2013 - http://arxiv.org/abs/1305.0574
7. Jabbour, S., Sais, L., Salhi, Y.: The top-k frequent closed itemset mining using top-k sat problem. In: European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD'13). pp. 403–418 (2013)
8. Khiari, M., Boizumault, P., Crmilleux, B.: Combining csp and constraint-based mining for pattern discovery. In: Computational Science and Its Applications ICCSA 2010. pp. 432–447 (2010)
9. Metivier, J.P., Boizumault, P., Crémilleux, B., Khiari, M., Loudni, S.: A Constraint-based Language for Declarative Pattern Discovery. In: Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on. pp. 1112–1119. Vancouver, Canada (2011)
10. Raedt, L.D., Guns, T., Nijssen, S.: Constraint programming for itemset mining. In: ACM SIGKDD. pp. 204–212 (2008)
11. Silva, J.P.M., Lynce, I.: Towards robust cnf encodings of cardinality constraints. In: 13th International Conference on Principles and Practice of Constraint Programming (CP 2007). pp. 483–497 (2007)
12. Sinz, C.: Towards an optimal cnf encoding of boolean cardinality constraints. In: 11th International Conference on Principles and Practice of Constraint Programming - CP 2005. pp. 827–831 (2005)
13. Tseitin, G.: On the complexity of derivations in the propositional calculus. In: Structures in Constructives Mathematics and Mathematical Logic, Part II. pp. 115–125 (1968)
14. Warners, J.P.: A linear-time transformation of linear inequalities into conjunctive normal form. Information Processing Letters (1996)
15. Yang, G.: The complexity of mining maximal frequent itemsets and maximal frequent patterns. In: In KDD 04: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data mining. pp. 344–353. ACM Press (2004)

Clustering Complex Data Represented as Propositional Formulas

A. Boudane, S. Jabbour, L. Sais and Y. Salhi. Clustering Complex Data Represented as Propositional Formulas. Advances in Knowledge Discovery and Data Mining - 21st Pacific-Asia Conference, PAKDD (2) 2017, Lecture Notes in Computer Science 10235, 441-452.

Clustering Complex Data Represented as propositional formulas

Abdelhamid Boudane, Said Jabbour, Lakhdar Sais, and Yakoub Salhi

CRIL-CNRS, Université d'Artois, F-62307 Lens Cedex, France
{boudane,jabbour,sais,salhi}@cril.fr

Abstract. Clustering has been extensively studied to deal with different kinds of data. Usually, datasets are represented as a n-dimensional vector of attributes described by numerical or nominal categorical values. Symbolic data is another concept where the objects are more complex such as intervals, multi-categorical or modal. However, new applications might give rise to even more complex data describing for example customer desires, constraints, and preferences. Such data can be expressed more compactly using logic-based representations. In this paper, we introduce a new clustering framework, where complex objects are described by propositional formulas. First, we extend the two well-known k-means and hierarchical agglomerative clustering techniques. Second, we introduce a new divisive algorithm for clustering objects represented explicitly by sets of models. Finally, we propose a propositional satisfiability based encoding of the problem of clustering propositional formulas without the need for an explicit representation of their models. Preliminary experimental results validating our proposed framework are provided.

1 Introduction

Clustering is a technique used to recover hidden structure in a dataset obtained by grouping data into clusters of similar objects. It is derived by several important applications ranging from scientific data exploration, to information retrieval, and computational biology (e.g. [1]). Such diversity in terms of application domains induces a variety of data types and clustering techniques (see. [2] for a survey). Indeed, data can be transactional, sequential, trees, graphs, texts, or even of a symbolic nature [6, 7, 10]. This last kind of data is particularly suitable for modeling complex and heterogeneous objects usually described by a set of multivalued variables of different types (e.g. intervals, multi-categorical or modal) (e.g. [3, 4, 8]). We can also mention conceptual clustering proposed more than thirty years ago by Michalski [14] and defined as a machine learning task. It accepts a set of object descriptions (events, facts, observations, ...) and produces a classification scheme over them. Conceptual clustering not only partitions the data, but generates clusters that can be summarized by a conceptual description. As a summary, conceptual and symbolic clustering are twoparadigm proposed to deal with kinds of data other than those usually described by numerical values.

In today's data-driven digital era, data might be even more complex and heterogenous. Such complex data might represent customers desires or preferences

collected in different possible ways using surveys and quizzes. As an example, one can cite configuration systems usually designed to provide customized products satisfying the different requirements of the customer, usually modeled by constraints or logic-formulas (e.g. [11]). These customers requirements-data or the data-models provided by the configuration systems are some kind of complex data that we are interested in. These data can be represented by logic-formulas (requirements) or by models (the products satisfying the requirements). Data can also represent more complex entities such as transaction databases. Indeed, suppose that we collected several transaction databases from stores chain selling the same products, one can be interested in determining similar stores (clusters) or stores with the same behavior. This could help the manager of the stores chain to better define its trade policy. In the two previous examples, data can be better represented as a set of propositional formulas or as sets of models.

In this paper, we introduce a new clustering framework, where complex objects are described by propositional formulas. We first extend the two well known k-means and hierarchical agglomerative clustering techniques. Then, we introduce a new divisive algorithm for clustering objects represented explicitly by sets of models. Finally, we propose a propositional satisfiability based encoding of clustering propositional formulas without the need for an explicit representation of their models. Preliminary experimental results validating our proposed framework are provided before concluding.

1.1 Propositional Satisfiability

Let $\mathcal{P}$ be a countably infinite set of propositional variables. The set of *propositional formulas*, denoted $F_{\mathcal{P}}$, is defined inductively starting from $\mathcal{P}$, the constant $\bot$ denoting absurdity, the constant $\top$ denoting true, We use the greek letters ϕ, ψ to represent formulas. A *Boolean interpretation* $\mathcal{I}$ of a formula ϕ is defined as a function from $\mathcal{P}(\phi)$ to $\{0,1\}$ (0 for $false$ and 1 for $true$). A *model* of a formula ϕ is a Boolean interpretation $\mathcal{I}$ that satisfies ϕ (written $\mathcal{I} \vDash \phi$), i.e. $\mathcal{I}(\phi) = 1$. We denote the set of models of ϕ by $\mathcal{M}(\phi)$. A formula ϕ is satisfiable (or consistent) if there exists a model of ϕ; otherwise it is called unsatisfiable (or inconsistent).

Let ϕ and ψ be two propositional formulas, we say that ψ is a logical consequence of ϕ, written $\phi \vDash \psi$, iff $\mathcal{M}(\phi) \subseteq \mathcal{M}(\psi)$. The two formulas ϕ and ψ are called equivalent iff $\phi \vDash \psi$ and $\psi \vDash \phi$, i.e $\mathcal{M}(\phi) = \mathcal{M}(\psi)$.

A CNF formula is a conjunction ($\wedge$) of clauses, where a *clause* is a disjunction ($\vee$) of literals. A *literal* is a propositional variable (p), called positive literal, or ($\neg p$), called negative literal. The *SAT problem* consists in deciding whether a given CNF formula admits a model or not. Another problem related to SAT is the SAT model enumeration problem. Enumeration requires generating all models of a problem instance without duplicates. Models enumeration is related to #SAT, the problem of computing the number of models for a given propositional formula. Model counting is the canonical #P-complete problem. On the practical side, for model counting, *SampleCount* a sampling based approach proposed by Gomes et al in [9], provides very good lower bounds with high confidence. Similarly, an efficient model enumeration algorithm has been proposed in [12, 5].

2 Motivating Example

To motivate our proposed framework, let us consider a simple example of a car dealer selling different cars bands with several possible options. For each car brand, several colors and types of fuels are available. The car dealer collected the preferences of four customers through a survey questionnaire. The first customer does not want red cars. The second wants a car with a diesel fuel, while the third wants a red car with gasoline fuel. Finally, the fourth customer prefers brand Peugeot cars. In addition to these customer desires, we also consider mutual exclusion constraints (mutex), allowing to express that each car must have only one color, one type of fuel and one car brand.

To express the different customer desires in propositional logic, we consider the following propositional variables: r (resp. b) represents red (resp. black) colors, p (resp. c) represents the Peugeot (resp. Citroen) car brand and d (resp. g) represents cars with diesel (resp. gasoline) fuel.

The mutex constraints are expressed by the following formula: $\mu = [(r \wedge \neg b) \vee (b \wedge \neg r)] \wedge [(g \wedge \neg d) \vee (d \wedge \neg g)] \wedge [(p \wedge \neg c) \vee (c \wedge \neg p)]$.

In Figure 1 (left hand side), for each customer c_i, we associate a propositional formula ϕ_{c_i} expressing its desires. We also provide the set of models satisfying both the desires of the customer and the mutex constraints ($\mathcal{M}(\phi_{c_i} \wedge \mu)$). The presentation of the models follows the variables ordering: $r \prec b \prec d \prec g \prec c \prec p$. In Figure 1 (right hand side), we give a graphical representation of the preferences of the four customers. This illustrative example highlights the expressiveness of

Customers	c_1	c_2	c_3	c_4
ϕ_{c_i}	$\neg r$	d	$g \wedge r$	p
$\mathcal{M}(\phi_{c_i} \wedge \mu)$	010101	011001		010101
	010110	011010		011001
	011001	101001	100101	100101
	011010	101010	100110	101001

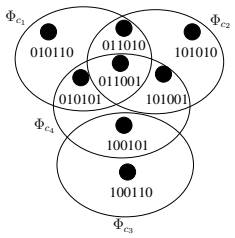


Fig. 1. Logical and Graphical Representation of Customers Preferences

logic-based data representation while allowing the possibility to define both user and background constraints.

3 Adapting Standard Clustering Algorithms

In this section, we present our extension of the well-known k-means and agglomerative hierarchical clustering algorithms to handle objects expressed as propositional formulas. Let us first fix some necessary notations and definitions.

We use $\mathcal{P}(k, \Phi)$ to denote the problem of clustering the set of propositional formulas $\Phi = \{\phi_1, \ldots, \phi_n\}$ into a set of k clusters with $k \leqslant n$. Let $\mathcal{C}$ be a family of sets over Φ. $\mathcal{C}$ is a solution of $\mathcal{P}(k, \Phi)$ if and only if $|\mathcal{C}| = k$, $\bigcup_{\mathcal{C}_i \in \mathcal{C}} \mathcal{C}_i = \Phi$ with $C_i \cap C_j = \emptyset$ for $1 \leqslant i < j \leqslant k$, and $\mathcal{M}(\bigwedge_{\phi \in \mathcal{C}_i} \phi) \neq \emptyset$ for every $\mathcal{C}_i \in \mathcal{C}$. We say that a clustering problem $\mathcal{P}(k, \Phi)$ is *consistent* if it admits a solution.

3.1 *k*-Means Algorithm for propositional formulas Clustering

Given a set of n data points in d-dimensional space $\mathbb{R}^d$ and a positive integer k, the k-means algorithm determines a set of k points in $\mathbb{R}^d$, called centers, so as to minimize an objective function such as the mean squared distance from each data point to its nearest center. To extend the k-means algorithm to clustering of objects described by propositional formulas, we need to define,

1. a distance between two formulas;
2. a centroid representing a given cluster;
3. an objective function to optimize.

Let us recall that a propositional formula ϕ can be equivalently expressed by its set of models $\mathcal{M}(\phi)$. With this representation in mind, one can consider that two formula ϕ_1 and ϕ_2 are similar if their set of common models $\mathcal{M}(\phi_1) \cap \mathcal{M}(\phi_2)$ is higher with respect to the remaining (distinctive) models $\mathcal{M}(\phi_1) \setminus \mathcal{M}(\phi_2) \cup \mathcal{M}(\phi_2) \setminus \mathcal{M}(\phi_1)$. This kind of similarity is related to the well-known contrast model of similarity proposed in a seminal paper by Tversky [15].

Definition 1 (Tversky [15]). *Let a and b be two objects described by two sets of features A and B respectively. Similarity between a and b, denoted $s(a, b)$, is defined as:*

$$s\,(a, b) = \frac{f\,(A \cap B)}{f\,(A \cap B) + \alpha f\,(A - B) + \beta f\,(B - A)} \quad \alpha, \beta \geqslant 0$$

The positive coefficients α and β reflects the weights given to the distinctive features of the two objects a and b. We usually assume that f is a matching function satisfying the additivity property $f(A \cup B) = f(A) + f(B)$, whenever A and B are disjoint. The ratio model defines a normalized value of similarity such that $0 \leqslant s(a, b) \leqslant 1$.

Contrast similarity model is particularly suitable in our context. To extend Definition 1, we consider the relationship between set operations and logical connectives. Indeed, the set union (resp. intersection) corresponds to disjunction (resp. conjunction). The difference between sets can be expressed using both conjunction and negation connectives, while the symmetric difference between sets can be expressed using the xor ($\oplus$) logical connective. Indeed, we have $\mathcal{M}(\phi_1) \setminus \mathcal{M}(\phi_2) \cup \mathcal{M}(\phi_2) \setminus \mathcal{M}(\phi_1) = \mathcal{M}((\phi_1 \wedge \neg\phi_2) \vee (\phi_2 \wedge \neg\phi_1)) = \mathcal{M}(\phi_1 \oplus \phi_2)$.

Using these relationships, we derive the following extension of the ratio model[16].

Definition 2. *Let a and b be two objects described by two propositional formulas ϕ_1 and ϕ_2 respectively. Similarity between a and b is defined as:*

$$s(a,b) = \frac{f(\phi_1 \wedge \phi_2)}{f(\phi_1 \wedge \phi_2) + \alpha f(\phi_1 \wedge \neg\phi_2) + \beta f(\phi_2 \wedge \neg\phi_1)} \quad \alpha, \beta \geqslant 0$$

In our context, as no distinction is made between the measure of $\phi_1 \wedge \neg\phi_2$ and $\phi_2 \wedge \neg\phi_1$, we derive the following similarity measure.

Definition 3. *Let a and b be two objects described by two propositional formulas ϕ_1 and ϕ_2 respectively. Similarity between a and b is defined as:*

$$s(a,b) = \frac{f(\phi_1 \wedge \phi_2)}{f(\phi_1 \wedge \phi_2) + \gamma f(\phi_1 \oplus \phi_2)}, \gamma \geqslant 0$$

From Definition 2 (resp. Definition 3), instantiating $\alpha = \beta = 1$ (resp. $\gamma = 1$), we derive a logic-based variant of the well known Jaccard similarity coefficient (resp. distance) [13]:

Definition 4. *Let a and b be two objects described by two propositional formulas ϕ_1 and ϕ_2 respectively. Similarity and distance between a and b or between ϕ_1 and ϕ_2 are defined respectively as:*

$$s_J(a,b) = s_J(\phi_1,\phi_2) = \frac{f(\phi_1 \wedge \phi_2)}{f(\phi_1 \vee \phi_2)} \text{ and } d_J(a,b) = 1 - s_J(a,b) = d_J(\phi_1,\phi_2)$$

As mentioned previously, considering the model based representation of propositional formulas, we define the function f as:

$$f : \left| \begin{array}{l} F_{\mathcal{P}} \longrightarrow \mathbf{N} \\ \phi \longmapsto |\mathcal{M}(\phi)| \end{array} \right.$$

Clearly, the function f satisfies the additive property. Indeed, we have $\mathcal{M}(\phi_1 \vee \phi_2) = \mathcal{M}(\phi_1) \cup \mathcal{M}(\phi_2)$. Computing f involves solving a #P-Complete model counting problem as discussed in Section 1.1.

Let us now define the representative of a cluster of propositional formulas.

Definition 5. *Let $\mathcal{C}_i$ be a cluster involving n_i formulas $\{\phi_{1_i}, \phi_{2_i}, \ldots, \phi_{n_i}\}$. We define the cluster representative (also called centroid) $\mathcal{O}_{\mathcal{C}_i}$ of the cluster $\mathcal{C}_i$ as:*

$$\mathcal{O}_{\mathcal{C}_i} = \phi_{1_i} \wedge \phi_{2_i} \wedge \ldots \wedge \phi_{n_i}$$

It is important to note that in our proposed extension, the goal is to group formulas into consistent clusters. Consequently, the *formula representing a given cluster must be consistent.*

We use the classical k-means objective function introduced in Definition 6

Definition 6. *Let $\mathcal{P}(k, \Phi)$ be the problem of clustering a set of propositional formulas $\Phi = \{\phi_1, \ldots, \phi_n\}$ to $k(\leqslant n)$ clusters $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$. The objective function is defined using Absolute-Error Criterion (AEC):*

$$C^* = \arg\min_{\mathcal{C}} \sum_{i=1}^{k} \sum_{\phi \in \mathcal{C}_i} d_J(\phi, \mathcal{O}_{\mathcal{C}_i}) \tag{1}$$

Our clustering algorithm of a set of propositional formulas can now be derived from the classical k-means algorithm using the new components (distance, centroid and objective function) defined above.

3.2 Hierarchical Agglomerative Algorithm for propositional formulas Clustering

Hierarchical algorithms can behave better than the k-means. The base idea of hierarchical agglomerative algorithms is to build a dendrogram such that at each level the two closest clusters are merged. By applying a hierarchical algorithm, we will ensure that if there are two objects that are closest to each other, they will necessarily be in the same cluster. In this adaptation, the similarity between two clusters is identical to the similarity between their representatives. Similarly to Definition 5, the conjunction of all formulas in a cluster represents its centroid. To merge clusters, we combine the two clusters with the smallest centroid distance. Using this adaptation, we can applay a standard hierarchical agglomerative algorithm on data represented as boolean formulas as illustrated in figure 2.Note that this algorithm needs at least $\mathcal{O}(n^2)$ calls to a # SAT oracle.

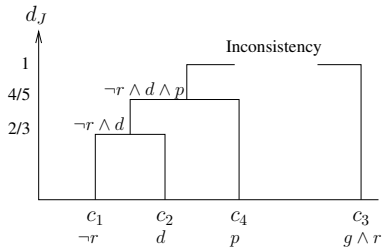


Fig. 2. Agglomerative Clustering on the Car Dealer Example

4 Divisive Algorithm for Model Based Representation

As mentioned previously, when we consider the problem of clustering a set of formulas $\Phi = \{\phi_1, \phi_1, \ldots, \phi_n\}$ without common model, i.e., $\Phi \vdash \bot$, agglomerative algorithm and k-means can fail to find a clustering with the desired number of clusters. In the sequel, we propose a top-down hierarchical (or divisive) algorithm for clustering a set of propositional formulas. Our proposed adaptation makes use of the well-known minimum hitting sets problem, that we recall.

Definition 7. *H is a hitting set of a set of sets Ω if $\forall S \in \Omega$, $H \cap S \neq \emptyset$. A hitting set H is irreducible if there is no other hitting set H' s.t $H' \subset H$. H is called minimum hitting set if there is no hitting set H' such that $|H'| < |H|$.*

Example 1. Let $\Phi = \{\phi_1, \phi_2, \phi_3\}$ be a set of propositional formulas such that $\mathcal{M}(\phi_1) = \{m_1, m_2, m_3\}$, $\mathcal{M}(\phi_2) = \{m_1, m_4\}$ and $\mathcal{M}(\phi_3) = \{m_3, m_5\}$. The set $H = \{m_1, m_3\}$ is a minimum and irreducible hitting set of the models of Φ.

In our adaptation, we choose the worst cluster to divide according to the following quality measure.

Definition 8. *Let $\mathcal{C}_i = \{\phi_{1_i}, \ldots, \phi_{n_i}\}$ be a cluster of n_i propositional formulas. We define the quality of $\mathcal{C}_i$ as:*

$$\mathcal{Q}(\mathcal{C}_i) = \frac{|\mathcal{M}(\phi_{1_i} \wedge \cdots \wedge \phi_{n_i})|}{|\mathcal{M}(\phi_{1_i} \vee \cdots \vee \phi_{n_i})|}$$

The quality of a cluster is obtained by extending the similarity measure between two formulas to a set of formulas. Indeed, a cluster is qualified to be of poor quality, when its formulas admits a great number of models while sharing a small number of models. Consequently, the worst cluster is obtained as follows:

$$\mathcal{C}_i^* = \underset{\mathcal{C}_i \in \mathcal{C}}{\operatorname{argmin}}\ \mathcal{Q}(\mathcal{C}_i)$$

Definition 9. *Let Φ be a set of propositional formulas and $\mathcal{I}$ a Boolean interpretation. We define the subset of formulas of Φ sharing the model $\mathcal{I}$ as $\mathcal{S}(\mathcal{I}, \Phi) = \{\phi \in \Phi \mid \mathcal{I} \vDash \phi\}$*

To build consistent clusters, Algorithm 1 starts by computing a minimum hitting set H of the set of sets of models of the formulas in Φ (line 1). The main idea behind our algorithm is to use the models of the computed minimum hitting set to divide a cluster into several consistent clusters. Each cluster is obtained by selecting for each model m of the minimum hitting set, the set of formulas admitting m as a model. In this way, the formulas in the obtained clusters share at least one model. If the size of the minimum hitting set H is greater than k, then no clustering is possible, and the algorithm returns an empty set (line 3), otherwise a consistent clustering can be obtained. In this last case, the algorithm starts by a clustering $\mathcal{C}$ where all the formulas in Φ are grouped into a single cluster (line 6). We start an iterative top-down divisive process (lines 7-20), until generating k clusters. At each iteration, we choose a cluster to divide (line 8) which is one of those with the worst quality (see Definition 8). Then, we build Ω the set of sets of models of the formulas involved in the selected cluster, while removing the set of common models M (lines 9-10). A minimum hitting set H of Ω is then computed (line 11). It is important to note that by removing the common models M from the models of each formula of the selected cluster, we avoid the trivial minimum hitting sets of size 1. Now, we use the hitting set H to divide the chosen cluster $\mathcal{C}_i^*$ into $|H|$ clusters (line 12). Indeed, for each model m in H, we associate a cluster Ψ_m made of formulas of $\mathcal{C}_i^*$ sharing the model m. In this way, we maintain the consistency property on each new cluster Ψ_m. Now, we substitute in $\mathcal{C}$ the cluster of poor quality $\mathcal{C}_i^*$ with the new set of clusters (line 18). However, this is only done when the size of the new

Algorithm 1: Model-Based Divisive Algorithm for Clustering Boolean Formulas

Input: A set of formulas $\Phi = \{\phi_1, \ldots, \phi_n\}$ and an integer $k \geqslant 1$
Output: A set of clusters $\mathcal{C} = \{C_1, \ldots, C_k\}$

1 $H \leftarrow minHittingSet(\{\mathcal{M}(\phi_1), \ldots, \mathcal{M}(\phi_n)\})$;
2 **if** $(|H| > k)$ **then**
3 | **return** $\emptyset$;
4 **end**
5 **else**
6 | $\mathcal{C} \leftarrow \{\Phi\}$;
7 | **while** $(|\mathcal{C}| \; != k)$ **do**
8 | | $\mathcal{C}_i^* = \{\phi_{i_1} \ldots \phi_{i_{n_i}}\} \leftarrow \underset{\mathcal{C}_i \in \mathcal{C}, |\mathcal{C}_i|>1}{\arg\min} \mathcal{Q}(\mathcal{C}_i)$, $\triangleright \; n_i = |\mathcal{C}_i^*|$;
9 | | $M = \mathcal{M}(\phi_{i_1}) \cap \cdots \cap \mathcal{M}(\phi_{i_{n_i}})$;
10 | | $\Omega = \{\mathcal{M}(\phi_{i_1}) \setminus M, \ldots, \mathcal{M}(\phi_{i_{n_i}}) \setminus M\}$;
11 | | $H \leftarrow minHittingSet(\Omega)$;
12 | | $\forall m \in H, \Psi_m \leftarrow \mathcal{S}(m, \mathcal{C}_i^*)$;
13 | | **if** $(|\mathcal{C}| + |H| - 1 > k)$ **then**
14 | | | $\Psi \leftarrow merge(\{\Psi_{m_1}, \ldots, \Psi_{m_{|\mathcal{C}|+|H|-1-k}}\})$;
15 | | | $\mathcal{C} \leftarrow (\mathcal{C} \setminus \mathcal{C}_i^*) \cup \{\Psi\} \cup \{\Psi_{m_{|\mathcal{C}|+|H|-k}}, \ldots, \Psi_{m_{|H|}}\}$
16 | | **end**
17 | | **else**
18 | | | $\mathcal{C} \leftarrow (\mathcal{C} \setminus \mathcal{C}_i^*) \cup \{\Psi_{m_1}, \ldots, \Psi_{m_{|H|}}\}$
19 | | **end**
20 | **end**
21 **end**
22 $\mathcal{C} \leftarrow eliminateOverlap(\mathcal{C})$;
23 **return** $\mathcal{C}$

clustering does not exceed k (line 13); otherwise to obtain exactly k clusters, we merge (function `merge`) the first $|\mathcal{C}| + |H| - (k+1)$ of these new clusters (line 14) before applying substitution (line 15). Note that in the divisive step (line 12), a formula can belong to several new clusters. The reason comes from the fact that a given formula can share several models of the minimum hitting set. Consequently, a last step is then performed to produce non overlapping clusters (line 20 - function *eliminateOverlap*). To do this, for each formula occurring in several clusters, we keep it in the cluster with the best quality, while removing it in the remaining clusters. Obviously, depending on applications, overlapping clusters might be more suitable. In this case, one only need to skip the call to the overlap elimination function.

Algorithm 1, involves $\mathcal{O}(n)$ calls to model enumeration problem (line 1), $\mathcal{O}(k)$ calls to # SAT oracle (line 8) and $\mathcal{O}(k)$ calls to minimum hitting set problem (line 1 and 11).

Let us now gives some interesting properties of our propositional formulas based divisive algorithm. The first one states the correctness of our algorithm.

Proposition 1. *If $\mathcal{P}(k, \Phi)$ is consistent, then Algorithm 1 produces a clustering.*

The proof trivially follows from the previous detailed explanation on how the algorithm operates.

The second property allows us to establish that two equivalent formulas might be located in the same cluster when overlaps between clusters are allowed.

Proposition 2. *Let $\mathcal{P}(k, \Phi)$ be a clustering problem with overlaps, $\mathcal{C}$ a clustering of $\mathcal{P}(k, \Phi)$ and $\phi_1, \phi_2 \in \Phi$. If $\phi_1 \equiv \phi_2$ then $\forall \mathcal{C}_i \in \mathcal{C}$, $\phi_1 \in \mathcal{C}_i$ iff $\phi_2 \in \mathcal{C}_i$.*

The last property generalizes the previous property to the case of two formulas where one is a logical consequence of the other.

Proposition 3. *Let $\mathcal{P}(k, \Phi)$ be a clustering problem with overlaps, $\mathcal{C}$ a clustering of $\mathcal{P}(k, \Phi)$ and $\phi_1, \phi_2 \in \Phi$. If $\phi_1 \vdash \phi_2$ then $\forall \mathcal{C}_i \in \mathcal{C}$, if $\phi_1 \in \mathcal{C}_i$ then $\phi_2 \in \mathcal{C}_i$.*

5 SAT encoding for a Bounded Consistent Clustering

As discussed in the previous section, when the propositional formulas are not represented by their models, our proposed model based divisive algorithm requires $\mathcal{O}(n)$ calls to model enumeration oracle, to compute the set of models of each formula. Such set of models might be of exponential size in the worst case. In addition to these limitations, one also need to compute a minimum hitting set of a set of sets of models ($\mathcal{O}(k)$ calls). In this section, we present an alternative approach that significantly reduces the overall complexity of our Algorithm. To this end, we introduce a SAT-based encoding that allows to find a bounded consistent clustering of a given set of propositional formulas.

Let $\Phi = \{\phi_1, \ldots, \phi_n\}$ be a set of propositional formulas and k a positive integer. To define our encoding, we associate to each propositional variable p appearing in Φ a set of k fresh propositional variables, denoted $p^1, \ldots, p^k$. Then, for every formula $\phi_i \in \Phi$ and $j \in \{1, \ldots, k\}$, we use ϕ_i^j to denote the formula obtained from ϕ_i by replacing each propositional variable p with the fresh variable p^j. The formula ϕ_i^j is used to model the fact that ϕ_i is in the j^{th} cluster.

The following formula expresses that each formula in Φ has to be true in at least one consistent cluster:

$$\bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{k} \phi_i^j \right) \tag{2}$$

One can easily see that (2) is satisfiable if and only if Φ can be partitioned in k consistent clusters. It is worth noting that in a model of (2) a formula can belong to more than one cluster. To obtain a bounded consistent clustering from a model m, we only have to consider for each formula $\phi_i \in \Phi$ a single positive integer j in the set $\{1 \leqslant j \leqslant k \mid m(\phi_i^j) = 1\}$. This problem can be avoided by reformulation. To this end, we associate to each formula ϕ_i in Φ a set of k fresh propositional variables, denoted $q_{\phi_i}^1, \ldots, q_{\phi_i}^k$. The variable $q_{\phi_i}^j$ is used to represent the fact that ϕ_i is in the j^{th} cluster by using the following formula:

$$\bigwedge_{i=1}^{n} \left(\bigwedge_{j=1}^{k} q_{\phi_i}^j \Leftrightarrow \phi_i^j \right) \tag{3}$$

Then, to express that each formula in Φ belongs to exactly one consistent cluster, we use the following formula:

$$\bigwedge_{i=1}^{n}(\sum_{j=1}^{k} q_{\phi_i}^{j} = 1) \tag{4}$$

Our second SAT encoding of the bounded consistent clustering problem $\mathcal{P}(k, \Phi)$ is defined by the formula $\mathcal{P}_{SAT}(k, \Phi) = (3) \wedge (4)$. From a model m of $\mathcal{P}_{SAT}(k, \Phi)$, a clustering can be easily extracted. Indeed, if $m(q_{\phi_i}^{j}) = true$ then $\phi_i \in \mathcal{C}_j$ otherwise $\phi_i \notin \mathcal{C}_j$.

Definition 10. *Let $\Phi = \{\phi_1, \ldots, \phi_n\}$. C is called a minimum consistent clustering of Φ if there is no consistent clustering $\mathcal{C}'$ of Φ such that $|\mathcal{C}'| < |\mathcal{C}|$.*

As we can observe, clustering propositional formulas can be done using Algorithm 1 by replacing the computation of the minimum hitting set with the computation of the minimum consistent clustering (Definition 10) using $\mathcal{P}_{SAT}(k, \Phi)$. Similarly to Algorithm 1, Properties 1, 2 and 3 holds.

6 Experimentation

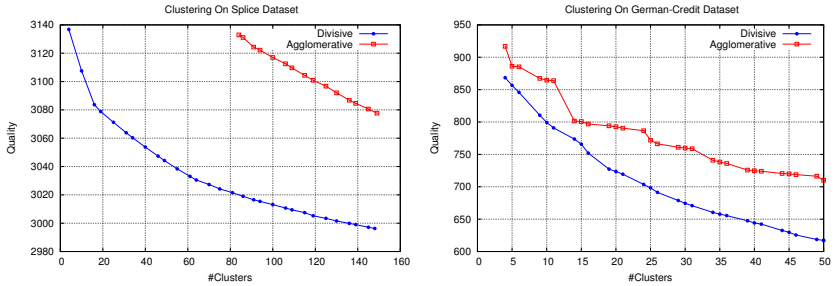
In this section, we carried out an experimental evaluation of the performance of our divisive and agglomerative algorithms for the clustering of a set of propositional formulas. Our goal is to assess the feasibility and effectiveness of our proposed framework.

We performed our experiments on a machine with Intel Core2 Quad CPU of 2.66GHz and 8G of RAM. Our first aim is to compare the performance of our divisive and agglomerative algorithms. To this end, We consider two datasets `splice`, and `german-credit`[1]. We consider each data set as a set of transactions, where each transaction is a formula (a set of models). Consequently, an item is assimilated to a model.

Figure 3 shows the performances of agglomerative (Algorithm **??**) and divisive (Algorithm 1) methods on the problem of clustering transaction databases. First, our divisive algorithm outperforms the agglomerative algorithm on `splice` and `german-credit`. Nevertheless, as illustrated in section 3.2, the agglomerative algorithm is unable to find a clustering all the time. This is the case on `splice` data, where such approach can not provide clustering answer when the number of desired clusters is less than 84.

To further investigate the expressiveness and the ability of our approach to scale, we enlarge our experiments of the previous problem by studying the clustering of a set of formulas resulting from a random-generated poll with 100 to 1000 participants where each participant is invited to report its preferences. The questions of the poll are organized in four levels. At the first level, the participant is invited to select its 3 preferred options among 5. According to the

[1] https://dtai.cs.kuleuven.be/CP4IM/



splice german-credit

Fig. 3. Model approach: Agglomerative vs Divisive

preferences of the participant, she/he is invited to select other preferences from the second level and so on until the last level (level 4). For illustration, assume that in the first level we consider a set S of courses (e.g. Artificial Intelligence, Data Mining, Databases, Networks and Web Programming). A student selects three courses from S (level 1). Then, for each selected course, she/he chooses chapters (level 2), and so on. The preferences of each participant are encoded as a propositional formula (the resulting formulas have between 567 and 1813 models). Agglomerative approach is not considered since it can not guaranty to find a clustering solution if it exists.

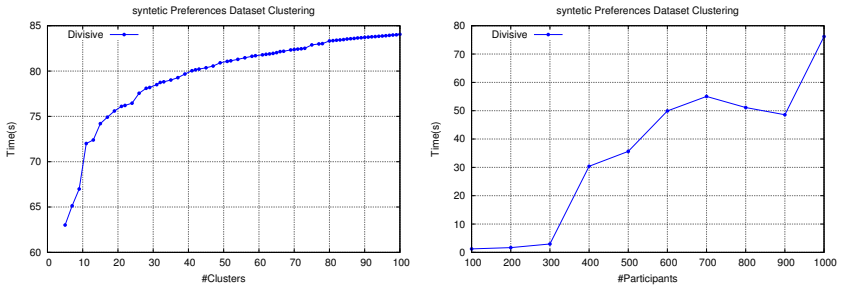


Fig. 4. Time vs #Clusters vs #Participants

The time needed to obtain a clustering, Figure 4, does not exceed 100 seconds for all values of k. This shows that our approach scale well. Finally, we study the evolution of the time needed to find a clustering when the number of clusters is fixed to 20 and the number of participants is varied from 100 to 1000 (Figure 4). Here again the time needed is reasonable, i.e., less than 100 seconds.

7 Conclusion et perspectives

In this work we introduced the concept of consistent clustering propositional formulas. We show how well-known k-means, agglomerative and divisive algorithms

can be adapted to this new framework. We then, propose two new solutions. The first one called model based, assume that the set of models of each formula are given. We then show how the hitting set notion is used to efficiently give a consistent clustering. In the second part, we propose an encoding into SAT of the divisive algorithm that make a linear number of calls to a #SAT oracle to count the set of models during the clustering steps. As a future work, we plan to explore other similarity measure, to define intuitive distance between propositional formulas. Improving our divisive algorithm by exploiting efficiently the overlaps deserves further investigation.

References

1. C. C. Aggarwal and C. K. Reddy. *Data clustering: algorithms and applications*. CRC Press, 2013.
2. P. Berkhin. A survey of clustering data mining techniques. In J. Kogan, C. K. Nicholas, and M. Teboulle, editors, *Grouping Multidimensional Data - Recent Advances in Clustering*, pages 25–71. Springer, 2006.
3. L. Billard and E. Diday. *Symbolic Data Analysis: Conceptual Statistics and Data Mining*. John Wiley & Sons, May 2012.
4. H. H. Bock. *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2000.
5. S. Chakraborty, K. Meel, and M. Vardi. A scalable approximate model counter. In *CP'2013)*, pages 200–216, 2013.
6. F. d. A. de Carvalho, M. Csernel, and Y. Lechevallier. Clustering constrained symbolic data. *Pattern Recognition Letters*, 30(11):1037–1045, 2009.
7. R. M. de Souza and F. d. A. De Carvalho. Clustering of interval data based on city–block distances. *Pattern Recognition Letters*, 25(3):353–365, 2004.
8. E. Diday and F. Esposito. An introduction to symbolic data analysis and the SODAS software. *Intell. Data Anal.*, 7(6):583–601, 2003.
9. C. P. Gomes, J. Hoffmann, A. Sabharwal, and B. Selman. From sampling to model counting. In *IJCAI'1997*, pages 2293–2299, 2007.
10. K. C. Gowda and E. Diday. *New Approaches in Classification and Data Analysis*, chapter Symbolic Clustering Algorithms using Similarity and Dissimilarity Measures, pages 414–422. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.
11. L. Hotz, A. Felfernig, M. Stumptner, A. Ryabokon, C. Bagley, and K. Wolter. Chapter 6 - configuration knowledge representation and reasoning. In *Knowledge-Based Configuration*, pages 41 – 72. Morgan Kaufmann, 2014.
12. S. Jabbour, J. Lonlac, L. Sais, and Y. Salhi. Extending modern SAT solvers for models enumeration. In *IEEE-IRI'2014*, pages 803–810, 2014.
13. P. Jaccard. The distribution of the flora of the alpine zon. *New Phytologist*, 11:37–50, 1912.
14. R. S. Michalski. Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts. *Journal of Policy Analysis and Information Systems*, 4(3):219–244, 1980.
15. A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.
16. A. Tversky. *Preference, Belief, and Similarity*. The MIT Press, November 2003.

On an Argument-centric Persuasion Framework

Y. Salhi. On an Argument-centric Persuasion Framework. International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2019. IFAAMAS, 1279-1287.

On an Argument-centric Persuasion Framework

Yakoub Salhi
CRIL - CNRS, Université d'Artois
Lens, France
salhi@cril.fr

ABSTRACT

In this paper, we propose an argument-centric persuasion framework. We first introduce a decision problem, called persuasion satisfiability, which is defined as the problem of determining whether there exists a sequence of arguments that starts from a given initial state, such as beliefs or wishes of the persuadee, and allows for achieving a given purpose of the persuader. This sequence should satisfy different constraints, including particularly upper bound constraints on the weight as well as on the length. We show that this decision problem is NP-complete and propose an encoding in partial weighted MaxSAT framework for solving it. Then, we show that the proposed encoding offers flexibility for dealing with different variants of the persuasion satisfiability problem. Finally, to avoid the explicit use of upper bound constraints on the weight and the length, we consider the notion of Pareto optimality by proposing an approach based on the use of partial weighted MaxSAT, which allows for finding non dominated (optimal) solutions.

KEYWORDS

Computational Persuasion; Argumentation; Knowledge Representation

ACM Reference Format:
Yakoub Salhi. 2019. On an Argument-centric Persuasion Framework. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

1 INTRODUCTION

Persuasion technologies aim at influencing users to make psychological and/or physical changes (thoughts, feelings, behaviors, motivation, etc) in several domains, such as healthcare, education, politics, marketing, etc (for interesting more complete definitions, see e.g. [12, 18]). One of the key approaches in the persuasion activity is the explicit use of convincing arguments [17, 18]. In this context, it is worth mentioning that in [15] the author has proposed interesting and reasonable key requirements for argument-centric persuasion in the particular case of behavior change. These requirements include those essential in this work, which are minimizing the effort involved on the part of the user and maintaining engagement by, for instance, avoiding long sequences of arguments.

In the literature, there are number of works which focus on the use of arguments in the persuasion activity. One can first mention the dialogical approach, where persuasion is defined as a dialogue between two agents trying to convince each other about an issue by exchanging arguments, that is, each agent plays both the role of persuader as well as that of persuadee (e.g. [2–4, 8, 21, 22]). In addition, in [16], the authors have proposed an asymmetric approach, where the word "asymmetric" refers to the fact that the persuadee agent is unable to posit arguments, but can accept and reject the arguments of the persuader agent. The main advantage of an asymmetric persuasion system is that it allows for avoiding natural language processing. As shown in this paper, this approach can be used in our argument-centric persuasion framework. Furthermore, there are works where the persuader system uses models and strategies for selecting the right sequence of arguments to convince the persuadee. To illustrate this point, in [13], the authors have proposed a decision-tree based framework for representing persuasion dialogues. In this general framework, the notion of decision tree is adapted to persuasion for selecting the argument to posit in the current state of dialogue. In the same vein, we propose in this work approaches that are based on the use of encodings in partial weighted MaxSAT framework for selecting the sequences of arguments that the persuader agent has to use. Let us note that other recent interesting studies on argument-centric persuasion have been proposed in [9, 14, 19, 20].

In this paper, we introduce an argument-centric persuasion framework. To do this, we consider the case where the persuasion activity consists in using a sequence of arguments following different approaches, in particular, that where the persuader agent shows that the target objective is a consequence of knowledge, beliefs and wishes of the persuadee agent. An argument in our framework is defined as an ordered pair of sets of literals, which can be seen as a simple intermediate approach between the logic-based one where the arguments are defined using logical formulas (e.g. see [1, 7]) and Dung's abstract one [10] where the internal structure is not considered at all. In this context, our main motivation is enhancing the expressivity of the abstract framework and avoiding at the same time important computational complexity issues, such as the fact that entailment in classical logic is coNP-complete. Moreover, to take into account particular requirements for argument-centric persuasion introduced in [15], we associate a weight to every argument to represent the effort needed on the part of the persuadee to integrate this argument, and we reason on the lengths of the argument sequences in order to prioritize shorter ones.

To formally define our framework, we introduce a decision problem, called persuasion satisfiability, which is defined as the problem of determining whether there exists a sequence of arguments that starts by a given set of literals, called the initial state, and allows for obtaining another given set of literals, called the objective state. This sequence should satisfy different constraints, in particular, upper bound constraints on the weight as well as on the length. The

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

initial state can be seen as the set of knowledge, beliefs and wishes of the persuadee agent, while the objective state as the purpose of the persuader agent. From the computational complexity point of view, we show that this decision problem is NP-complete. Note that we propose a simple approach for using the proposed framework in the context of a bidirectional persuasion by allowing the persuadee agent to accept and reject the arguments.

Then, we propose an encoding in partial weighted MaxSAT framework for solving the persuasion satisfiability problem. Using the fact that partial weighted MaxSAT is an optimization problem, our encoding allows for finding a solution of the smallest weight. Next, we show that the proposed partial weighted MaxSAT encoding offers flexibility for dealing with different variants of the persuasion satisfiability problem, such as allowing the persuader to hide parts of conclusions of the used arguments to avoid inconsistency, or considering a conflict graph between arguments.

As said before, we use in the persuasion satisfiability problem upper bound constraints on the length and on the weight. The choice of the bounds may be arbitrary without specific knowledge about the considered case, which can be seen as a drawback of our approach. Thus, to avoid the use of upper bounds constraints, we consider the notion of Pareto optimality. Indeed, we propose a method based on the use of partial weighted MaxSAT encodings that allows for finding solutions that are not dominated (optimal in Pareto sense). More precisely, a sequence of arguments is optimal if there is no sequence that has a smaller length without a greater weight, or a smaller weight without a greater length.

2 A PERSUASION FRAMEWORK

In this section, we introduce a simple argument-centric persuasion framework. We first describe the approches that we consider in the persuasion activity. Then, we formally introduce our framework and, in particular, the persuasion satisfiability problem. After that, we provide some computational complexity results. Finally, we describe a basic bidirectional approach within our framework.

2.1 Persuasion Approaches

In this work, we consider that the persuasion activity consists in using a sequence of arguments that allows the persuader agent to achieve a target objective by using one of the following three approaches:

- **Strong approach:** the persuader agent shows that the target objective is a consequence of knowledge, beliefs and wishes of the persuadee agent.
- **Weak approach:** the persuader agent shows that the target objective has as a consequence wishes of the persuadee agent. In other words, the persuadee may satisfy her/his purpose by accepting/doing the purpose of the persuader, but the latter is not a requirement to satisfy the persuadee purpose.
- **Mixed approach:** the persuader agent uses both the strong and the weak approaches for convincing the persuadee agent. For instance, the persuader can use the strong approach to achieve a part of the purpose and the weak approach to achieve the remaining part.

For instance, using the strong approach, the persuader agent can use the argument stating that the fact that the persuadee knows that a given product is efficient and not expensive has as a consequence that this product should be purchased: the purchase of the considered product by the persuadee agent is the purpose of the persuader agent and the fact that the considered product is efficient and not expensive are beliefs of the persuadee agent. In addition, using the weak approach, the persuader agent can use the argument stating that playing sports has as a consequence that the persuadee agent may develop friendships: playing sports is the purpose of the persuader agent and developing friendships is a wish of the persuadee agent. It is worth noting that the previous argument does not mean that playing sports is the unique way to develop friendships. In summary, in the strong approach the persuader starts with knowledge, beliefs and wishes of the persuadee, while in the weak approach the persuader starts with the target objective. Regarding the mixed approach, the persuader agent can first use the weak approach with the argument that stop smoking allows for avoiding dangerous carcinogens, and then the strong approach with the argument that avoiding dangerous carcinogens allows for avoiding dangerous diseases: the purpose of the persuader is convincing the persuadee to stop smoking, and the persuadee wishes to avoid any dangerous disease. As a side note, the mixed approach can be used by combining in different other manners the strong and weak approaches.

In this paper, we define the notion of argument as an ordered pair of sets of literals. In a sense, this can be seen as an abstraction of the standard logic-based argument structure (e.g. see [1, 7]). Indeed, a logic-based argument is defined as an ordered pair of a set of formulas representing the support and a formula representing the conclusion. Thus, instead of using logical formulas, we use here literals. Our aim in this context is to use a simple intermediate approach between the logic-based one and Dung's abstract one [10] where the internal structure is not considered at all.

It is noteworthy that our framework is inspired, in part, from the requirements for argument-centric persuasion introduced in [15], in particular, the requirements 3 and 6. Indeed, we associate a weight to each argument in order to represent the effort needed on the part of the persuadee to integrate this argument, and we reason on the lengths of the arguments sequences in order to prioritize shorter ones.

2.2 Framework Definition

We here define the notion of persuasion frame and some related notions. Then, we introduce a decision problem, called persuasion satisfiability, which is defined as the problem of determining whether there exists a sequence of arguments that starts by a given set of literals, called the initial state, and allows for obtaining another given set of literals, called the objective state.

First, let us recall that a *propositional variable* is a variable that can either be true or false. A *literal* is either a propositional variable or a negated propositional variable. As usual, we use the unary logical connective $\neg$ to denote the negation. Moreover, given a set of propositional variables V, we use $Lit(V)$ to denote the set of all the possible literals that are defined using the variables in V, i.e., $Lit(V) = V \cup \{\neg p \mid p \in V\}$.

Definition 2.1 (Persuasion Frame). A *persuasion frame* is a tuple $(\mathcal{S}, \mathcal{A}, W)$ where $\mathcal{S}$ is a non empty finite set of propositional variables representing abstract statements, $\mathcal{A}$ is a finite set of arguments built over $\mathcal{S}$ and W a mapping that associates a weight (an integer) to each argument in $\mathcal{A}$. An *argument* a over $\mathcal{S}$ is an ordered pair $\langle X, C\rangle$ where $X, C \subseteq Lit(\mathcal{S})$, $X \cup C$ is consistent, $X \cap C = \emptyset$ and $C \neq \emptyset$, X is called the *support* of a and C the *conclusion* of the latter.

Given an argument a, we use $Supp(a)$ and $Conc(a)$ to denote its support and its conclusion respectively.

The weight of an argument is used to represent the cost of the latter. In particular, the weight mapping can be used as a weakness measure of the arguments regarding the impact on the persuadee agent.

Example 2.2. Let us first consider the following set of statements $\mathcal{S}$:

- ad_i: the advertisement number i convinces the considered agent.
- h: the considered agent thinks that the product p is healthy.
- eff: the considered agent thinks that the product p is effective.
- exp: the considered agent thinks that the product p is expensive.
- $pack$: the considered agent likes the packaging of p.
- buy: the considered agent is convinced that the product p has to be purchased.

In this example, we consider the following set of arguments $\mathcal{A}$: $a_1 = \langle\{ad_1\}, \{h, eff\}\rangle$, $a_2 = \langle\{ad_2\}, \{\neg exp\}\rangle$, $a_3 = \langle\{ad_3\}, \{pack\}\rangle$, $a_4 = \langle\{h, eff, \neg exp, pack\}, \{buy\}\rangle$, $a_5 = \langle\{h, eff, pack\}, \{buy\}\rangle$ and $a_6 = \langle\{pack\}, \{buy\}\rangle$, where $\langle X, C\rangle$ means that the truth of the elements of X has as a consequence the truth of the elements of C. For instance, the argument a_6 means that if the persuadee agent likes the packaging of the product p, then she/he is convinced that she/he has to buy it. Clearly, the argument a_4 is stronger than a_5 which is stronger than a_6. To represent this fact, one can use the weight mapping by setting for instance $W(a_4) = 0$, $W(a_5) = 1$ and $W(a_6) = 2$.

A persuasion frame can be developed by using a crowdsourcing-based collaborative approach. For instance, an interesting approach may consist in allowing crowd members to propose arguments and also to vote on arguments as describe in Figure 1. The votes are then exploited for defining the weights of the arguments. For the sake of illustration, one can use the following simple vote-based mapping:

$$W(a) = \#NegVotes(a) - \#PosVotes(a)$$

where $\#NegVotes(a)$ and $\#PosVotes(a)$ corresponds respectively to the number of votes against and for the argument a. This function means that more votes for a and less votes against it reduce the weight.

Definition 2.3 (Argument Path). Given a persuasion frame $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$ and a set $I \subseteq Lit(\mathcal{S})$, an *$I$-path* in $\mathcal{F}$ is a sequence $s = a_1, \ldots, a_k$ of distinct arguments in $\mathcal{A}$ where, for all $i \in 1..k$, $Supp(a_i) \subseteq I \cup \bigcup_{1 \leq j < i} Conc(a_j)$.

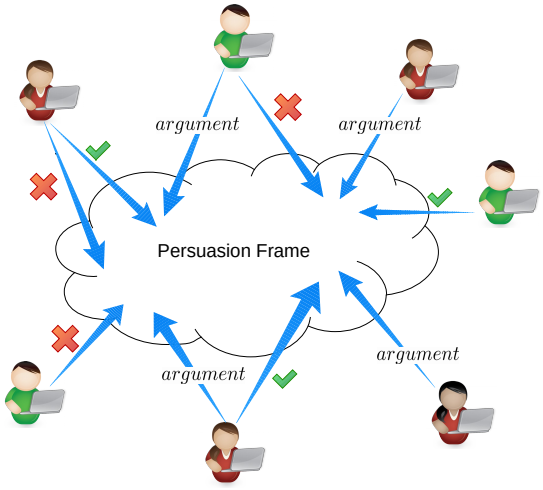


Figure 1: Building a persuasion frame

The condition on a sequence of arguments to be an I-path means only that we have to obtain the support of an argument before using it.

Given an I-path $s = a_1, \ldots, a_k$, we use $Arg(s)$, $\mathcal{L}(s)$, $\mathcal{W}(s)$ and $C(s)$ to denote respectively the set $\{a_1, \ldots, a_k\}$, the length of the sequence k, $\sum_{i=1}^{k} W(a_i)$ and $I \cup \bigcup_{i=1}^{k} Conc(a_i)$.

Definition 2.4 (Consistency). An I-path $a_1, \ldots, a_k$ is said to be *consistent* if the set of literals $I \cup \bigcup_{i=1}^{k} Conc(a_i)$ is consistent.

Roughly speaking, a consistent I-path is a sequence of arguments that can be used together starting from I and do not produce contradictory pieces of information. Consider again Example 2.2. The sequence a_1, a_2, a_3, a_4 is a consistent $\{ad_1, ad_2, ad_3\}$-path, but it is not a consistent $\{ad_1, ad_2, ad_3, exp\}$-path since a_2 produce $\neg exp$ which is in contradiction with exp in the initial state.

Now, we introduce the central decision problem studied in this work.

Definition 2.5 (Persuasion Satisfiability Problem). Given a persuasion frame $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$, a consistent set of literals $I \subseteq Lit(\mathcal{S})$, called the *initial state*, a non empty consistent set of literals $O \subseteq Lit(\mathcal{S})$, called the *objective state*, a length bound $k \in \mathbb{N} \cup \{\infty\}$ and a weight bound $v \in \mathbb{Z} \cup \{\infty\}$, the persuasion satisfiability problem is to check whether there exists a consistent I-path s in $\mathcal{F}$ such that $O \subseteq C(s)$, $\mathcal{L}(s) \leq k$ and $\mathcal{W}(s) \leq b$.

We denote every instance of the persuasion satisfiability problem as a tuple of the form $(\mathcal{F}, I, O, k, b)$. Further, note that ∞ is only used to formally represent the absence of a bound.

For example, if we only consider the strong approach, the initial state corresponds to knowledge, beliefs and wishes of the persudee agent and objective state to the purpose of the persuader agent. Conversely, if we only consider the weak approach, the initial state corresponds to the purpose of the persuader and the objective state to wishes of the persuadee.

Example 2.6. We here provide an example inspired from [9]. Let us consider the following statements: s_1 =being healthy, s_2 =stop smoking, s_3 =a long life, s_4 =good looking, s_5 =supporting family. Let $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$ be a persuasion frame such that $\mathcal{S} = \{s_1, s_2, s_3, s_4, s_5\}$, $\mathcal{A} = \{a_1 = \langle\{s_1\}, \{s_2\}\rangle, a_2 = \langle\{s_3\}, \{s_1\}\rangle, a_3 = \langle\{s_4\}, \{s_1\}\rangle, a_4 = \langle\{s_5\}, \{s_3\}\rangle, a_5 = \langle\{s_4\}, \{s_2\}\rangle\}$, and $W(a_1) = 3$, $W(a_2) = 2$, $W(a_3) = 2$, $W(a_4) = 1$ and $W(a_5) = 6$. In this context, we consider the instance of the persuasion satisfiability problem $P = (\mathcal{F}, \{s_5\}, \{s_2\}, 3, 6)$. Thus, using the strong approach, the aim in P is to convince the persuadee to stop smoking by using the fact that it is important for her/him to support her/his family. The $\{s_5\}$-path a_4, a_2, a_1 is a solution of P which corresponds to the sequence: supporting family $\rightarrow$ a long life $\rightarrow$ being healthy $\rightarrow$ stop smoking.

It is worth noticing that the support and the conclusion of an argument are not treated in our framework as an implication in classical logic. Consider for instance the two arguments $a = \langle\{p, \neg q\}, \{r\}\rangle$ and $a' = \langle\{q\}, \{r\}\rangle$. Clearly, a, a' and a', a are both not $\{p\}$-paths since the supports of a and a' are both not included in $\{p\}$ (see Definition 2.3). However, we have in classical logic $\{p\} \cup \{p \wedge \neg q \rightarrow r, q \rightarrow r\} \vdash r$. In fact, r is obtained from $\{p\} \cup \{p \wedge \neg q \rightarrow r, q \rightarrow r\}$ using the law of excluded middle $q \vee \neg q$ that is valid in classical logic. But in our framework, we need to know the disjunct in $q \vee \neg q$ that we have to take into account for choosing the appropriate argument to produce r: in the case where q is true, we use the argument a', otherwise, we use a. For example, consider that $p = \mathit{having\ time}$, $q = \mathit{there\ is\ an\ exam}$ and $r = \mathit{review\ lessons}$. In this case, the argument a can be used when the persuadee agent prefers reviewing her lessons if she has time to do it without the constraint of an exam, while a' can be used when she can be convinced by the constraint of an exam. In a sense, this example shows that the support and the conclusion of an argument are treated in our framework as a constructive implication. The word "constructive" is used to refer to the constructivism principles in mathematics (see e.g. [6, 24]). One of the main constructivism principles is the rejection of the law of excluded middle.

2.3 Computational Complexity

We here show that the persuasion satisfiability problem is NP-complete, even when we do not consider one of the bounds on the length and the weight. In order to show NP-hardness, we use the well-known NP-complete problem of Hamiltonian cycle.

Theorem 2.7. *The persuasion satisfiability problem is NP-Complete.*

Proof. Given an instance $P = (\mathcal{F}, I, O, k, b)$ with $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$ and an I-path s in $\mathcal{F}$, one can check that s is a solution of P in polynomial time. Indeed, we only have to check the properties $O \subseteq C(s)$, $\mathcal{L}(s) \leq k$ and $\mathcal{W}(s) \leq b$. Moreover, one can easily see that P is satisfiable iff there exists a solution bounded by the number of abstract statement ($|\mathcal{S}|$). Indeed, this comes from the fact that one can require w.l.o.g. that each used argument has to bring at least one additional piece of information in $Lit(\mathcal{S})$. As a consequence, the persuasion satisfiability problem is in NP. To show that the latter is NP-hard, we use the well-known NP-complete problem of Hamiltonian cycle, which consists in determining if an undirected graph contains a Hamiltonian cycle. Let us recall that a Hamiltonian path in a graph is a path that visits each vertex exactly once. A Hamiltonian cycle is a Hamiltonian path that is a cycle. Let us now define our reduction of the Hamiltonian cycle problem into the persuasion satisfiability problem. Let $G = (V, E)$ be an undirected graph. Then, we associate to G the instance $P_G = (\mathcal{F}, \{v_0\}, V \cup \{v_0'\}, |V|, \infty)$ where $v_0 \in V$, v_0' is a fresh vertex ($v_0' \notin V$) and $\mathcal{F} = (V, \mathcal{A} = (\bigcup_{\{v,v'\} \in E} \{\langle\{v\}, \{v'\}\rangle\}) \cup (\bigcup_{\{v,v_0\} \in E} \{\langle\{v\}, \{v_0'\}\rangle\}), W)$ such that $W(a) = 0$ for every $a \in \mathcal{A}$. Assume that G admits as a Hamiltonian cycle the following path $c = \{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}\{v_n, v_0\}$. Then, the sequence $s = \langle\{v_0\}, \{v_1\}\rangle, \langle\{v_1\}, \{v_2\}\rangle, \ldots, \langle\{v_{n-1}\}, \{v_n\}\rangle, \langle\{v_n\}, \{v_0'\}\rangle$ is a $\{v_0\}$-path of $\mathcal{F}$ with $C(s) = V \cup \{v_0\}$ since c is a Hamiltonian cycle. Using again the fact that c is a Hamiltonian cycle, we get $\mathcal{L}(s) \leq |V|$. Then, s is a solution of P_G. It is easy to use a similar approach to show that every solution of P_G can be transformed into a Hamiltonian cycle of G. □

Our proof of Theorem 2.7 shows that the persuasion satisfiability problem is NP-complete, even when we do not consider the upper bound on the weight. Similarly, by using the weights of the arguments to represent the path length, we can also show that the problem remains NP-complete even when we do not consider the upper bound on the length.

Proposition 2.8. *The restriction of the persuasion satisfiability problem to the instances of the form $(\mathcal{F}, I, O, k, \infty)$ is NP-complete. Further, the restriction of the persuasion satisfiability problem to the instances of the form $(\mathcal{F}, I, O, \infty, b)$ is also NP-complete.*

2.4 A Basic Bidirectional Approach

In our framework, we do not explicitly describe how the persuader agent enters into a dialogue with the persuadee agent. However, this does not mean that we only consider a unidirectional approach for persuasion. For instance, in the same way as the asymmetric persuasion framework introduced in [16], our framework can be used in the context of a bidirectional asymmetric persuasion approach by allowing the persuadee to only accept or reject every argument.

Consider the approach described in Figure 2. First, the persuader agent uses the solution $a_1 \rightarrow \cdots \rightarrow a_l$. The persuadee agent accepts only the subsequence $a_1 \rightarrow \cdots \rightarrow a_{i-1}$ and rejects the argument a_i. Then, the persuader agent recomputes a new solution without taking into account the argument $a_1, \ldots, a_i$, since $a_1, \ldots, a_{i-1}$ are already accepted and a_i is rejected. Using the fact that the persuader agent communicated already i arguments to the persuadee agent, the length upper bound is set to $k - i$. Further, the weight upper bound has to be reduced by the weights of the accepted arguments $a_1, \ldots, a_{i-1}$. More precisely, the new weight upper bound is equal to $b - \sum_{j=1}^{i-1} W(a_j)$. Note that $W_{|\mathcal{A}'}$ corresponds to the restriction of W to $\mathcal{A}'$

3 AN ENCODING IN PARTIAL WEIGHTED MAXSAT

In this section, we employ a declarative approach for solving the persuasion satisfiability problem. Indeed, we propose an encoding of this problem in partial weighted MaxSAT, which is a well-known optimization problem within the artificial intelligence community.

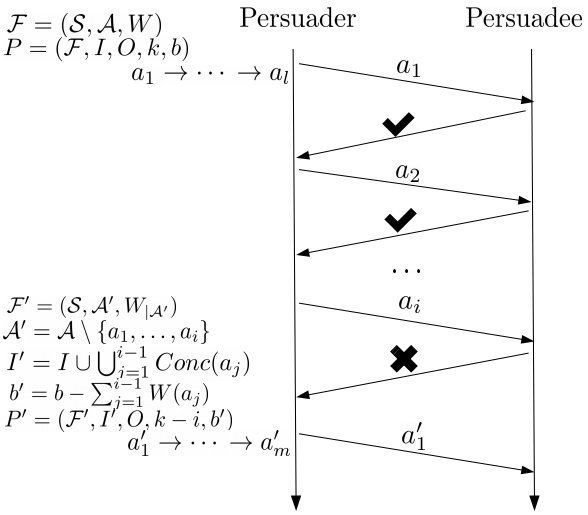


Figure 2: A basic bidirectional approach

In fact, our encoding allows for finding a solution of the smallest weight, since partial weighted MaxSAT is an optimization problem.

First, let us recall that a CNF formula is a conjunction of clauses where a *clause* is a disjunction of literals. It is well-known that every propositional formula can be translated to CNF w.r.t. the satisfiability problem using Tseitin's linear encoding [25]. A *Boolean interpretation* of a CNF formula ϕ is an assignment that associates truth values in $\{0, 1\}$ to propositional variables in $Var(\phi)$, where 0 stands for false and 1 stands for true. A Boolean interpretation is extended to CNF formulas as usual. A *model* of a CNF formula ϕ is a Boolean interpretation $\mathcal{B}$ satisfying this formula, i.e., $\mathcal{B}(\phi) = 1$. The problem of determining whether there exists a model that satisfies a given CNF formula, abbreviated as SAT, is one of the most studied NP-complete problems.

In partial weighted MAX-SAT, each clause is either relaxable (soft) with an associate cost or non-relaxable (hard). The objective is to find a Boolean interpretation that satisfies all the hard clauses and minimize the cost of the falsified soft clauses. Given a satisfiable partial weighted MAX-SAT instance ϕ and a solution $\mathcal{B}$ of ϕ, we use $cost(\mathcal{B})$ to denote the cost of $\mathcal{B}$.

Consider for instance the partial weighted MAX-SAT instance $\phi = \phi_h \wedge \phi_s$ where $\phi_h = (p \vee q) \wedge (p \vee r)$ is the hard part and $\phi_s = (3 : \neg p) \wedge (1 : \neg q) \wedge (1 : \neg r)$ is the soft part. Clearly, it is not possible to satisfy at the same time all the hard and soft clauses. The Boolean interpretation $\mathcal{B}$ defined by $\mathcal{B}(p) = 0$ and $\mathcal{B}(q) = \mathcal{B}(r) = 1$ is a solution of the instance ϕ. Indeed, the cost of $\mathcal{B}$ is equal to 2 and the costs of all the other models of the hard part ϕ_h are greater than or equal to 3.

Let $P = (\mathcal{F}, I, O, k, b)$ be an instance of the persuasion satisfiability problem, where $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$. In order to define our encoding for the instance P, we need some syntactic elements. We first associate to each literal $l \in Lit(\mathcal{S})$ a set of $k+1$ distinct propositional variables denoted $p_l^0, \ldots, p_l^k$. The propositional variable p_l^i is used to express whether or not the literal l is produced at the step number i. In particular, p_l^0 is true if and only if l occurs in the initial state I. Moreover, given a subset of literals $X \subseteq Lit(\mathcal{S})$ and $0 \leq i \leq k$, we use $R(X, i)$ to denote the set of variables $\{p_l^i \mid l \in X\}$. We also associate to each argument $a \in \mathcal{A}$ a distinct propositional variable denoted q_a and a set of k distinct propositional variables denoted $r_a^1, \ldots, r_a^k$. The propositional variable q_a is used to express whether or not the argument a is used in the solution, and similarly r_a^i is used to express whether or not the argument a is used in the solution at the step number i. Further, given a literal $l \in Lit(\mathcal{S})$, we use $Arg(l, \mathcal{A})$ to denote the set of the arguments $\{\langle X, C\rangle \in \mathcal{A} \mid l \in C\}$. We generalize this notation to the sets of literals as follows: $Arg(X, \mathcal{A}) = \{\langle X, C\rangle \in \mathcal{A} \mid X \cap C \neq \emptyset\}$.

In the following, we describe our partial weighted MaxSAT encoding. We first define the hard part of this encoding. In this context, the following formula expresses that I corresponds to the initial state:

$$\bigwedge_{l \in I} p_l^0 \wedge \bigwedge_{l' \in Lit(\mathcal{S}) \setminus I} \neg p_{l'}^0 \tag{1}$$

Then, we introduce a formula that allows us to require a consistent I-path:

$$\bigwedge_{e \in \mathcal{S}} \bigwedge_{i=0}^{k} \bigwedge_{j=0}^{k} (\neg p_e^i \vee \neg p_{\neg e}^j) \tag{2}$$

The following formula is used to relate the truth of every variable of the form p_l^i to the use of at least one argument containing l in its conclusion at the step i:

$$\bigwedge_{l \in Lit(\mathcal{S})} \bigwedge_{i=1}^{k} (p_l^i \rightarrow \bigvee_{a \in Arg(l, \mathcal{A})} r_a^i) \tag{3}$$

The next formula expresses that the use of an argument at the step i requires the truth of its support before this step:

$$\bigwedge_{a = \langle X, C\rangle \in \mathcal{A}} \bigwedge_{i=1}^{k} (r_a^i \rightarrow (\bigwedge_{l \in X} (\bigvee_{j=0}^{i-1} p_l^i))) \tag{4}$$

Then, we propose a formula that states that if an argument is used at the step i, then the literals of its conclusion are true at this step:

$$\bigwedge_{a = \langle X, C\rangle \in \mathcal{A}} \bigwedge_{i=1}^{k} (r_a^i \rightarrow (\bigwedge R(C, i))) \tag{5}$$

where $\bigwedge R(C, i)$ represents the conjunction of the literals occurring in $R(C, i)$.

We now introduce the formula expressing the fact that each argument is used at most once:

$$\bigwedge_{a \in \mathcal{A}} \sum_{i=1}^{k} r_a^i \leq 1 \tag{6}$$

Similarly, the following formula expresses the fact that in each step we use at most one argument:

$$\bigwedge_{i=1}^{k} \sum_{a \in \mathcal{A}} r_a^i \leq 1 \tag{7}$$

Essentially, the next conjunction of clauses states that we have to obtain the objective state:

$$(\bigwedge_{l \in O \setminus I} (\bigvee_{i=1}^{k} p_l^i)) \tag{8}$$

The following formula is only used to associate the truth of q_a to the use of the argument a:

$$\bigwedge_{a \in \mathcal{A}} ((\bigvee_{i=1}^{k} r_a^i) \leftrightarrow q_a) \tag{9}$$

The relaxable part contains only the following soft clauses:

$$W(a) : \neg q_a \ for \ every \ a \in \mathcal{A} \tag{10}$$

Every soft clause allows for associating each argument to its weight.

The formulas (6) and (7) involve the well-known at-most-one constraint. There are several linear encodings of this constraint as CNF formulas w.r.t. the propositional satisfiability problem (e.g. see [23]). Furthermore, for the sake of clarity, we do not use the conjunctive normal form for describing the hard part of our encoding. However, it is easy to transform the hard part into an equisatisfiable CNF formula by using Tseitin's linear encoding [25]. The main idea of this encoding consists in using De Morgan's laws with other valid equivalence rules, and associating fresh propositional variables to subformulas. Consider for instance the formula (4). We first associate a fresh variable s_l^i for every subformula of the form $\bigvee_{j=0}^{i-1} p_l^i$. Then, we replace (4) with the conjunction of clauses $\bigwedge_{a=\langle X,C\rangle \in \mathcal{A}} \bigwedge_{i=1}^{k} \bigwedge_{l \in X} (\neg r_a^i \vee s_l^i)$ and conjunctively add implications of the form $s_X^i \rightarrow \bigvee_{j=0}^{i-1} p_l^i$, which can be easily replaced by simple clauses. It is worth mentioning that we do not need to use the equivalence connective for relating the fresh variables to their associated subformulas, since the implication connective is sufficient to preserve satisfiability.

From now on, we use $\mathcal{E}(P)$ to denote the encoding that corresponds to the partial weighted MaxSAT instance (1)$\wedge \cdots \wedge$(10).

Proposition 3.1 (Soundness). *Let $P = (\mathcal{F}, I, O, k, b)$ be an instance of the persuasion satisfiability problem. Then, P admits a solution iff $\mathcal{E}(P)$ admits a solution $\mathcal{B}$ s.t. $cost(\mathcal{B}) \leq b$.*

Proof.

Part $\Rightarrow$. Assume that P admits as a solution the I-path $s = a_1, \ldots, a_m$. Then, we associate to s a Boolean interpretation $\mathcal{B}$ as follows. First, for all $1 \leq i \leq m$, $\mathcal{B}(q_{a_i}) = 1$, and for all $a \notin \{a_1, \ldots, a_m\}$, $\mathcal{B}(q_a) = 0$. Then, for all $1 \leq i \leq m$, $\mathcal{B}(r_{a_i}^i) = 1$, and for all $r_a^j \notin \{r_{a_i}^i \mid 1 \leq i \leq m\}$, $\mathcal{B}(r_a^j) = 0$. Clearly, the previous definitions implies that $\mathcal{B}$ satisfies the formulas (6), (7) and (9). The following property on $\mathcal{B}$ shows that the latter satisfies (1): for all $l \in Lit(\mathcal{S})$, if $l \in I$ then $\mathcal{B}(p_l^0) = 1$, otherwise $\mathcal{B}(p_l^0) = 0$. Further, we define the truth values of the variables of the form p_l^i for $i \in 1..k$ by using the following property: $\mathcal{B}(p_l^i) = 1$ iff $l \in Conc(a_i)$. Using the previous property and the fact that s is an I-path, we know that $\mathcal{B}$ satisfies the formulas (3), (4) and (5). Moreover, using the fact that s is consistent, $\mathcal{B}$ satisfies also the formula (2). Then, using the property $O \subseteq C(s)$, $\mathcal{B}$ satisfies (8). Finally, using the fact $\mathcal{W}(s) \leq b$, $cost(\mathcal{B}) \leq b$ holds. As a consequence, the partial MaxSAT instance $\mathcal{E}(P)$ admits a solution $\mathcal{B}'$ s.t. $cost(\mathcal{B}') \leq b$.

Part $\Leftarrow$. Assume that $\mathcal{E}(P)$ admits a solution $\mathcal{B}$ s.t. $cost(\mathcal{B}) \leq b$. Using the formula (7), we know that there is at most one true variable of the form r_a^i for every $1 \leq i \leq k$. Then, there is a unique sequence of argument $s = a_{i_1}, \ldots, a_{i_m}$ s.t. $i_n < i_{n'}$ for every $n < n'$, $\mathcal{B}(r_{a_{i_j}}^j) = 1$ for every $1 \leq j \leq m$, and $\mathcal{B}(r_a^{j'}) = 0$ for every $r_a^{j'} \notin \{r_{a_{i_j}}^j \mid 1 \leq j \leq m\}$. Further, using the formula (6), we know that the arguments in s are pairwise distinct. Then, using the formulas (1), (3), (4) and (5), $Supp(a_{i_j}) \subseteq I \cup \bigcup_{1 \leq j' < j} Conc(a_{i_{j'}})$ holds for every $1 \leq j \leq m$. As a consequence, s is an I-path. Further, using the formula (2), we know that s is consistent. The formula (8) allows us to obtain $O \subseteq C(s)$. Using the formula (9) and the soft part (10), $cost(\mathcal{B}) = \mathcal{W}(s)$ holds. Thus, knowing that $cost(\mathcal{B}) \leq b$, we obtain $\mathcal{W}(s) \leq b$. Therefore, s is a solution of P. □

An improvement of our encoding can be accomplished by removing the formula (6). Indeed, this formula is used to express that each argument is applied at most once, but one can easily see that it is not problematic to have arguments that occur more than once in a solution respecting the considered upper-bounds. In fact, we only need to keep the first application of each argument to obtain a solution where every argument occurs at most once.

4 WEAK CONSISTENCY AND FLEXIBILITY

In this section, we introduce a property on paths, called weak consistency, that formalizes the fact that the persuder agent may hide parts of conclusions of used arguments in order to present a consistent path.

To illustrate our motivation, consider a persuadee agent that has the following set of wishes $I = \{being\ healthy, \neg\ playing\ sports\}$, which means that this agent wants to be healthy without playing sports. Moreover, consider a persuader agent that has only the argument $a = \langle\{being\ healthy\}, \{stop\ smoking, playing\ sports\}\rangle$, that is, being healthy requires to stop smoking and to play sports. Clearly, the I-path a is not consistent. However, in certain cases, it would be interesting to hide literals in the conclusions of the considered arguments, as for instance, hiding the literal *playing sports* in the case of the previous argument. To this end, we introduce here the notion of weak inconsistency, which states that we only have to inspect the consistency of the supports of the considered arguments in a path.

Definition 4.1 (Weak Consistency). An I-path $a_1, \ldots, a_k$ is said to be *weak consistent* if the set of literals $I \cup \bigcup_{i=1}^{k} Supp(a_i)$ is consistent.

In the following proposition, we formally show that weak consistency is a weaker version of consistency.

Proposition 4.2. *Given a set of literals I, if an I-path is consistent, then it is also weak consistent.*

Proof. Let $s = a_1, \ldots, a_k$ be an I-path. Assume that s is not weak consistent. Then, the set of literals $I \cup \bigcup_{i=1}^{k} Supp(a_i)$ is not consistent. Further, using the definition of an I-path, we know that $Supp(a_i) \subseteq I \cup \bigcup_{1 \leq j < i} Conc(a_j)$ for every $i \in 1..k$. Thus, $I \cup \bigcup_{i=1}^{k} Supp(a_i) \subseteq I \cup \bigcup_{i=1}^{k} Conc(a_i)$ holds. As a consequence,

$I \cup \bigcup_{i=1}^{k} Conc(a_i)$ is not consistent and we deduce that s is not consistent. □

Consider for instance the persuasion frame $\mathcal{F} = (\{a, b, c, d\}, \{a_1 = \langle\{a\}, \{b, c\}\rangle, a_2 = \langle\{d\}, \{\neg c\}\rangle\}, W)$ where $W(a_1) = 1$ and $W(a_2) = 1$, and the instance of the persuasion satisfiability problem $P = (\mathcal{F}, \{a, d\}, \{b, \neg c\}, 2, 2)$. Clearly, there is no consistent $\{a, d\}$-path that satisfies P. However, a_1, a_2 is a weak consistent $\{a, d\}$-path that allows for obtaining $\{b, \neg c\}$ from $\{a, b\}$.

In order to use a partial weighted MaxSAT encoding for solving the persuasion satisfiability problem where we use weak consistency instead of consistency, we just need to remove the formula (5) from the encoding described in Section 3. Indeed, the formula (5) is used to impose the satisfaction of the propositional variables representing the conclusion of the applied arguments. It is noteworthy that the satisfaction of the variables representing the supports of the applied arguments comes particularly from the formulas (2) and (4).

In general, it would be interesting to allow the persuader agent to select the literals that can be hided to avoid inconsistency. For instance, the persuader agent may decide to hide only the negations of the literals that occur in the objective state and/or the initial state. To adapt our partial weighted MaxSAT encoding, we only have to use the following formula instead of (5):

$$\bigwedge_{a=\langle X,C\rangle \in \mathcal{A}} \bigwedge_{i=1}^{k} (r_a^i \rightarrow (\bigwedge R(C \setminus T, i))) \tag{11}$$

where T is the set containing the literals that can be hided. This formula states that if an argument is used at the step i, then the literals of its conclusion are true at this step except the literals that can be hided.

The previous variant show the flexibility of the proposed framework and our solution based on the use of encodings in partial weighted MaxSAT. In this context, one can easily define other variants to take into account other aspects. In order to better illustrate this point, we introduce a variant where the persuader agent considers conflicts between the available arguments. More precisely, given a persuasion frame $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$, we assume that the persuader agent has a conflict graph $G = (\mathcal{A}, E)$, which is an undirected graph where the set of vertices is $\mathcal{A}$, and having an edge $\{a, a'\}$ in E means that the arguments a and a' cannot be used together in any solution. For this variant, our encoding can be adapted by adding the following formula to the hard part:

$$\bigwedge_{\{a,a'\} \in E} (\sum_{i=1}^{k} r_a^i) + (\sum_{j=1}^{k} r_{a'}^j) \leq 1 \tag{12}$$

Indeed, assume w.l.o.g. that there exists $i \in 1..k$ such that r_a^i is assigned to 1. Then, $\sum_{i=1}^{k} r_a^i = 1$ holds, and consequently, we obtain $\sum_{j=1}^{k} r_{a'}^j = 0$, which means that a' is not used in the found solution.

Several other variants can be defined by reasoning on different other aspects, such as the opening argument, the last argument, etc.

5 PARETO OPTIMALITY

In the persuasion satisfiability problem, there are explicite upper bounds on the length and on the weight of the solution. The choice of these bounds may be arbitrary without specific knowledge about the considered case, which can be seen as a real drawback of our approach. Thus, to avoid the use of bounds, we here consider the notion of Pareto optimality.

We define a *PO-instance* as a triple of the form $P = (\mathcal{F}, I, O)$ where $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$ is a persuasion frame, $I \subseteq Lit(\mathcal{S})$ a consistent set of literals, called the *initial state*, and $O \subseteq Lit(\mathcal{S})$ a non empty consistent set of literals, called the *objective state*. Clearly, a PO-instance can be seen as an instance of the persuasion satisfiability problem without the upper bounds on the length and on the weight of the solution. From this angle, we say that a sequence of arguments s is a solution of PO-instance $P = (\mathcal{F}, I, O)$ if it is a solution of the instance of the persuasion satisfiability problem $P = (\mathcal{F}, I, O, \infty, \infty)$.

Definition 5.1 (Pareto-Optimal Solution). A *Pareto-optimal solution* of a PO-instance P is a solution s of P where, for all solution s' of P, (*i*) if $\mathcal{L}(s') < \mathcal{L}(s)$ then $\mathcal{W}(s) < \mathcal{W}(s')$, and (*ii*) if $\mathcal{W}(s') < \mathcal{W}(s)$ then $\mathcal{L}(s) < \mathcal{L}(s')$.

Given two solutions s and s' of a PO-instance, we say that s dominates s' if at least one the following properties is satisfied:

- $\mathcal{L}(s) < \mathcal{L}(s')$ and $\mathcal{W}(s) \leq \mathcal{W}(s')$;
- $\mathcal{W}(s) < \mathcal{W}(s')$ and $\mathcal{L}(s) \leq \mathcal{L}(s')$.

In other words, s dominates s' if s is better than s' on one criterion (length or weight) and s is not worse than s' on the other criterion. Note that a solution is Pareto-optimal if and only if it is not dominated by any other solution.

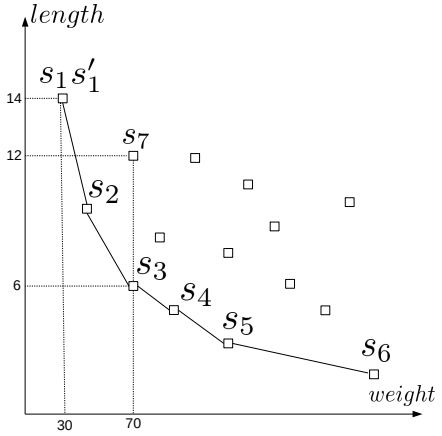


Figure 3: Pareto Front

For instance, in Figure 3, we show all the solutions of a given PO-instance. The x-axis and y-axis represent respectively the weights and the lengths of these solutions. It is important to note that a point can represent more than one solution (e.g. the point that represents the solutions s_1 and s'_1), since there may exist several solutions with

both the same length and the same weight. The solutions s_1, s_1' and s_3 are Pareto-optimal, but s_7 is not because it is dominated by s_3 ($\mathcal{L}(s_3) = 6 < \mathcal{L}(s_7) = 12$ and $\mathcal{W}(s_3) = \mathcal{W}(s_7) = 70$). Moreover, s_1 does not dominate s_3 because of $\mathcal{L}(s_3) = 6 < \mathcal{L}(s_1) = 14$, and s_3 does not dominate s_1 because of $\mathcal{W}(s_1) = 30 < \mathcal{W}(s_3) = 70$.

We call *Pareto frontier* of a given PO-instance the set of all its optimal solutions. For instance, the Pareto frontier in the example described in Figure 3 is the set $\{s_1, s_1', s_2, s_3, s_4, s_5, s_6\}$.

Our partial weighted MaxSAT encoding for solving the persuasion satisfiability problem described in Section 3 allows for computing a solution that satisfies the length bound constraint and has the smallest weight. More precisely, given an instance $P = (\mathcal{F}, I, O, k, b)$ of the persuasion satisfiability problem, every solution of the encoding $\mathcal{E}(P)$ corresponds to a solution s of P where $\mathcal{L}(s) \leq k$ and $\mathcal{W}(s) \leq \mathcal{W}(s')$ for every solution s' of P of length smaller than or equal to k. Thus, given a solution s of P which is obtained using $\mathcal{E}(P)$, we know that there exists a Pareto-optimal s' of the PO-instance $(\mathcal{F}, I, O)$ such that $\mathcal{W}(s') = \mathcal{W}(s)$.

Proposition 5.2. *Given a PO-instance $P = (\mathcal{F}, I, O)$ with $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$, the length of every Pareto-optimal solution is smaller than or equal to $min(|\mathcal{S}| - |I|, |\mathcal{A}|)$, where min stands for the minimum.*

Proof. Let $s = a_1, \ldots, a_n$ be a Pareto-optimal solution of P. It is trivial that $n \leq |\mathcal{A}|$ since the arguments used in s are pairwise distinct and belong to $\mathcal{A}$. Furthermore, we have $I \cup \bigcup_{i=1}^{j} Conc(a_i) \subset I \cup \bigcup_{i=1}^{j+1} Conc(a_i)$ for every $j \in 1..(n-1)$. Indeed, if there exists $j \in 1..(n-1)$, $I \cup \bigcup_{i=1}^{j} Conc(a_i) = I \cup \bigcup_{i=1}^{j+1} Conc(a_i)$, then the use of a_{j+1} is useless and we get a contradiction since s is Pareto-optimal. Thus, using the fact that $|I \cup \bigcup_{i=1}^{n} Conc(a_i)| \leq |\mathcal{S}|$, $n \leq |\mathcal{S}| - |I|$ holds. □

Given a PO-instance $P = (\mathcal{F}, I, O)$ with $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$, we use $LB(P)$ to denote the value $min(|\mathcal{S}| - |I|, |\mathcal{A}|)$. Furthermore, using Proposition 5.2, one can easily obtain a weight upper bound. Indeed, this bound can be defined as the sum of the weights of the arguments occurring in a set of $LB(P)$ greatest weights. More precisely, the weight of every Pareto-optimal solution of P is smaller than or equal to $\sum_{a \in \mathcal{A}'} W(a)$, where $\mathcal{A}' \subseteq \mathcal{A}$, $|\mathcal{A}'| = LB(P)$ and, for all $a' \in \mathcal{A} \setminus \mathcal{A}'$, $W(a') \leq min\{W(a) \mid a \in \mathcal{A}'\}$. We use $WB(P)$ to denote the bound $\sum_{a \in \mathcal{A}'} W(a)$.

In order to introduce our approach for computing certain Pareto-optimal solutions, we propose a partial weighted MaxSAT encoding that allows for computing a solution of a Given PO-instance with the smallest length. Let $P = (\mathcal{F}, I, O)$ be a PO-instance with $\mathcal{F} = (\mathcal{S}, \mathcal{A}, W)$. We use $\mathcal{E}_{length}(P)$ to denote the encoding that allows for computing a solution of smallest length. The hard part of this encoding is exactly the same as the hard part of $\mathcal{E}(P')$ with $P' = (\mathcal{F}, I, O, LB(P), \infty)$. The relaxable part is defined as follows:

$$1 : \neg q_a \ for\ every\ a \in \mathcal{A} \tag{13}$$

The soundness of $\mathcal{E}_{length}(P)$ can be obtained in the same way as the soundness of the encoding of the persuasion satisfiability problem. Indeed, knowing that the hard part is the same as that of $\mathcal{E}(P')$, we know that every solution of $\mathcal{E}_{length}(P)$ corresponds to a solution of P. Moreover, using Proposition 5.2, the hard part of $\mathcal{E}_{length}(P)$ is satisfiable if and only if P admits a solution. In addition, one can easily see that the relaxable part allows clearly for reducing the number of used arguments.

A simple approach for finding a Pareto-optimal solution of a given PO-instance P can be defined by solving two partial weighted MaxSAT encodings. Indeed, we first compute a solution s_0 using the encoding $\mathcal{E}_{length}(P)$. Then, every solution of $\mathcal{E}(P')$ with $P' = (\mathcal{F}, I, O, \mathcal{L}(s_0), \infty)$ is a Pareto-optimal solution of P. Indeed, let s be a solution of P obtained from $\mathcal{E}(P')$. Then, for every solution s' with $\mathcal{L}(s') \leq \mathcal{L}(s_0)$, $\mathcal{W}(s') \leq \mathcal{W}(s)$ holds. Moreover, we know that $\mathcal{L}(s_0)$ is the smallest length of the solutions of P. As a consequence, s is not dominated by any other solution of P.

A similar approach for finding a Pareto-optimal solution can be defined by computing first the smallest weight that can be obtained from our encoding $\mathcal{E}(P)$, and then, we use an encoding that allows for finding one of the shortest paths with respect to the previous weight. Indeed, let $P = (\mathcal{F}, I, O)$ be a PO-instance. First, a solution s_0 for $\mathcal{E}(P')$ is computed where $P' = (\mathcal{F}, I, O, LB(P), \infty)$. Using Proposition 5.2, we know that $\mathcal{W}(s_0)$ is the smallest weight of the solutions of P. Then, we use a partial weighted MaxSAT encoding $\mathcal{E}_{weight}(P, W(s_0))$ to find a Pareto-optimal solution. The encoding $\mathcal{E}_{weight}(P, \mathcal{W}(s_0))$ is obtained by only adding the following hard pseudo-Boolean constraint to $\mathcal{E}_{length}(P)$:

$$\sum_{a \in \mathcal{A}} W(a) * q_a \leq \mathcal{W}(s_0) \tag{14}$$

Clearly, this constraint allows for guaranteeing that every solution of $\mathcal{E}_{weight}(P, \mathcal{W}(s_0))$ has the smallest weight, i.e., it is also a solution of $\mathcal{E}(P')$. Moreover, for every solution s of $\mathcal{E}_{weight}(P, \mathcal{W}(s_0))$, there is no solution of P that has the weight $\mathcal{W}(s_0)$ and is also shorter than s. As a consequence, every solution of the encoding $\mathcal{E}_{weight}(P, \mathcal{W}(s_0))$ is Pareto-optimal.

It is worth mentioning that there are several efficient polynomial encodings of the pseudo-Boolean constraints as CNF formulas in the literature (e.g. see [5, 11]).

6 CONCLUSION AND PERSPECTIVES

In this paper, we have presented an argument-centric persuasion framework. This contribution proposes a decision problem, called persuasion satisfiability, that allows for dealing with computational persuasion in a simple and intuitive way. From the computational complexity point of view, we showed that this decision problem is NP-complete. We have proposed an encoding in partial weighted MaxSAT for solving this problem. We also showed the flexibility of our framework and the approach based on the use of partial weighted MaxSAT. Finally, in order to avoid the use of explicit upper bound constraints, which can be seen as a drawback of our framework in the absence of specific knowledge about the considered case, we have proposed an approach that allows for finding optimal solutions in Pareto sense.

In our future work, we intend first to improve the proposed framework following two directions: (1) allowing the persuadee to use different kinds of counterarguments; and (2) defining an updating method for persuasion frames to take into account the responses of the persuadee. We also plan to implement the proposed solving methods based on partial weighted MaxSAT to provide an experimental study on the use of the proposed framework.

REFERENCES

[1] L. Amgoud and P. Besnard. 2013. Logical limits of abstract argumentation frameworks. *Journal of Applied Non-Classical Logics* 23, 3 (2013), 229–267.

[2] L. Amgoud and F. Dupin de Saint-Cyr. 2013. An Axiomatic Approach for Persuasion Dialogs. In *25th IEEE International Conference on Tools with Artificial Intelligence, ICTAI*. IEEE Computer Society, Herndon, VA, USA, 618–625.

[3] L. Amgoud, N. Maudet, and S. Parsons. 2000. Modeling Dialogues Using Argumentation. In *4th International Conference on Multi-Agent Systems, ICMAS*. IEEE Computer Society, Boston, MA, USA, 31–38.

[4] L. Amgoud, S. Parsons, and N. Maudet. 2000. Arguments, Dialogue, and Negotiation. In *ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence*. IOS Press, Berlin, Germany, 338–342.

[5] O. Bailleux, Y. Boufkhad, and Olivier Roussel. 2009. New Encodings of Pseudo-Boolean Constraints into CNF. In *Theory and Applications of Satisfiability Testing - SAT, 12th International Conference, SAT. Proceedings (Lecture Notes in Computer Science)*, Vol. 5584. Springer, Swansea, UK, 181–194.

[6] M.J. Beeson. 1985. *Foundations of constructive mathematics: metamathematical studies*. Springer-Verlag.

[7] P. Besnard and A. Hunter. 2008. *Elements of Argumentation*. MIT Press.

[8] E. Bonzon and N. Maudet. 2011. On the outcomes of multiparty persuasion. In *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Volume 1-3*. IFAAMAS, Taipei, Taiwan, 47–54.

[9] L. A. Chalaguine, E. Hadoux, F. Hamilton, A. Hayward, A. Hunter, S. Polberg, and H. W. W. Potts. 2018. Domain Modelling in Computational Persuasion for Behaviour Change in Healthcare. *CoRR* (2018).

[10] P. M. Dung. 1995. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77 (1995), 321–357.

[11] N. Eén and N. Sörensson. 2006. Translating Pseudo-Boolean Constraints into SAT. *JSAT* 2, 1-4 (2006), 1–26.

[12] B. J. Fogg. 1998. Persuasive Computers: Perspectives and Research Directions. In *Proceeding of the CHI '98 Conference on Human Factors in Computing Systems*. ACM, Los Angeles, California, USA, 225–232.

[13] E. Hadoux and A. Hunter. 2017. Strategic Sequences of Arguments for Persuasion Using Decision Trees. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI Press, San Francisco, California, USA, 1128–1134.

[14] E. Hadoux and A. Hunter. 2018. Learning and Updating User Models for Subpopulations in Persuasive Argumentation Using Beta Distributions. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, Stockholm, Sweden, 1141–1149.

[15] A. Hunter. 2014. Opportunities for Argument-Centric Persuasion in Behaviour Change. In *Logics in Artificial Intelligence - 14th European Conference, JELIA. Proceedings*. Springer, Funchal, Madeira, Portugal, 48–61.

[16] A. Hunter. 2015. Modelling the Persuadee in Asymmetric Argumentation Dialogues for Persuasion. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*. AAAI Press, Buenos Aires, Argentina, 3055–3061.

[17] A. Hunter. 2016. Computational Persuasion with Applications in Behaviour Change. In *Computational Models of Argument - Proceedings of COMMA*. IOS Press, Potsdam, Germany, 5–18.

[18] A. Hunter. 2018. Towards a framework for computational persuasion with applications in behaviour change. *Argument & Computation* 9, 1 (2018), 15–40.

[19] A. Hunter and S. Polberg. 2017. Empirical Methods for Modelling Persuadees in Dialogical Argumentation. In *29th IEEE International Conference on Tools with Artificial Intelligence, ICTAI*. IEEE Computer Society, Boston, MA, USA, 382–389.

[20] A. Hunter and N. Potyka. 2017. Updating Probabilistic Epistemic States in Persuasion Dialogues. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 14th European Conference, ECSQARU, Proceedings (Lecture Notes in Computer Science)*, Vol. 10369. Springer, Lugano, Switzerland, 46–56.

[21] A. Hunter and M. Thimm. 2016. Optimization of dialectical outcomes in dialogical argumentation. *International Journal of Approximate Reasoning* 78 (2016), 73–102.

[22] H. Prakken. 2005. Coherence and Flexibility in Dialogue Games for Argumentation. *Journal of Logic and Computation* 15, 6 (2005), 1009–1040.

[23] Carsten Sinz. 2005. Towards an Optimal CNF Encoding of Boolean Cardinality Constraints. In *Principles and Practice of Constraint Programming - CP 2005, 11th International Conference, CP, Proceedings*. Springer, Sitges, Spain, 827–831.

[24] A.S. Troelstra and D. van Dalen. 1988. *Constructivism in Mathematics: an Introduction, Vol. I*. Studies in Logic and the Foundations of Mathematics, Vol. 121. North Holland Publishing Company.

[25] G. S. Tseitin. 1968. On the complexity of derivations in the propositional calculus. *Studies in Mathematics and Mathematical Logic* Part II (1968), 115–125.

On an MCS-based inconsistency measure

M. Ammoura, Y. Salhi, B. Oukacha and B. Raddaoui. On an MCS-based inconsistency measure. International Journal of Approximate Reasoning 80. 443-459 (2017).

On an MCS-based Inconsistency Measure

Meriem Ammoura[a,b], Yakoub Salhi[a], Brahim Oukacha[b], Badran Raddaoui[a]

[a] *CRIL - CNRS UMR 8188*
Univ. Artois, F-62307 Lens Cedex, France
[b] *LAROMAD - Univ. Mouloud Mammeri*
Tizi Ouzou, Algeria

Abstract

An important problem in knowledge-based systems is inconsistency handling. This problem has recently been attracting a lot of attention in AI community. In this paper, we tackle the problem of evaluating the amount of conflicts in knowledge bases, and provide a new fine grained inconsistency measure, denoted I_{MCC}, based on maximal consistent sets (MCSes). The main idea consists in quantifying the inconsistency of a knowledge base by considering that all its consistent pieces of information are possible. Furthermore, we provide an epistemic interpretation of our inconsistency measure using the multimodal logic $\mathsf{S5}$. Then, we show that I_{MCC} satisfies several state-of-the-art postulates. Moreover, we provide an encoding in integer linear programming for computing our inconsistency measure, which is defined from the set of MCSes. We also propose a Partial Max-SAT encoding, which allows us to avoid the computation of the MCSes. Finally, we provide a comparison between I_{MCC} and two related existing inconsistency measures.

Keywords: Knowledge Base, Inconsistency Measure, Maximal Consistent Subset

1. Introduction

In classical logics, the principle of explosion is a law which states that any formula can be deduced from a contradiction using the inference process. This principle means that the inference process alone in classical logic

Email addresses: meriem.am21@gmail.com (Meriem Ammoura), salhi@cril.fr (Yakoub Salhi), oukachabrahim@yahoo.fr (Brahim Oukacha), raddaoui@cril.fr (Badran Raddaoui)

does not allow to reason under inconsistency. To remedy this problem, several approaches have been proposed in the literature, such as argumentation theory, paraconsistent logics, belief revision, etc. The main goal of these approaches is to deal with inconsistency as an informative concept. In the same vein, inconsistency measures have been introduced in order to be used in analyzing inconsistency. In the literature, an inconsistency measure is defined as a function that associates a non negative value to each knowledge base [1]. Several inconsistency measures have been proposed in the literature (e.g. [2, 3, 4, 5, 1, 6, 7, 8]), and it has been shown that they are suitable for various applications such as e-commerce protocols [9], software specifications [10], belief merging [5], news reports [11], requirements engineering [1], integrity constraints [12], databases [13], ontologies [14], semantic web [14], network intrusion detection [15], and multi-agent systems [8].

In [1], Hunter and Konieczny have proposed four axiomatic properties that any inconsistency measure should satisfy, namely Consistency, Monotonicity, Free-Formula-Independence and Dominance. However, in a recent article [16], Besnard has provided objections to Free-Formula-Independence and Dominance. Indeed, the author has pointed out in his article undesirable consequences of these properties, such as ignoring certain conflicts, and has provided alternative properties in order to avoid these consequences. It is worth noting that Hunter and Konieczny have also provided similar objections in [1] to Fee-Formula-Independence by pointing cases where this property can be considered as too strong. To address these objections, the authors have proposed to consider the weaker property introduced in [17], called Safe-Formula-Independence.

Inconsistency is often measured by quantifying its origin in a monodimensional way, such as the number of minimal inconsistent subsets [1] and the number of maximal consistent subsets [18]. However, the numerous existing inconsistency measures show in a sense that no measure alone can capture the multiple aspects of inconsistency. Indeed, the inconsistency of a knowledge base may result from several reasons and has to be quantified in a multi-dimensional way. For instance, let us consider the knowledge base $K = \{p, \neg p \wedge q, \neg p \wedge r, \neg p \wedge s\}$. Clearly, the inconsistency of K results from the conflict between the formula p and the subformula $\neg p$ in the other formulæ. If we use the inconsistency measure I_{LP_m} defined in [4] and based on Priest's three-valued logic [19], then we can capture the conflict between p and $\neg p$. Indeed, $\mathsf{I}_{LP_m}(K) = 1$ means that there is only a unique propositional variable which is involved in the conflicts of K. However, since I_{LP_m} considers

K as a single formula, it does not reveal the fact that there are three conflicts between the formulæ of K. To this end, one can use the measure I_{MI} [1] defined as the number of the minimal inconsistent subsets of K ($\mathsf{I}_{MI}(K) = 3$). We are not arguing that the measure I_{MI} is more informative than I_{LP_m} or conversely, but the two measures provide information about incomparable facets of inconsistency. In other words, two measures are not necessarily comparable in the sense that one is better than the other, they can capture incomparable aspects that constitute inconsistency. We think that Besnard's objections to Free-Formula-Independence and Dominance may be used to argue in this sense. For instance, Free-formula-Independence has a sense when we do not consider the internal structure of the formulæ in a knowledge base. Indeed, it simply means that adding a new formula which is not involved in any conflict does not change the amount of inconsistency. However, we need other properties in cases where we consider the internal structure of formulæ. For instance, the fact that I_{LP_m} does not satisfy Free-Formula-independence does not mean that this inconsistency measure is not suitable, since it captures important aspects of inconsistency.

Before introducing our inconsistency measure, we first present in this work results on the compatibility of some state-of-the-art properties of inconsistency measures. Our aim behind establishing these incompatibility results is to show that there is no inconsistency measure that is better than all the others, knowing that most of the proposed properties for measuring inconsistency in the literature have reasonable motivations. Moreover, we aim at providing formal reasons explaining why our inconsistency measure does not satisfy certain properties. We show in particular that the properties of additivity introduced in [1] and [20][1] are incompatible with Dominance [1] and the property of Subsumption Orientation introduced in [16]. We also show that the inconsistency measures that satisfy the properties of Besnard's strong system [16] allow only to distinguish the inconsistent knowledge bases from the consistent ones[2] (the amount of inconsistency is the same in all the inconsistent knowledge bases).

Then, we introduce our inconsistency measure, denoted I_{MCC}, by following an approach based on the use of maximal consistent subsets (MCSes).

[1] Super-Additivity and MinInc Separability.

[2] These measures are called *drastic inconsistency measures* in [1].

Using this measure, the amount of conflicts in a knowledge base is defined as the smallest number of pieces of information that are not in the intersection of MCSes covering all the consistent pieces of informations. The main idea is twofold. First, the inconsistency has to be quantified by considering that all the consistent formulæ of a knowledge are possible. Second, a knowledge base with MCSes sharing a lot of formulæ should be assigned a smaller value of inconsistency than a knowledge base with MCSes sharing a small number of formulæ. Intuitively, by taking into account the formulæ shared between MCSes, we aim at capturing the formulæ that are involved in a small number of conflicts. Furthermore, we describe a multi-agent consensus based interpretation of I_{MCC} using a well known epistemic logic, namely the multimodal logic $\mathsf{S5}$. In this approach, we consider each consistent piece of information as a claim which is possible according to a distinct agent, and the amount of conflicts is defined from the size of the largest consensus between all the agents. A possible consensus is a subset of consistent pieces of information that are not rejected by any agent. An agent rejects a subset of pieces of information if it is inconsistent with the piece of information that she/he supports.

After the description of I_{MCC}, we show that it satisfies several desirable properties proposed in the literature, such as Consistency, Free-Formula-Independence, Monotonicity and Super-additivity. We also point out other interesting properties satisfied by our inconsistency measure such as a generalization of Super-Additivity and a weak variant of Dominance. Then, we show that the problem of computing our inconsistency measure can be formulated as an integer linear program. This encoding is mainly defined from the set of maximal consistent subsets. It is worth noting that the number of the MCSes of a knowledge base is exponential in its size in the worst case. Thus, to avoid the problem of computing the MCSes, we propose a Partial Max-SAT encoding, which is of polynomial size and defined without computing any maximal consistent subset.

Finally, we study relationships between our inconsistency measure and two related state-of-the-art inconsistency measures, namely I_{CC} proposed in [21] and I_{M} proposed in [18]. The measure I_{CC} aims at taking into account the formulæ shared between the minimal inconsistent subsets (MISes). Intuitively, this measure quantifies the amount of conflicts as the number of MISes that can be isolated by removing formulæ. Our comparison of I_{MCC} and I_{CC} is motivated by the fact that these two measures satisfy several existing properties, in particular, the interesting property introduced

in [21], called Independent MI-Additivity[3]. Moreover, I_{MCC} and I_{CC} take into account the distribution of the formulæ among the conflicts. We show in particular that there are inconsistent knowledge bases that are not distinguishable by I_{CC} but distinguishable by I_{MCC}, and conversely. Furthermore, our comparison of I_{MCC} and the inconsistency measure I_M is motivated by the fact that they are both based on MCSes. Indeed, I_M quantifies the amount of conflicts from the number of MCSes: more MCSes means more conflicts. We show that I_{MCC} and I_M do not quantify the amount of conflicts in the same way. Indeed, unlike I_{MCC}, I_M does not take into account the distribution of formulæ among the conflicts.

The rest of the paper is structured as follows. Section 2 provides some preliminary definitions and introduces the problem of measuring inconsistency. Section 3 provides results on the compatibility of some state-of-the-art properties of inconsistency measures. In Section 4, we introduce our inconsistency measure. In Section 5, we show that I_{MCC} satisfies several desirable properties. Then, in Section 6, we present two modeling approaches for the problem of computing I_{MCC}. In Section 7, we provide a comparison between I_{MCC} and two related existing inconsistency measures. Finally, Section 8 presents our conclusion.

2. Formal Setting

2.1. Classical Propositional Logic

In this section, we define the syntax and the semantics of classical propositional logic. Let Prop be a countable set of propositional variables. We use the letters p, q, r, etc to range over Prop. The set of *propositional formulæ*, denoted Form, is defined inductively started from Prop, and using the logical connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$. Notationally, we use the greek letters ϕ, ψ to represent propositional formulæ. Given a syntactic object O, we use $Var(O)$ to denote the set of propositional variables appearing in O. For a set S, we denote by $|S|$ its cardinality.

A *Boolean interpretation* $\mathcal{B}$ of a formula ϕ is defined as a function from $Var(\phi)$ to $\{0, 1\}$ (0 corresponds to $false$ and 1 to $true$). It is inductively

[3]In the original paper, this property is called *Enhanced Additivity*

extended to propositional formulæ as usual. A formula ϕ is consistent if there exists a Boolean interpretation $\mathcal{B}$ of ϕ such that $\mathcal{B}(\phi) = 1$. It is *valid* or *a theorem*, if every Boolean interpretation is a model of ϕ.

It is worth noting that we can restrict the language to the connectives $\neg$ and $\wedge$, since we have the following equivalences: $\phi \vee \psi \equiv \neg(\neg\phi \wedge \neg\psi)$ and $\phi \rightarrow \psi \equiv \neg\phi \vee \psi$.

A *knowledge base* K is a finite set of propositional formulæ.

Definition 1. *Let K be a knowledge base. M is a minimal inconsistent subset (MIS) of K iff (i) $M \subseteq K$, (ii) $M \vdash \bot$ and (iii) $\forall \phi \in M$, $M \setminus \{\phi\} \nvdash \bot$.*

Definition 2. *Let K be a knowledge base. M is a maximal consistent subset (MCS) of K iff (i) $M \subseteq K$, (ii) $M \nvdash \bot$, (iii) $\forall \phi \in K \setminus M$, $M \cup \{\phi\} \vdash \bot$.*

We use $\mathsf{MC}(K)$ (resp. $\mathsf{MI}(K)$) to denote the set of all maximal consistent (resp. minimal inconsistent) subsets of K.

Definition 3. *Let K be a knowledge base and ϕ a formula in K. ϕ is a* free formula *in K iff $\phi \notin M$ for every $M \in \mathsf{MI}(K)$.*

We use $\mathsf{Free}(K)$ to denote the set of free formulæ in K. Moreover, we use $\mathsf{IF}(K)$ to denote the set of inconsistent formulæ in K, i.e., $\mathsf{IF}(K) = \{\phi \in K \mid \phi \vdash \bot\}$.

Let us now describe the SAT problem. A formula ϕ in *Conjunctive Normal Form* (CNF) is a conjunction of clauses, where a *clause* is a disjunction of literals. A *literal* is a positive (p) or negated ($\neg p$) propositional variable. A CNF formula can also be seen as a set of clauses, and a clause as a set of literals. The SAT problem consists in deciding whether a given CNF formula admits a model. We recall that every propositional formula can be translated to CNF using linear Tseitin's encoding [22].

Given a finite set S, we use $|S|$ to denote its cardinality. Moreover, we use $\uplus$ to denote disjoint union, which is just like regular union, except that its operand sets are required to have no element in common. More precisely, given two sets S_1 and S_2, $S_1 \uplus S_2$ means $S_1 \cup S_2$ such that $S_1 \cap S_2 = \emptyset$.

2.2. Measuring Inconsistency and Axioms

In recent years, inconsistent data reasoning has seen a revival in interest because of a number of challenges in terms of collecting, modeling, representing, and querying information. In this context, various logic-based approaches have been proposed in the literature for quantifying the amount

of inconsistency. In the literature, *an inconsistency measure is defined as a function that maps a knowledge base onto a non negative real number*. Several properties have been defined to characterize such measures. In particular, in [1] the authors propose axiomatic properties that any inconsistency measure should satisfy. More precisely, an inconsistency measure I is called a *basic inconsistency measure* if it satisfies the following properties, for all knowledge bases K and K', and for all formulæ ϕ and ψ:

- *Consistency*: $I(K) = 0$ iff K is consistent;
- *Monotonicity*: $I(K) \leq I(K \cup K')$;
- *Free-Formula-Independence*: if $\phi \in \mathsf{Free}(K)$, then $I(K) = I(K \setminus \{\phi\})$;
- *Dominance*: if $\phi \vdash \psi$ and $\phi \nvdash \bot$, then $I(K \cup \{\psi\}) \leq I(K \cup \{\phi\})$.

It is worth noting that Besnard has provided in [16] objections on Free-Formula-Independence and Dominance. In particular, the objection to Free-Formula-Independence comes from the fact that a free formula may be involved in a conflict if we consider the internal structure of formulæ, and in this case it has to increase the amount of inconsistency. Let us consider, for instance, the following knowledge base proposed in [16]: $K = \{p \wedge r, q \wedge \neg r, \neg p \vee \neg q\}$. The knowledge base K has a single minimal inconsistent subset $M = \{p \wedge r, q \wedge \neg r\}$ and, consequently, $\neg p \vee \neg q$ is a free formula in K. Using Free-Formula-Independence, we should have $I(M) = I(K)$. However, p and q are compatible and $p \wedge q$ is contradicted by the free formula $\neg p \vee \neg q$. As a consequence, one can consider that K contains more conflicts than M and in this case Free-Formula-Independence fails. Moreover, Besnard has showed that Free-Formula-Independence is strongly related to the concept of minimal inconsistent subset, which can be seen as a strong restriction in defining inconsistency measures. To illustrate this point, let us consider the following property:

For all K, K', if $\mathsf{MI}(K) \subseteq \mathsf{MI}(K')$, then $I(K) \leqslant I(K')$ (MI-Dependence)

Then, we have the following proposition:

Proposition 1. *I satisfies Monotonicity and Free-Formula-Independence iff I satisfies MI-Dependence.*

Proof.
Part $\Rightarrow$. Assume that I is an inconsistency measure satisfying Free-Formula-Independence and Monotonicity. Let K and K' be two knowledge bases such that $\mathsf{MI}(K) \subseteq \mathsf{MI}(K')$. Since I satisfies Free-Formula-Independence, we have both $I(K) = I(\bigcup_{M \in \mathsf{MI}(K)} M)$ and $I(K') = I(\bigcup_{M \in \mathsf{MI}(K')} M)$. Thus, using Monotonicity, we get $I(K) \leqslant I(K')$, since $\mathsf{MI}(K) \subseteq \mathsf{MI}(K')$.
Part $\Leftarrow$. Assume now that I is an inconsistency measure satisfying MI-Dependence. For all K and K' with $K \subseteq K'$, $\mathsf{MI}(K) \subseteq \mathsf{MI}(K')$ holds. As a consequence, we obtain that I satisfies Monotonicity. We now show that I satisfies also Free-Formula-Independence. Let K be a knowledge base and ϕ a free formula in K. Then, we get $\mathsf{MI}(K) = \mathsf{MI}(K \setminus \{\phi\})$. Thus, using MI-Dependence, $I(K) = I(K \setminus \{\phi\})$ holds. □

Note that Hunter and Konieczny have also provided objections in [1] to Fee-Formula-Independence. Indeed, from the fact that the inconsistency measure I_{LP_m} based on Priest's three-valued logic [19] does not satisfy Fee-Formula-Independence, the authors argue in favor of a weaker property, called Safe-Formula-Independence (also called Weak-Independence in [20]). A *safe formula* in a knowledge base is a consistent formula that does not share any propositional variable with the other formulæ. Clearly, every safe formula is a free formula. Safe-Formula-Independence means that if we add safe formulæ, which have no relation with the existing conflicts, we do not change the amount of inconsistency.

Regarding Besnard's objection to Dominance, it is related to the internal structure of formulæ. Let us consider knowledge bases similar to those proposed in [16]: $K_1 = \{p \wedge q \wedge r, \neg p\}$ and $K_2 = \{p \wedge q \wedge r, \neg p \vee (\neg q \wedge \neg r)\}$. We have $\neg p \vdash \neg p \vee (\neg q \wedge \neg r)$ and $\neg p \nvdash \bot$. Besnard has argued that K_2 is more inconsistent than K_1, since the inconsistency of K_2 comes either from p and $\neg p$ or from $q \wedge r$ and $\neg q \wedge \neg r$, while that of K_1 comes only from p and $\neg p$. The property of dominance is considered as violated in this case because of the used interpretation of disjunction within inconsistency. As pointed out by Besnard, it is a delicate matter to assess how inconsistent a disjunction is. In this context, the author has introduced the following property, called Disjunction-Maximality:

$$I(K \cup \{\phi \vee \psi\}) \leqslant Max(I(K \cup \{\phi\}), I(K \cup \{\psi\}))$$

However, if we consider, for instance, an inconsistency measure as a function associating to each knowledge base the amount of effort needed to resolve its possible inconsistency or the cost that an agent should pay because of the possible inconsistency of his knowledge base, then it is reasonable to use the following bound:

$$I(K \cup \{\phi \vee \psi\}) \leqslant Min(I(K \cup \{\phi\}), I(K \cup \{\psi\}))$$

since by resolving the inconsistency in $K \cup \{\phi\}$ or in $K \cup \{\psi\}$ we resolve the inconsistency in $K \cup \{\phi \vee \psi\}$. Using this interpretation of measuring inconsistency, Besnard's objection to the property of dominance can be avoided. Moreover, consider the knowledge base $K = \{p, p \vee \neg p\}$. The right part of the law of excluded middle is involved in a conflict. If we consider the interpretation used in Besnard's objection to the dominance property, K is considered as containing more conflicts than $\{p\}$, however, K and $\{p\}$ are both consistent.

Other objections to Dominance can be obtained from its incompatibility with some state-of-the-art properties, which have reasonable motivations. For instance, we show in Section 3 that Dominance is incompatible with the additivity properties introduced in [1, 20].

We agree with Besnard's objections in the sense that it is not suitable to require Hunter and Konieczny's basic properties for every inconsistency measure. However, we think that inconsistency is a multi-dimensional concept and a single inconsistency measure is insufficient to capture all the information about the amount of conflicts. In this context, to capture certain aspects that constitute inconsistency, we need inconsistency measures satisfying the basic properties.

3. Additivity and Inconsistent Axiomatization Systems

In this section, we consider important properties on inconsistency measures introduced in the literature. We show in particular that combinations of some properties lead to inconsistent systems. An axiomatization system is inconsistent if there is no inconsistency measure satisfying all its axioms.

Our aim behind establishing incompatibility results between state-of-the-art properties is twofold. First, showing that there is no inconsistency measure that is better than all the others, knowing that most of the proposed

properties for measuring inconsistency in the literature have reasonable motivations. Second, providing formal reasons explaining why our inconsistency measure, introduced in Section 4, does not satisfy certain properties.

3.1. Properties for Measuring Inconsistency

We here present some state-of-the-art properties on inconsistency measures. The aim of such properties is to provide a reasonable way for measuring the amount of conflicts.

Let us first consider the additivity properties introduced in [20] and [1] respectively:

- Super-Additivity: if $K \cap K' = \emptyset$, then $I(K \cup K') \geqslant I(K) + I(K')$.
- MI-Additivity[4]: if $\mathsf{MI}(K \cup K') = \mathsf{MI}(K) \uplus \mathsf{MI}(K')$, then $I(K \cup K') = I(K) + I(K')$, where $\uplus$ stands disjoint union.

Super-Additivity means that the amounts of inconsistency in two disjoint knowledge bases are preserved in the union of these bases. It is worth noting that Super-Additivity and Consistency implies Monotonicity. Regarding MI-Additivity, it captures a similar idea to Super-Additivity, but it relates the amount of conflicts to the minimal inconsistent subsets.

Let us now define two concepts that are used in the definition of Besnard's postulates. First, a *substitution* is a mapping $\sigma : \mathsf{Prop} \to \mathsf{Form}$ from the set of propositional variables to the set of propositional formulæ. Then, given a propositional formula ϕ and a substitution σ, $\sigma(\phi)$ is the result of replacing each propositional variable p with $\sigma(p)$. For instance, for $\sigma(p) = r \wedge s$ and $\sigma(q) = r \rightarrow s$, we get $\sigma(\neg p \wedge q) = \neg\sigma(p) \wedge \sigma(q) = \neg(r \wedge s) \wedge (r \rightarrow s)$. Given a knowledge base K, we use $\sigma(K)$ to denote the knowledge base $\bigcup_{\phi \in K} \{\sigma(\phi)\}$. Second, a *primitive conflict* is a notion that is considered to quantify the amount of inconsistency. For example, one can consider the notion of minimal inconsistent subset and, in this case, the primitive conflicts of a knowledge base are its minimal inconsistent subsets. Besnard has used primitive conflict as an abstract notion that allows to represent the conflicts in a knowledge base considered by a measurer.

In the following we describe Besnard's postulates proposed in [16]:

[4]In the original paper, this property is called *MinInc Separability.*

- Subsumption-Orientation: if $\mathcal{C}(\sigma K) \subseteq \mathcal{C}(K')$ for a substitution σ then $I(K) \leqslant I(K')$, where $\mathcal{C}(K)$ is the set of primitive conflicts in K.
- Conjunction-Dominance: $I(K \cup \{\phi \wedge \psi\}) \geqslant I(K \cup \{\phi\})$.
- Tautology-Independence: if $\phi \equiv \top$ then $I(K \cup \{\phi\}) = I(K)$.
- Rewriting: if ψ is a prenormal form of ϕ then $I(K \cup \{\phi\}) = I(K \cup \{\psi\})$ where ψ is a prenormal form of ϕ if ψ is obtained from ϕ by applying (possibly repeatly) one or more of the following principles: commutativity, associativity, distribution from $\wedge$ and $\vee$, De Morgan laws, double negation equivalence.
- Instance-Low: if $\sigma K \subseteq K'$ for some substitution σ then $I(K) \leqslant I(K')$.
- Disjunction-Maximality: $I(K \cup \{\phi \vee \psi\}) \leqslant max(I(K \cup \{\phi\}), I(K \cup \{\psi\}))$.
- Disjunction-Minimality: $I(K \cup \{\phi \vee \psi\}) \geqslant min(I(K \cup \{\phi\}), I(K \cup \{\psi\}))$.
- Exchange: if $K' \equiv K''$ and $K' \nvdash \bot$ then $I(K \cup K') = I(K \cup K'')$.
- Adjunction-Invariancy: $I(K \cup \{\phi, \psi\}) = I(K \cup \{\phi \wedge \psi\})$.

One can see that, unlike Hunter and Konieczny's properties, most of these properties take explicitly into account the internal structure of formulæ in a knowledge base. For instance, as explained in [16], Rewriting takes into account any inessential difference in which a formula can be be written. Note that Besnard's *strong system* is defined as Consistency with Rewriting, Instance-Low, Disjunction-Maximality, Disjunction-Minimality, Exchange, Adjunction-Invariancy. Moreover, note that the properties in the strong system entail the following properties: Conjunction-Dominance and Tautology-Independence. Furthermore, all the properties in the strong system can be entailed by Subsumption-Orientation from specific properties on $\mathcal{C}$. For example, if $\mathcal{C}$ satisfy the property $\mathcal{C}(K \cup \{\phi\}) = \mathcal{C}(K \cup \{\phi'\})$ for every knowledge base K, formula ϕ and ϕ' prenormal form of ϕ, then Rewriting can be derived from Subsumption-Orientation (for more details, see [16]).

3.2. Dominance and Additivity

We here study the consistency of systems combining Super-Additivity and MI-Additivity with Dominance. Our main aim is to point out incompatibility results between these properties.

In the following proposition, we show that MI-Additivity is stronger than Free-Formula-Independence.

Proposition 2. *Given an inconsistency measure I, if I satisfies Consistency and MI-Additivity, then I satisfies Free-Formula-Independence.*

Proof. Let K be a knowledge base. Then we have $\mathsf{MI}(K) = \mathsf{MI}(K \setminus \mathsf{Free}(K))$. Thus, we get $\mathsf{MI}(K) = \mathsf{MI}(K \setminus \mathsf{Free}(K)) \uplus \mathsf{MI}(\mathsf{Free}(K))$ since $\mathsf{MI}(\mathsf{Free}(K)) = \emptyset$. As a consequence, using MI-Additivity, we get $I(K) = I(K \setminus \mathsf{Free}(K)) + I(\mathsf{Free}(K))$. Using Consistency, we get $I(\mathsf{Free}(K)) = 0$ since $\mathsf{Free}(K) \nvdash \bot$. Thus, $I(K) = I(K \setminus \mathsf{Free}(K))$ holds. □

We now show that Dominance is incompatible with the considered additivity properties.

Proposition 3. *The following systems are inconsistent:*

1. {*Consistency, Dominance, Super-Additivity*};
2. {*Consistency, Dominance, MI-Additivity*}.

Proof.
(1). Assume that there exists an inconsistency measure I satisfying Consistency, Dominance and Super-Additivity. Let $K = \{p \wedge q, \neg p \wedge q\}$ be a knowledge base. Using Dominance twice, $I(K \cup \{p, \neg p\}) \leqslant I(K \cup \{p \wedge q, \neg p\}) \leqslant I(K \cup \{p \wedge q, \neg p \wedge q\})$ holds, since $p \wedge q \nvdash \bot$, $\neg p \wedge q \nvdash \bot$, $p \wedge q \vdash p$ and $\neg p \wedge q \vdash \neg p$. Note that $K \cup \{p \wedge q, \neg p \wedge q\} = K$. Moreover, using Super-Additivity, $I(K \cup \{p, \neg p\}) \geqslant I(K) + I(\{p, \neg p\})$ holds. Then, using Consistency, $I(\{p, \neg p\}) > 0$ holds and, as a consequence, $I(K \cup \{p, \neg p\}) > I(K)$ holds and we get a contradiction.
(2). Assume that there exists an inconsistency measure I satisfying Consistency, Dominance and MI-Additivity. Let $K = \{p \wedge r, q \wedge \neg r, \neg p \vee \neg q\}$ be a knowledge base. Using Dominance, $I(K \cup \{p, q\}) \leqslant I(K \cup \{p \wedge r, q \wedge \neg r\})$ holds, since $p \wedge r \vdash p$ and $q \wedge \neg r \vdash q$. Note that $K \cup \{p \wedge r, q \wedge \neg r\} = K$. We have $\mathsf{MI}(K \cup \{p, q\}) = \mathsf{MI}(K \setminus \{\neg p \vee \neg q\} \cup \{p, q, \neg p \vee \neg q\}) = \mathsf{MI}(K) \uplus \mathsf{MI}(\{p, q, \neg p \vee \neg q\})$. Then, using MI-Additivity, $I(K \cup \{p, q\}) =$

$I(K) + I(\{p, q, \neg p \vee \neg q\})$ holds. Using Consistency, $I(K \cup \{p, q\}) > I(K)$ holds since $I(\{p, q, \neg p \vee \neg q\}) > 0$, and we get a contradiction. □

In our proof of Proposition 3, we use the fact that Dominance allows us to add to a knowledge base logical consequences of its formulæ without changing the amount of conflicts. Indeed, Dominance does not exactly express that the amount of conflicts does not increase when a consistent formula is replaced with one of its logical consequences. In order to avoid this problem, we consider the following variant of Dominance:

- Weak-Dominance: if $\phi \notin K$, $\phi \vdash \psi$ and $\phi \nvdash \bot$, then $I(K \cup \{\psi\}) \leq I(K \cup \{\phi\})$.

The condition $\phi \notin K$ in Weak-Dominance means $\phi \notin K \cup \{\psi\}$, which allows us to express that ϕ is replaced with ψ.

We now show that Super-Additivity is compatible with Weak-Dominance. To this end, we use the inconsistency measure I_{CC} introduced in [21]. This measure takes into account the formulæ shared between the minimal inconsistent subsets. Given a knowledge base K, an MI-decomposition of K is a pair $(\{K_1, \ldots, K_n\}, K')$ satisfying the following properties:

(*i*) $(\bigcup_{i=1}^{n} K_i) \cup K' = K$;

(*ii*) $(\bigcup_{i=1}^{n} K_i) \cap K' = \emptyset$;

(*iii*) $K_i \vdash \bot$ for every $1 \leqslant i \leqslant n$;

(*iv*) $K_i \cap K_j = \emptyset$ for every $1 \leqslant i \neq j \leqslant n$; and

(*v*) $\mathsf{MI}(\bigcup_{i=1}^{n} K_i) = \biguplus_{i=1}^{n} \mathsf{MI}(K_i)$.

Given a knowledge base K, $\mathsf{I}_{CC}(K) = n$ if there is an MI-decomposition (D, K') such that $|D| = n$, and there is no MI-decomposition (D', K'') such that $|D'| > n$. Intuitively, this measure can be seen as the maximum number of minimal inconsistent subsets that can be isolated by removing formulæ.

For example, consider the knowledge base $K = \{p, \neg p \wedge q, \neg q, q \wedge r, q \wedge \neg r\}$. Then, we have $\mathsf{MI}(K) = \{\{p, \neg p \wedge q\}, \{\neg p \wedge q, \neg q\}, \{\neg q, q \wedge r\}, \{\neg q, q \wedge \neg r\}, \{q \wedge r, q \wedge \neg r\}\}$. Let us note that each one of the formulæ $\neg p \wedge q$, $\neg q$, $q \wedge r$ and $q \wedge \neg r$ belongs to at least two minimal inconsistent subsets. To build an MI-decomposition, we have to put some of the previous formulæ in the right

side. For example, the pair $S = (\{\{p, \neg p \wedge q\}, \{q \wedge r, q \wedge \neg r\}\}, \{\neg q\})$ is an MI-decomposition. Indeed, we have: Condition (i): $\{p, \neg p \wedge q\} \cup \{q \wedge r, q \wedge \neg r\} \cup \{\neg q\} = K$; Condition (ii): $(\{p, \neg p \wedge q\} \cup \{q \wedge r, q \wedge \neg r\}) \cap \{\neg q\} = \emptyset$; Condition (iii): $\{p, \neg p \wedge q\} \vdash \bot$ and $\{q \wedge r, q \wedge \neg r\} \vdash \bot$; Condition (iv): $\{p, \neg p \wedge q\} \cap \{q \wedge r, q \wedge \neg r\} = \emptyset$; and Condition (v): $\mathsf{MI}(\{p, \neg p \wedge q\} \cup \{q \wedge r, q \wedge \neg r\}) = \{\{p, \neg p \wedge q\}, \{q \wedge r, q \wedge \neg r\}\} = \mathsf{MI}(\{p, \neg p \wedge q\}) \uplus \mathsf{MI}(\{q \wedge r, q \wedge \neg r\})$. Moreover, one can see that there is no MI-decomposition (D, K') of K such that $|D| > 2$, since there is no more than two disjoint minimal inconsistent subsets. As a consequence, we get $\mathsf{I}_{CC}(K) = 2$.

Proposition 4. *The system* {*Consistency, Free-Formula-Independence, Monotonicity, Weak-Dominance, Super-Additivity*} *is consistent.*

Proof. We here show that I_{CC} satisfies all the properties in the previous system. One can easily see that I_{CC} satisfies Consistency, since $\mathsf{I}_{CC}(K) = 0$ iff $\mathsf{MI}(K) = \emptyset$. It satisfies also Free-Formula-Independence since the amount of conflicts is computed through the minimal inconsistent subsets. Moreover, Monotonicity is implied by Consistency and Super-Additivity.
Super-Additivity. Let K and K' be two knowledge bases s.t. $K \cap K' = \emptyset$, and $S = (D, K_1)$ and $S' = (D', K_2)$ MI-decompositions of K and K' respectively such that $\mathsf{I}_{CC}(K) = |D|$ and $\mathsf{I}_{CC}(K') = |D'|$. Thus, we obtain that $S^3 = (D \uplus D', K_1 \uplus K_2)$ is an MI-decomposition of $K \cup K'$. Indeed, S^3 satisfies the conditions (i) and (iii) since these conditions are satisfied by S and S'. Moreover, S^3 satisfies the conditions (ii), (iv) and (v), since $K \cap K' = \emptyset$ and these conditions are satisfied by S and S'. As a consequence, $\mathsf{I}_{CC}(K \uplus K') \geqslant \mathsf{I}_{CC}(K) + \mathsf{I}_{CC}(K')$ holds.
Weak-Dominance. Let K be a knowledge base, and ϕ and ψ two formulæ s.t. $\phi \notin K$, $\phi \nvdash \bot$ and $\phi \vdash \psi$. If $\psi \in K$, then we get $K \cup \{\psi\} \subseteq K \cup \{\phi\}$. Using the fact that I_{CC} satisfies Monotonicity, $\mathsf{I}_{CC}(K \cup \{\phi\}) \geqslant \mathsf{I}_{CC}(K \cup \{\psi\})$ holds. Let us now consider that $\psi \notin K$. Let $S = (\{K_1, \ldots, K_n\}, K')$ be an MI-decomposition of $K \cup \{\psi\}$. If $\psi \notin K_i$ for every $1 \leqslant i \leqslant n$, then $S' = (\{K_1, \ldots, K_n\}, (K' \setminus \{\psi\}) \cup \{\phi\})$ is an MI-decomposition of $K \cup \{\phi\}$. Otherwise, there is a unique $i \in 1..n$ such that $\psi \in K_i$. We have $(K_i \setminus \{\psi\}) \cup \{\phi\} \vdash \bot$ since $\phi \vdash \psi$. Hence, there exists $K_i' \in \mathsf{MI}(K \cup \{\phi\})$ s.t. $K_i' \subseteq (K_i \setminus \{\psi\}) \cup \{\phi\}$ and $\phi \in K_i'$. Thus, there is a K'' s.t. $(\{K_1, \ldots, K_i', \ldots, K_n\}, K'')$ is an MI-decomposition of $K \cup \{\phi\}$. As a consequence, we know that if $K \cup \{\psi\}$ has an MI-decomposition (D, K'), then $K \cup \{\phi\}$ has an MI-decomposition (D', K'') such that $|D| \leqslant |D'|$. Therefore, we obtain $\mathsf{I}_{CC}(K \cup \{\phi\}) \geq \mathsf{I}_{CC}(K \cup \{\psi\})$. □

In the following proposition, we show that, contrary to Super-Additivity, MI-Additivity is also incompatible with Weak-Dominance.

Proposition 5. *The system* {*Consistency, Weak-Dominance, MI-Additivity*} *is inconsistent.*

Proof. Assume that there is an inconsistency measure satisfying the properties of Consistency, Weak-Dominance and MI-Additivity. Let $K_1 = \{\neg q, p \wedge (\neg p \vee q)\}$ and $K_2 = \{p, \neg q, \neg p \vee q\}$ be two knowledge bases. Using Consistency, we have both $I(K_1)$ and $I(K_2)$ greater than 0. We define n as an integer strictly greater than $\frac{I(K_1)}{I(K_2)}$. Let us now consider the knowledge base $K_3 = \{p \wedge r_1, \ldots, p \wedge r_n, \neg q, p \wedge (\neg p \vee q)\}$ where $r_1, \ldots, r_n$ are n distinct fresh propositional variables. Then, using Proposition 2, we get $I(K_3) = I(K_1)$ since $\mathsf{MI}(K_3) = \mathsf{MI}(K_1) = \{\{\neg q, p \wedge (\neg p \vee q)\}\}$. Moreover, using Weak-Dominance and $p \wedge (\neg p \vee q) \vdash \neg p \vee q$, we have $I(K_3) \geqslant I(K_4)$ where $K_4 = \{p \wedge r_1, \ldots, p \wedge r_n, \neg q, \neg p \vee q\}$. Using MI-Additivity, we obtain $I(K_4) = I(\{p \wedge r_1, \neg q, \neg p \vee q\}) + \cdots + I(\{p \wedge r_n, \neg q, \neg p \vee q\})$, since $\mathsf{MI}(K_4) = \{\{p \wedge r_1, \neg q, \neg p \vee q\}, \cdots, \{p \wedge r_n, \neg q, \neg p \vee q\}\}$. Then, using Weak-Dominance, $I(K_4) \geqslant n \times I(K_2)$ holds since $p \wedge r_i \vdash p$ for every $i \leqslant i \leqslant n$. As a consequence, we obtain $I(K_3) \geqslant n \times I(K_2)$. Thus, we get a contradiction since $I(K_3) = I(K_1)$ and $n > \frac{I(K_1)}{I(K_2)}$. □

It is worth noting that Proposition 5 implies the second property in Proposition 3, since Dominance is stronger than Weak-Dominance.

3.3. Subsumption-Orientation and Additivity

In the same way as for Dominance, we here show incompatibility results between Subsumption-Orientation and the considered additivity properties. We also show that the inconsistency measures that satisfy the properties of Besnard's strong system allow only to distinguish the inconsistent knowledge bases from the consistent ones[5] (the amount of inconsistency is the same for all the knowledge bases that are inconsistent).

Proposition 6. *The following systems are inconsistent:*

1. $S_1 = \{$*Consistency, Subsumption-Orientation, Super-Additivity*$\}$;

[5]These measures are called *drastic inconsistency measures* in [1].

2. $S_2 = \{$*Consistency, Subsumption Orientation, MI-Additivity*$\}$.

Proof.
(1) and (2). Assume that there exists an inconsistency measure satisfying the properties of S_1 (resp. S_2). Let $K = \{p, \neg p, q, \neg q\}$ be a knowledge base and σ a substitution such that $\sigma(p) = p$ and $\sigma(q) = p$. Then, using Super-Additivity (resp. MI-Additivity), we get $I(K) \geqslant I(\{p, \neg p\}) + I(\{q, \neg q\})$ (resp. $I(K) = I(\{p, \neg p\}) + I(\{q, \neg q\})$) since $K = \{p, \neg p\} \uplus \{q, \neg q\}$ (resp. $\mathsf{MI}(K) = \mathsf{MI}(\{p, \neg p\}) \uplus \mathsf{MI}(\{q, \neg q\})$). Moreover, since $\sigma(K) = \{p, \neg p\}$, we get $\mathcal{C}(\sigma(K)) = \mathcal{C}(\{p, \neg p\})$. Thus, using Subsumption-Orientation, $I(K) \leqslant I(\{p, \neg p\})$ holds. Using Consistency, we get a contradiction since we have $I(\{q, \neg q\}) > 0$. □

Proposition 7. *An inconsistency measure I satisfies the properties of the strong system iff it is defined as follows:*

$$I(K) = \begin{cases} 0 & if\ K \nvdash \bot \\ n & otherwise \end{cases}$$

where n is a constant different from 0.

Proof. One can consider w.l.o.g. that all the formulæ in the knowledge bases are consistent. Indeed, for each knowledge base K, using Rewriting and Adjunction-Invariancy, there exists a set of clauses K' such that $I(K) = I(K')$. This comes from the fact that each propositional formula can be rewritten into a formula in Conjunctive Normal Form (CNF) using De Morgan's laws. Moreover, we know that each non-empty clause is consistent.
Claim 1: For all inconsistent knowledge base K and $p \in Var(K)$, $I(K) \leqslant I(\{p, \neg p\})$.
Let K be an inconsistent knowledge base such that $\phi \nvdash \bot$ for every $\phi \in K$. Then, we get $\phi \equiv \bigwedge_{\mathcal{B} \in Mod(\neg\phi)} \overline{\mathcal{B}}$ for every $\phi \in K$, where $Mod(\neg\phi)$ is the set of the models of $\neg\phi$ over $Var(K)$ and $\overline{\mathcal{B}}$ denotes the clause $(\bigvee_{p \in Var(K), \mathcal{B}(p)=1} \neg p) \vee (\bigvee_{q \in Var(K), \mathcal{B}(p)=0} q)$. Let us note that we consider Boolean interpretations built over the set $Var(K)$. Using Exchange and Adjunction-Invariancy, $I(K) = I(K')$ where $K' = \bigcup_{\phi \in K} \{\overline{\mathcal{B}} \mid \mathcal{B} \in Mod(\neg\phi)\}$. Let p be a propositional variable in $Var(K)$ and σ a substitution defined by $\sigma(q) = p$ for every $q \in Var(K)$. Using Instance-Low, we get $I(K) \leqslant I(\sigma(K))$. Then, using Tautology-Independence and Rewriting, we get $I(\sigma(K)) = I(\{p, \neg p\})$. Indeed, each positive clause $p_1 \vee \cdots \vee p_n$ (resp. negative clause $\neg p_1 \vee \cdots \vee \neg p_n$)

is transformed into $p \vee \cdots \vee p$ (resp. $\neg p \vee \cdots \vee \neg p$) using σ which is equivalent to p (resp. $\neg p$). Moreover, each clause of the form $p_1 \vee \cdots \vee p_l \vee \neg q_1 \vee \cdots \vee \neg q_m$ is transformed into $p \vee \cdots \vee p \vee \neg p \vee \cdots \vee \neg p$ using σ which is a tautology. As a consequence, we get $I(K) \leqslant I(\{p, \neg p\})$.
Claim 2: For all inconsistent knowledge base K and $p \in Var(K)$, $I(K) \geqslant I(\{p, \neg p\})$.
We have $K' = K_1 \uplus K_2$ where $K_1 = \{\overline{\mathcal{B}} \in K' \mid \mathcal{B}(p) = 1\}$ and $K_2 = \{\overline{\mathcal{B}} \in K' \mid \mathcal{B}(p) = 0\}$. Clearly, K_1 and K_2 are consistent and, as a consequence, they are not empty since K' is inconsistent. Moreover, using the principle of distribution of conjunction over disjunction, we know that there exist two formulæ ψ_1 and ψ_2 such that $K_1 \equiv p \wedge \psi_1$ and $K_2 \equiv \neg p \wedge \psi_2$. Then, using Conjunction-Dominance and Exchange, we get $I(\{p, \neg p\}) \leqslant I(K') = I(K)$.

As a consequence, using *Claim 1* and *Claim 2*, we get $I(K) = I(\{p, \neg p\})$. Moreover, using Instance-Low, we have $I(\{p, \neg p\}) = I(\{q, \neg q\})$ for every two propositional variables p and q. Therefore, there exists $n > 0$ such that, for all inconsistent knowledge base K, $I(K) = n$ holds. □

4. MCS-Cover based Inconsistency Measure

In this section, we introduce a new inconsistency measure, denoted I_{MCC}, which is based on the use of maximal consistent subsets (MCSes). To define I_{MCC}, we use a concept called MCS-cover, which consists in a set of MCSes covering all the consistent pieces of informations. Technically, the amount of conflicts in a knowledge base using our measure is defined as the smallest number of pieces of information that are not in all the elements of an MCS-cover. Furthermore, we provide an epistemic interpretation of our inconsistency measure using the multimodal logic $\mathsf{S5}$.

4.1. MCS-Cover and I_{MCC} measure

Let us first define fundamental concepts that will be useful in the sequel.

Definition 4 (MCS-Cover)**.** *Let K be a knowledge base. An MCS-cover $\mathcal{C}$ of K is a subset of $\mathsf{MC}(K)$ such that $\bigcup_{S \in \mathcal{C}} S = K \setminus \mathsf{IF}(K)$.*

In other words, an MCS-cover of a knowledge base is a subset of its MCSes that cover all the consistent formulæ. Intuitively, an MCS-cover of a knowledge base can be seen as a possible scenario of the origin of its consistent formulæ in the sense that each MCS can be viewed as set of pieces of information provided by a possible source agent. Following this interpretation,

the intersection of the elements of an MCS-cover can be seen as a possible consensus between the different possible source agents. This interpretation is formally described in Section 4.2.

Example 1. *Let us consider, for instance, the knowledge base* $K = \{\neg p \vee \neg q, \neg p \vee \neg r, \neg q \vee \neg r, p, q, r, r \wedge \neg r\}$. *The following two sets are MCS-covers of* K: $\mathcal{C}_1 = \{\{\neg p \vee \neg q, \neg p \vee \neg r, \neg q \vee \neg r, p\}, \{p, q, r\}\}$, $\mathcal{C}_2 = \{\{\neg p \vee \neg q, \neg p \vee \neg r, \neg q \vee \neg r, p\}, \{\neg p \vee \neg q, \neg p \vee \neg r, q, r\}\}$. *Indeed, we have* $\{\neg p \vee \neg q, \neg p \vee \neg r, \neg q \vee \neg r, p\} \cup \{p, q, r\} = \{\neg p \vee \neg q, \neg p \vee \neg r, \neg q \vee \neg r, p\} \cup \{\neg p \vee \neg q, \neg p \vee \neg r, q, r\} = K \setminus \{r \wedge \neg r\}$.

We now define a complete preorder relation on the MCS-covers of a given knowledge base, denoted $\succeq$. Let K be a knowledge base. For all $\mathcal{C}$ and $\mathcal{C}'$ two MCS-covers of K, $\mathcal{C} \succeq \mathcal{C}'$ if and only if $|\bigcap_{M \in \mathcal{C}} M| \geqslant |\bigcap_{M' \in \mathcal{C}'} M'|$. Let us consider again the previous example. We get $\mathcal{C}_2 \succeq \mathcal{C}_1$ since $|\{\neg p \vee \neg q, \neg p \vee \neg r, \neg q \vee \neg r, p\} \cap \{p, q, r\}| = 1$ and $|\{\neg p \vee \neg q, \neg p \vee \neg r, \neg q \vee \neg r, p\} \cap \{\neg p \vee \neg q, \neg p \vee \neg r, q, r\}| = 2$.

Definition 5 (Normal MCS-Cover)**.** *Let* K *be a knowledge base and* $\mathcal{C}$ *an MCS-cover of* K. *Then,* $\mathcal{C}$ *is a normal MCS-cover if* $\mathcal{C}'$ *is not an MCS-cover for every* $\mathcal{C}' \subset \mathcal{C}$.

The normalization aims at considering a minimal number of MCSes, since we intend to maximize the number of shared formulæ between MCSes. Indeed, one can easily see that if $\mathcal{C}$ is an MCS-cover which is not normal than there exists an MCS-cover $\mathcal{C}' \subset \mathcal{C}$ such that the MCSes in $\mathcal{C}'$ share at least the same number of formulæ as those in $\mathcal{C}$. For instance, in Example 1, the MCS-covers $\mathcal{C}_1$ and $\mathcal{C}_2$ are normal, but the MCS-cover $\mathcal{C}_3 = \{\{\neg p \vee \neg q, \neg p \vee \neg r, \neg q \vee \neg r, p\}, \{\neg p \vee \neg q, \neg p \vee \neg r, q, r\}, \{p, q, r\}\}$ is not since $\mathcal{C}_2 \subset \mathcal{C}_3$.

Definition 6 (Maximum MCS-Cover)**.** *Let* K *be a knowledge base and* $\mathcal{C}$ *an MCS-cover of* K. *Then,* $\mathcal{C}$ *is said to be a maximum MCS-cover of* K *if it is normal and* $\forall\ \mathcal{C}'$ *MCS-cover of* K, $\mathcal{C} \succeq \mathcal{C}'$. *We denote by* $\lambda(K)$ *the value* $|\bigcap_{M \in \mathcal{C}} M|$.

Let us consider again Example 1. The set $\mathcal{C}_2$ is a maximum MCS-cover of K, since there is no subset of MCSes of K that covers all the consistent formulæ of K such that the number of shared formulæ between the MCSes of this subset is greater than 2.

Intuitively, a maximum MCS-cover of a knowledge base captures the maximum number of its consistent formulæ that do not contradict any consistent formula in this base. More precisely, given a knowledge base K and a maximum MCS-cover $\mathcal{C} = \{M_1, \ldots, M_n\}$ of K, we know that $\bigcap_{i=1}^{n} M_i \nvdash \neg\phi$ for every $\phi \in K$ with $\phi \nvdash \bot$. Indeed, this property is a consequence of the fact that the MCSes of $\mathcal{C}$ cover all the consistent formulæ in K.

Classical reasoning states that an inconsistent knowledge base contains no useful information, which is counter-intuitive in several cases. For instance, it is reasonable to consider that the knowledge base $\{p, q, \neg q\}$ is more informative (or "less inconsistent") than $\{p, \neg p, q, \neg q\}$, since the formula p in the first knowledge base is not involved in any conflict. This explains the need to have techniques and principles that allow us to analyze inconsistent information, such as inconsistency measures. Let us recall that an inconsistency measure is defined as a function that maps a knowledge base onto a non negative real number.

We now present our inconsistency measure, denoted I_{MCC}, which is based on the notion of maximum MCS-cover. The main idea is twofold. First, the inconsistency has to be quantified by considering that all the consistent formulæ of a knowledge are possible. This explains why we use the notion of MCS-cover. Second, a knowledge base with MCSes sharing a lot of formulæ should be assigned a smaller value of inconsistency than a knowledge base with MCSes sharing a small number of formulæ. Intuitively, by taking into account the formulæ shared between MCSes, we aim at capturing the formulæ that are involved in a small number of conflicts. In particular, a formula in all the MCSes of a knowledge base does not belong to any minimal inconsistent subset.

Definition 7 (I_{MCC} Measure)**.** *The inconsistency measure I_{MCC} is defined as follows:* $\mathsf{I}_{MCC}(K) = |K| - \lambda(K)$.

In other words, I_{MCC} corresponds to the minimum number of formulæ that can not be shared between MCSes covering all the consistent formulæ. Regarding the knowledge base in Example 1, we have $\mathsf{I}_{MCC}(K) = 5$ since $\mathcal{C}_2$ is a maximum MCS-cover and $r \wedge \neg r$ is the unique inconsistent formula in K.

Example 2. *A propositional classical logic can be used to represent preferences of an agent. This can be be achieved by considering the models*

of a formula as the preferred outcomes and its counter-models as the rejected outcomes. For instance, the formula $(fish \rightarrow white\_wine) \wedge (meat \rightarrow red\_wine) \wedge (\neg red\_wine \vee \neg white\_wine)$ *means that one prefers to take white wine with fish and red wine with meat. Moreover, one rejects white wine with meat and red wine with fish. In this context, let us consider the following knowledge base of an agent:*

$$K = \left\{ \begin{array}{l} \phi_1 : (fish \rightarrow (white\_wine \wedge tea)) \wedge (meat \rightarrow (red\_wine \wedge cafe)), \\ \phi_2 : (cheese \rightarrow (red\_wine \wedge tea)) \wedge (cake \rightarrow (white\_wine \wedge cafe)), \\ \phi_3 : (fish \wedge cheese) \wedge (\neg red\_wine \vee \neg white\_wine) \wedge \\ (\neg cheese \vee \neg cake) \wedge (\neg cafe \vee \neg tea), \\ \phi_4 : cafe \vee tea \end{array} \right\}$$

Clearly, K *is inconsistent, since the agent prefers fish with white wine* (ϕ_1) *and cheese with red wine* (ϕ_2) *but she/he also prefers fish with cheese* (ϕ_3). *It is worth noting that the knowledge base* K *admits three maximum MCS-covers:* $\mathcal{C}_1 = \{\{\phi_1, \phi_2, \phi_4\}, \{\phi_1, \phi_3, \phi_4\}\}$, $\mathcal{C}_2 = \{\{\phi_1, \phi_2, \phi_4\}, \{\phi_2, \phi_3, \phi_4\}\}$ *and* $\mathcal{C}_3 = \{\{\phi_1, \phi_3, \phi_4\}, \{\phi_2, \phi_3, \phi_4\}\}$. *The number of shared formulæ between the MCSes of each maximum MCS-cover is equal to* 2, *i.e.,* $\lambda(K) = 2$. *As a consequence, we get* $\mathsf{I}_{MCC}(K) = 4 - 2 = 2$. *The number* 2 *here means that we have to ignore at least two formulæ in* K *to not contradict any formula in* K. *Indeed, if we ignore a single formula, then we know that it is contradicted by the four remaining formulæ. For instance, if we ignore only* ϕ_3, *then we have* $\{\phi_1, \phi_2, \phi_4\} \vdash \neg\phi_3$. *However, if we ignore some two formulæ, for instance, the formulæ that are not shared between the MCSes of the maximum MCS-cover* $\mathcal{C}_2$, *then these two formulæ are not contradicted by the formulæ shared between the MCSes of* $\mathcal{C}_2$, *i.e.,* $\{\phi_2, \phi_4\} \nvdash \neg\phi_1$ *and* $\{\phi_2, \phi_4\} \nvdash \neg\phi_3$. *In other words, by considering for the preferences of the agent the formulæ* ϕ_2 *and* ϕ_4, *the two formulae* ϕ_1 *and* ϕ_3 *remain possible (they are satisfied by interpretations that satisfy* $\phi_2 \wedge \phi_4$*).*

Let us now consider the knowledge base K' *obtained from* K *by replacing* ϕ_4 *with* $cafe$, *i.e.,* $K' = \{\phi_1, \phi_2, \phi_3, cafe\}$. *In the same way as* K, *the knowledge base* K' *admits three maximum MCS-covers:* $\mathcal{C}_1 = \{\{\phi_1, \phi_2, \phi_4\}, \{\phi_1, \phi_3\}\}$, $\mathcal{C}_2 = \{\{\phi_1, \phi_2, \phi_4\}, \{\phi_2, \phi_3\}\}$ *and* $\mathcal{C}_3 = \{\{\phi_1, \phi_2, \phi_4\}, \{\phi_3, \phi_4\}\}$. *Thus, we get* $\mathsf{I}_{MCC}(K') = 3$ *since* $\lambda(K') = 1$. *As a consequence, following our inconsistency measure, the amount of conflicts in* K' *is greater than that in* K. *This can be explained by the fact that the formula* $cafe \vee tea$ *is not involved in any conflict in* K, *but* $cafe$ *is involved in a conflict in* K'.

4.2. An Interpretation of I_{MCC} within an Epistemic Logic

The multimodal logic $\mathsf{S5}$ is among the most studied epistemic logics. It is suitable for representing and reasoning about the knowledge of agents [23]. For instance, given a propositional formula ϕ, we use the formula $\mathcal{K}_a\phi$ to represent the fact that the agent a knows ϕ. For a better understanding of our inconsistency measure, we here use the multimodal logic $\mathsf{S5}$ to provide an epistemic interpretation of I_{MCC}. In this context, we consider that each consistent piece of information is possible according to a distinct agent. Then, we show that the amount of conflicts is defined from the size of the largest consensus between all the considered agents. A possible consensus is a subset of consistent pieces of information that are not rejected by any agent. An agent rejects a subset of pieces of information if it is inconsistent with the piece of information that she/he supports.

We now provide an overview of the multimodal logic $\mathsf{S5}$. Given a countable set of agents $\mathcal{A}$, the set of the multimodal $\mathsf{S5}$ formulæ is obtained by extending the propositional language with the primitive unary modal connectives $\mathcal{K}_a$ for $a \in \mathcal{A}$. Given an $\mathsf{S5}$ formula ϕ, we use $Var(\phi)$ (resp. $A(\phi)$) to denote the set of propositional variables (resp. agents) occurring in ϕ. For instance, for $\phi = \mathcal{K}_a p \wedge \mathcal{K}_b \neg q$, we get $Var(\phi) = \{p, q\}$ and $A(\phi) = \{a, b\}$. To define the possible-worlds semantics of the multimodal logic $\mathsf{S5}$, we first define the structure of $\mathsf{S5}$-interpretation.

Definition 8 ($\mathsf{S5}$-Interpretation)**.** *An $\mathsf{S5}$-interpretation of an $\mathsf{S5}$ formula ϕ is a structure of the form $(W, \{\sim_a\}_{a\in A(\phi)}, V)$ where W is a non-empty set of worlds, V is a function from W to $2^{Var(\phi)}$ (the powerset of $Var(\phi)$), and for all $a \in A(\phi)$, $\sim_a \subseteq W \times W$ is an equivalence relation.*

Definition 9 (Satisfaction Relation)**.** *The satisfaction relation between an $\mathsf{S5}$ formula ϕ, an $\mathsf{S5}$-Interpretation $\mathcal{M} = (W, \{\sim_a\}_{a\in A(\phi)}, V)$ and a world $w \in W$, written $\mathcal{M}, w \vDash \phi$, is inductively defined as follows:*

$\mathcal{M}, w \vDash p$ *iff* $p \in V(w)$*;*

$\mathcal{M}, w \vDash \phi \wedge \psi$ *iff* $\mathcal{M}, w \vDash \phi$ *and* $\mathcal{M}, w \vDash \psi$*;*

$\mathcal{M}, w \vDash \phi \vee \psi$ *iff* $\mathcal{M}, w \vDash \phi$ *or* $\mathcal{M}, w \vDash \psi$*;*

$\mathcal{M}, w \vDash \neg\phi$ *iff* $\mathcal{M}, w \nvDash \phi$*;*

$\mathcal{M}, w \vDash \mathcal{K}_a\phi$ *iff* $\forall w' \in W$*, if* $w \sim_a w'$ *then* $\mathcal{M}, w' \vDash \phi$*.*

Definition 10 ($\mathsf{S5}$ Satisfiability Problem)**.** *Given an $\mathsf{S5}$ formula ϕ, determine whether there exists an $\mathsf{S5}$-interpretation $\mathcal{M} = (W, \{\sim_a\}_{a\in A(\phi)}, V)$ and*

a world $w \in W$ such that $\mathcal{M}, w \vDash \phi$. If $\mathcal{M}$ satisfies ϕ, we say that $\mathcal{M}$ is an $\mathsf{S5}$*-model of ϕ.*

To describe our interpretation of I_{MCC}, we associate to each consistent formula in a knowledge base a distinct agent which considers this formula as possible (she/he does not know its negation). In other words, each consistent piece of information is possible according to its associated agent. More precisely, given a knowledge base K, we associate to each formula $\phi \in K \setminus \mathsf{IF}(K)$ a distinct agent a_ϕ, and we use the following $\mathsf{S5}$ formula, denoted $\mathcal{IS}(K)$ ($\mathcal{IS}$ stands for *Initial State*), to express that each consistent formula is possible according to its associated agent:

$$\bigwedge_{\phi \in K \setminus \mathsf{IF}(K)} \neg \mathcal{K}_{a_\phi} \neg \phi$$

It is worth noting that each agent consider her/his corresponding formula possible but not necessarily known. Indeed, using $\mathcal{K}_{a_\phi}\phi$ instead of $\neg \mathcal{K}_{a_\phi} \neg \phi$ means that we require that each agent knows his corresponding piece of information, which is impossible in the case of an inconsistent knowledge base since there is no world that satisfies all its formulæ, i.e., there is no common knowledge state. Thus, the formula $\mathcal{IS}(K)$ is used to be able to have a common knowledge state between all the agents, which can be seen as a consensus.

Then, given a knowledge base K and $K' \subseteq K$, we use $\mathcal{PC}(K, K')$ ($\mathcal{PC}$ stands for *Possible Consensus*) to denote the following formula:

$$\mathcal{IS}(K) \wedge \bigwedge_{\phi \in K \setminus \mathsf{IF}(K)} \mathcal{K}_{a_\phi} \bigwedge_{\psi \in K'} \psi$$

The formula $\mathcal{PC}(K, K')$ is used to represent the fact that all the agents that satisfy $\mathcal{IS}(K)$ know all the formulæ in K'. We say that K' is a *possible consensus* in K if $\mathcal{PC}(K, K') \nvdash \bot$. We use $\mathcal{PCS}(K)$ ($\mathcal{PCS}$ stands for *Possible Consensus Set*) to denote the set of all possible consensus in K, i.e., $\mathcal{PCS}(K) = \{K' \subseteq K \mid \mathcal{PC}(K, K') \nvdash \bot\}$.

Intuitively, the consistency of $\mathcal{PC}(K, K')$ means that there exists a scenario (an $\mathsf{S5}$-interpretation) where all the agents know the pieces of information in K', knowing that each agent considers his corresponding piece of information possible. In other words, K' can be seen as a possible common knowledge state between all the agents.

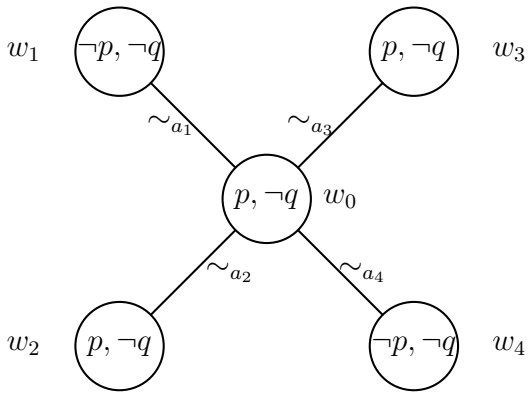


Figure 1: An S5-interpretation

Example 3. *Let us consider the knowledge base* $K = \{p \rightarrow q, p \wedge \neg q, \neg q, \neg p\}$. *Then, we associate to each formula in* K *a distinct agent:* a_1, a_2, a_3 *and* a_4 *are associated to* $p \rightarrow q$, $p \wedge \neg q$, $\neg q$ *and* $\neg p$ *respectively. Let* $K' = \{\neg q\}$ *be a subset of* K. *Then, we get* $\mathcal{PC}(K, K') = (\neg \mathcal{K}_{a_1} \neg (p \rightarrow q)) \wedge (\neg \mathcal{K}_{a_2} \neg (p \wedge \neg q)) \wedge (\neg \mathcal{K}_{a_3} \neg (\neg q)) \wedge (\neg \mathcal{K}_{a_4} \neg (\neg p)) \wedge (\mathcal{K}_{a_1} \neg q) \wedge (\mathcal{K}_{a_2} \neg q) \wedge (\mathcal{K}_{a_3} \neg q) \wedge (\mathcal{K}_{a_4} \neg q)$. *The* S5*-interpretation* $\mathcal{M}$ *described in Figure 1 is an* S5*-model of the formula* $\mathcal{PC}(K, K')$. *Indeed, following the* S5*-interpretation* $\mathcal{M}$, *the piece of information* $\neg q$ *is known by the four agents, and* a_i *considers its associated piece of information possible in the world* w_i *for* $i = 1, 2, 3, 4$. *More precisely, we have* $\mathcal{M}, w_0 \models (\mathcal{K}_{a_1} \neg q) \wedge (\mathcal{K}_{a_2} \neg q) \wedge (\mathcal{K}_{a_3} \neg q) \wedge (\mathcal{K}_{a_4} \neg q)$ *since* $\neg q$ *is satisfied in every world. Moreover,* $\mathcal{M}, w_0 \models \neg \mathcal{K}_{a_1} \neg (p \rightarrow q)$, $\mathcal{M}, w_0 \models \neg \mathcal{K}_{a_2} \neg (p \wedge \neg q)$, $\mathcal{M}, w_0 \models \neg \mathcal{K}_{a_3} \neg (\neg q)$ *and* $\mathcal{M}, w_0 \models \neg \mathcal{K}_{a_4} \neg (\neg p)$ *hold since* $\mathcal{M}, w_1 \models \neg p$, $\mathcal{M}, w_2 \models p \wedge \neg q$, $\mathcal{M}, w_3 \models \neg q$ *and* $\mathcal{M}, w_4 \models \neg p$ *respectively. As a consequence, we get* $\mathcal{M}, w_0 \models \mathcal{PC}(K, K')$. *Thus,* K' *is a possible consensus.*

In the following theorem, we show that I_{MCC} can be defined from the set of possible consensus between the considered agents. More precisely, given a knowledge base K, we show that $\mathsf{I}_{MCC}(K)$ is the number of formulæ that are not in one of the largest possible consensus. Theorem 1 shows that our inconsistency measure is directly related to the notion of possible consensus described previously.

Theorem 1. *Given a knowledge base* K, $\mathsf{I}_{MCC}(K) = |K| - max\{|K'| \mid K' \in \mathcal{PCS}(K)\}$ *holds.*

Proof. Using the definition of I_{MCC}, it suffices to show that $\lambda(K) = max\{|K'| \mid K' \in \mathcal{PCS}(K)\}$. Let us first show that $\lambda(K) \leqslant max\{|K'| \mid K' \in \mathcal{PCS}(K)\}$. Let $\mathcal{C} = \{M_1, \ldots, M_l\}$ be a maximum MCS-cover of K and $\mathcal{B}$ a model of $\bigcap_{1 \leqslant i \leqslant l} M_i$. In order to associate an $\mathsf{S5}$-interpretation that satisfies the formula $\mathcal{PC}(K, \bigcap_{1 \leqslant i \leqslant l} M_i)$, we associate two distinct worlds w_ϕ and w'_ϕ to every element ϕ in $K \setminus \mathsf{IF}(K)$. Then, we associate to $\mathcal{C}$ an $\mathsf{S5}$-interpretation $\mathcal{M} = (W, \{\sim_{a_\phi}\}_{\phi \in K \setminus \mathsf{IF}(K)}, V)$, where $W = \bigcup_{\phi \in K \setminus \mathsf{IF}(K)} \{w_\phi, w'_\phi\}$, and for all $\phi \in K \setminus \mathsf{IF}(K)$, we have:

- $V(w_\phi) = \{p \mid \mathcal{B}'(p) = 1\}$ with $\mathcal{B}'$ a model of ϕ,
- $V(w'_\phi) = \{p \mid \mathcal{B}(p) = 1\}$, and
- $\sim_{a_\phi} = \{(w_\phi, w_\phi), (w'_\phi, w'_\phi), (w_\phi, w'_\phi), (w'_\phi, w_\phi)\}$.

Clearly, for all $\phi \in K \setminus \mathsf{IF}(K)$, we get $\mathcal{M}, w_\phi \vDash \phi$ and $\mathcal{M}, w'_\phi \vDash \bigcap_{1 \leqslant i \leqslant l} M_i$, since $V(w_\phi) = \{p \mid \mathcal{B}'(p) = 1\}$ with $\mathcal{B}'$ a model of ϕ and $V(w'_\phi) = \{p \mid \mathcal{B}(p) = 1\}$ respectively. As a consequence, $\mathcal{M}$ is an $\mathsf{S5}$-model of $\mathcal{PC}(K, \bigcap_{1 \leqslant i \leqslant l} M_i)$. Thus, we get $\lambda(K) \leqslant max\{|K'| \mid K' \in \mathcal{PCS}(K)\}$.

Let us now show $\lambda(K) \geqslant max\{|K'| \mid K' \in \mathcal{PCS}(K)\}$. Let $K' \in \mathcal{PCS}(K)$ and $\mathcal{M} = (W, \{\sim_{a_\phi}\}_{\phi \in K \setminus \mathsf{IF}(K)}, V)$ an $\mathsf{S5}$-model of $\mathcal{PC}(K, K')$. Then, for all $\phi \in K \setminus \mathsf{IF}(K)$, there exists a world $w_\phi \in W$ such that $\mathcal{M}, w_\phi \vDash \phi \wedge \bigwedge_{\psi \in K'} \psi$. Thus, $\bigcup_{\phi \in K \setminus \mathsf{IF}(K)} \{K' \cup \{\phi\}\}$ is a set of consistent subsets of K. Therefore, there exists an MCS-cover $\mathcal{C}$ of K such that, for all $\phi \in K \setminus \mathsf{IF}(K)$, there exists $M \in \mathcal{C}$ such that $K' \cup \{\phi\} \subseteq M$. We get $|\bigcap_{M \in \mathcal{C}} M| \geqslant |K'|$ and, as a consequence, $\lambda(K) \geqslant |K'|$. □

5. On Properties of I_{MCC}

In the section, we show that I_{MCC} satisfies several reasonable properties. We first show that it satisfies Consistency, Monotonicity and Free-Formula-Independence. Then, we show that our measure satisfies Weak-Dominance and a stronger variant of Super-Additivity. We recall, as stated in Proposition 3, that there is no inconsistency measure that satisfies Consistency, Weak-Dominance and Super-Additivity. This can be seen as a reason explaining why our inconsistency measure does not satisfy Dominance.

Proposition 8. *I_{MCC} measure satisfies Consistency, Monotonicity and Free-Formula-Independence.*

Proof.
Consistency. We first consider the "if part". Let K be a consistent knowledge base. Then, $\{K\}$ is the unique maximum MCS-cover of K ($\lambda(K) = |K|$). Hence, $\mathsf{I}_{MCC}(K) = 0$ holds. We now consider the "only if part". Let K be a knowledge base such that $\mathsf{I}_{MCC}(K) = 0$. Then, $|K| - \lambda(K) = 0$ holds. Thus, we get $\lambda(K) = |K|$ and, as a consequence, K is consistent.
Monotonicity. Proposition 10 implies that I_{MCC} satisfies Monotonicity.
Free-Formula-Independence. Let K be a knowledge base and ϕ a free formula in K. Let $\mathcal{C} = \{M_1, \ldots, M_n\}$ be a maximum MCS-cover of $K \setminus \{\phi\}$. Since $\phi \in \mathsf{Free}(K)$, $M_i \cup \{\phi\} \nvdash \bot$ holds for every $1 \leqslant i \leqslant n$. Thus, $\{M_1 \cup \{\phi\}, \ldots, M_n \cup \{\phi\}\}$ is a maximum MCS-cover of K and we obtain $\lambda(K) = \lambda(K \setminus \{\phi\}) + 1$. As a consequence, $\mathsf{I}_{MCC}(K) = |K| - \lambda(K) = |K \setminus \{\phi\}| + 1 - \lambda(K \setminus \{\phi\}) - 1 = \mathsf{I}_{MCC}(K \setminus \{\phi\})$ holds. □

Proposition 9. *I_{MCC} measure satisfies Weak-Dominance.*

Proof. Let K be a knowledge base and ϕ and ψ two formulæ s.t. $\phi \nvdash \bot$, $\phi \notin K$ and $\phi \vdash \psi$. Let $\mathcal{C}$ be a maximum MCS-cover of $K \cup \{\phi\}$. We consider w.l.o.g. that $\psi \notin K$, since I_{MCC} satisfies Monotonicity. Clearly, by replacing in $\mathcal{C}$ the formula ϕ with ψ we obtain a set of satisfiable subsets of $K \cup \{\psi\}$. As a consequence, we get $\lambda(K \cup \{\psi\}) \geqslant \lambda(K \cup \{\phi\})$. Thus, $\mathsf{I}_{MCC}(K \cup \{\psi\}) = |K \cup \{\psi\}| - \lambda(K \cup \{\psi\}) \leqslant |K \cup \{\phi\}| - \lambda(K \cup \{\phi\}) = \mathsf{I}_{MCC}(K \cup \{\phi\})$ holds. □

Let us now provide an example that shows that I_{MCC} does not satisfy Dominance. Consider for instance the knowledge base $K = \{p \wedge (p \rightarrow q), \neg q\}$. We have $p \wedge (p \rightarrow q) \nvdash \bot$, $p \wedge (p \rightarrow q) \vdash q$ and $\lambda(K) = 0$. Moreover, $\lambda(K \cup \{q\}) = 0$ holds. Then, we have $\mathsf{I}_{MCC}(K \cup \{q\}) = 3 > \mathsf{I}_{MCC}(K \cup \{p \wedge (p \rightarrow q)\}) = 2$. In other words, Dominance fails in this case because it sates that we can add q to K without changing the amount of conflicts.

Proposition 10. *I_{MCC} measure satisfies Super-Additivity.*

Proof. A direct consequence of Proposition 11. □

In the following proposition, we show that I_{MCC} measure satisfies a property generalizing Super-Additivity.

Proposition 11 (Generalized Super-Additivity)**.** *Given two knowledge bases K and K', we have:*

$$\mathsf{I}_{MCC}(K \cup K') \geqslant \mathsf{I}_{MCC}(K) + \mathsf{I}_{MCC}(K') - |K \cap K'|$$

Proof. We have, for all $M \in \mathsf{MC}(K \cup K')$, $M' = M \cap K$ and $M'' = M \cap K'$ are both consistent. Let $\mathcal{C} = \{M_1, \ldots, M_n\}$ be a maximum MCS-cover of $K \cup K'$. Then, $\mathcal{C}' = \{M_1 \cap K, \ldots, M_n \cap K\}$ and $\mathcal{C}'' = \{M_1 \cap K', \ldots, M_n \cap K'\}$ are sets of consistent subsets. Moreover, $\bigcap_{M \in \mathcal{C}} M = (\bigcap_{M' \in \mathcal{C}'} M') \cup (\bigcap_{M'' \in \mathcal{C}''} M'')$. Thus, $\lambda(K \cup K') \leqslant \lambda(K) + \lambda(K')$ holds. As a consequence, $\mathsf{I}_{MCC}(K \cup K') \geqslant |K \cup K'| - \lambda(K) - \lambda(K')$ holds. Moreover, we have $|K \cup K'| = |K| + |K'| - |K \cap K'|$. Therefore, $\mathsf{I}_{MCC}(K \cup K') \geqslant \mathsf{I}_{MCC}(K) + \mathsf{I}_{MCC}(K') - |K \cap K'|$ holds. □

It is worth noting that I_{MCC} does not satisfy MI-Additivity, since there is no inconsistency measure that satisfies Consistency, Weak-Dominance and MI-Additivity (Proposition 5). To illustrate this point, consider for instance $K = \{p, q, \neg p \wedge \neg q\}$, $K_1 = \{p, \neg p \wedge \neg q\}$ and $K_2 = \{q, \neg p \wedge \neg q\}$. It is easy to see that $\mathsf{I}_{MCC}(K) = 3$, $\mathsf{I}_{MCC}(K_1) = 2$ and $\mathsf{I}_{MCC}(K_2) = 2$. We have $\mathsf{MI}(K) = \mathsf{MI}(K_1) \uplus \mathsf{MI}(K_2)$, but $\mathsf{I}_{MCC}(K) < \mathsf{I}_{MCC}(K_1) + \mathsf{I}_{MCC}(K_2)$.

In the following proposition, we provide interesting lower and upper bounds for I_{MCC}.

Proposition 12. *Given a knowledge base K, $|\mathsf{IF}(K)| \leqslant \mathsf{I}_{MCC}(K) \leqslant |K| - |\mathsf{Free}(K)|$.*

Proof. The inequality $\mathsf{I}_{MCC}(K) \leqslant |K| - |\mathsf{Free}(K)|$ is a direct consequence of the fact that, for all $M \in \mathsf{MC}(K)$, $\mathsf{Free}(K) \subseteq M$. Moreover, the inequality $|\mathsf{IF}(K)| \leqslant \mathsf{I}_{MCC}(K)$ is a direct consequence of the fact that, for all $M \in \mathsf{MC}(K)$, $\mathsf{IF}(K) \cap M = \emptyset$. □

In the same way as the property MinInc introduced in [1], the following proposition expresses that all the minimal inconsistent subsets are considered equally.

Proposition 13. *Given a minimal inconsistent set of formulæ K such that $|K| > 1$, we have $\mathsf{I}_{MCC}(K) = 2$.*

Proof. Let $K = \{\phi_1, \ldots, \phi_n\}$ be a minimal inconsistent set such that $n > 1$. Then, $M = \{\phi_1, \ldots, \phi_{n-1}\}$ and $M' = \{\phi_2, \ldots, \phi_n\}$ are MCSes of K, and $\{S, S'\}$ are an MCS-cover of K. Then, $\mathsf{I}_{MCC}(K) \leqslant n - (n-2) = 2$ holds. Let us assume that $\mathsf{I}_{MCC}(K) = 1$. Then, there exist M and M' in $\mathsf{MC}(K)$ such that $M \neq M'$ and $|M \cap M'| \geqslant n - 1$. Thus, $|S| = |S'| = n$ holds and we get a contradiction. Therefore, we obtain $\mathsf{I}_{MCC}(K) = 2$. □

6. On Computing I_{MCC}

In this section, we describe two modeling approaches for I_{MCC} computation. On one hand, we propose an integer linear programming model that is mainly defined from the set of MCSes. On the other hand, we propose a Partial Max-SAT model, which allows us to avoid the computation of the set of MCSes.

6.1. An Integer Linear Programming Formulation

We here show that the problem of I_{MCC} computation can be formulated as an integer linear program (ILP) by providing an encoding defined mainly from the set of the MCSes of a knowledge base. To do this, each variable used in our encoding is binary (a 0-1 variable) and corresponds to either a formula or an MCS. The constraints are defined so that the objective consists in maximizing the function corresponding to the sum of the variables associated to formulæ.

Variables. We associate a binary variable X_ϕ having as domain $\{0, 1\}$ to each consistent formula ϕ in K. We also associate a binary variable Y_M having as domain $\{0, 1\}$ to each MCS M of K.

The integer linear program ILP-MCC(K) is defined as follows:

$$\begin{array}{c} minimize \ \ |K| - \sum_{\phi \in K \setminus \mathsf{IF}(K)} X_\phi \\ \text{subject to:} \end{array}$$

$$\sum_{M \in \mathsf{MC}(K): \phi \in M} Y_M \geqslant 1 \ for\ all\ \phi \in K \setminus \mathsf{IF}(K) \tag{1}$$

$$X_\phi + Y_M \leqslant 1 \ for\ all\ \phi \in K\ and\ M \in \mathsf{MC}(K)\ with\ \phi \notin M \tag{2}$$

Intuitively, the linear inequality (1) allows us to consider the subsets of MCSes that cover all the consistent formulæ of K. Then, the linear inequality (2) says that if $X_\phi = 1$ then ϕ is a formula shared between all the considered MCSes. The objective function aims at maximizing the number of shared formulæ between the considered MCSes.

Proposition 14 (Soundness)**.** *Given a knowledge base K and a solution S of ILP-MCC(K), then $\mathsf{I}_{MCC}(K) = |K| - |\{\phi \in K \mid S(X_\phi) = 1\}|$.*

Proof. Each solution S_1 of the linear inequality (1) means that the set $\mathcal{C} = \{M \in \mathsf{MC}(K) \mid S_1(Y_M) = 1\}$ is an MCS-cover of K. Moreover, each solution S_2 of the linear inequality (2) means that $\{\phi \in K \mid S_2(X_\phi) = 1\} \subseteq \bigcap_L M$ where $L = \{M \in \mathsf{MC}(K) \mid S_2(Y_M) = 1\}$. Thus, since minimizing $|K| - \sum_{\phi \in K \setminus \mathsf{IF}(K)} X_\phi$ corresponds to maximizing $\sum_{\phi \in K \setminus \mathsf{IF}(K)} X_\phi$, we have $\lambda(K) = |\{\phi \in K \mid S(X_\phi) = 1\}|$. As a consequence, $\mathsf{I}_{MCC}(K) = |K| - |\{\phi \in K \mid S(X_\phi) = 1\}|$ holds. □

6.2. A Partial Max-SAT Formulation

We here propose a Partial Max-SAT encoding of the problem of I_{MCC} computation, which is of polynomial size and defined directly from the knowledge base without computing its MCSes.

Partial MAX-SAT is an optimization problem such that each clause is either relaxable (soft) or non-relaxable (hard). The objective is to find an interpretation that satisfies all the hard clauses together with the maximum number of soft clauses (e.g. [24]).

In the following proposition, we show that it suffices to consider a linear-bounded number of MCSes of a knowledge base for computing I_{MCC}.

Proposition 15. *Given a knowledge base K, if $\mathcal{C}$ is a maximum MCS-cover of K, then $|\mathcal{C}| \leqslant |K \setminus \mathsf{IF}(K)|$.*

Proof. Let $\mathcal{C} = \{M_1, \ldots, M_k\}$ be a maximum MCS-cover of K. Assume that $k > n$ where $n = |K \setminus \mathsf{IF}(K)|$. If we consider that each MCS in $\mathcal{C}$ contains a formula that belongs only to it, we get a contradiction, since we have $k > n$. Then, there exists $1 \leqslant i \leqslant k$ such that $M_i \subseteq \bigcup_{1 \leqslant j \leqslant n, j \neq i} M_j$. However, $\mathcal{C}$ is a normal MCS-cover (see Definition 5 and Definition 6). Thus, we get a contradiction and deduce that $k \leqslant n$. □

Using Proposition 15, we know that the value $\lambda(K)$ of a knowledge base K can be redefined as the maximum number of formulæ shared between $n = |K \setminus \mathsf{IF}(K)|$ consistent subsets $\{S_1, \ldots, S_n\}$ of K where $\bigcup_{i=1}^{n} S_i = K \setminus \mathsf{IF}(K)$. Below we introduce a Partial Max-SAT model that encodes $\lambda(K)$ using this definition.

Let $K = \{\phi_1, \ldots, \phi_n\}$ be a knowledge base. We assume w.l.o.g. that K does not contain any inconsistent formula. Indeed, to define our model, it

suffices to consider the subset of the consistent formulæ of a knowledge base.

Variables. For each propositional variable p appearing in K and $1 \leqslant j \leqslant n$, we introduce a fresh propositional variable p^j. Then, for all $1 \leqslant i, j \leqslant n$, we use ϕ_i^j to denote the formula obtained from ϕ_i by renaming each propositional variable p with its j^{th} corresponding fresh variable p^j. Moreover, for all $1 \leqslant i, j \leqslant n$, we introduce a fresh variable q_i^j, which is used to represent the fact that ϕ_i belongs to the j^{th} consistent subset. Further, for all $1 \leqslant i \leqslant n$, we introduce a fresh propositional variable r_i, which is used to represent the fact that ϕ_i is shared by all the n consistent subsets.

Hard part. We first provide the formula stating that q_i^j is true if and only if its corresponding formula ϕ_i^j is true (ϕ_i belongs to the j^{th} consistent subset):

$$\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} q_i^j \leftrightarrow \phi_i^j \tag{3}$$

We then provide the formula encoding the fact that each formula in K belongs to at least one consistent subset:

$$\bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} q_i^j \tag{4}$$

Finally, the following formula relates the truth values of the variables of the form r_i to the formulæ shared between the n consistent subsets:

$$\bigwedge_{i=1}^{n} (r_i \leftrightarrow \bigwedge_{j=1}^{n} q_i^j) \tag{5}$$

Let us note that we do not use the conjunctive normal form in our definition of the hard part. However, using linear Tseitin's encoding [22], we know that every propositional formula can be translated to CNF.

Soft part. To maximize the number of shared formulæ between the n consistent subsets, we only need the following unary soft clauses:

$$r_1, \ldots, r_n \tag{6}$$

Given a knowledge base K, $MSAT(K)$ is used to denote the Partial Max-SAT encoding described above.

Proposition 16 (Soundness)**.** *Given a knowledge base K and a solution $\mathcal{B}$ of $MSAT(K \setminus \mathsf{IF}(K))$, then $\lambda(K) = |\{r_i \mid 1 \leqslant i \leqslant |K \setminus \mathsf{IF}(K)| \text{ and } \mathcal{B}(r_i) = 1\}|$.*

Proof. Let $K = \{\phi_1, \ldots, \phi_n, \psi_1, \ldots \psi_m\}$ be a knowledge base s.t. $\psi_1, \ldots \psi_m$ are its inconsistent formulæ. Every Model $\mathcal{B}$ of the hard part of $MSAT(K \setminus \mathsf{IF}(K))$ $((3) \wedge (4) \wedge (5))$ represents a set $\{S_1, \ldots, S_n\}$ of $n = |K \setminus \mathsf{IF}(K)|$ consistent subsets of K. Then, using the formula (5), we know that $\mathcal{B}(r_i) = 1$ iff $\phi_i \in \bigcap_{i=1}^{n} S_i$. As a consequence, since the objective is to find an interpretation that satisfies the hard part together with the maximum number of soft clauses in (6), we know that $|\{r_i \mid 1 \leqslant i \leqslant n \text{ and } \mathcal{B}(r_i) = 1\}|$ is the the maximum number of formulæ that can be shared between n consistent subsets $\{S_1, \ldots, S_n\}$ of K where $\bigcup_{i=1}^{n} S_i = K \setminus \mathsf{IF}(K)$. Therefore, $\lambda(K) = |\{r_i \mid 1 \leqslant i \leqslant n \text{ and } \mathcal{B}(r_i) = 1\}|$ holds. □

7. Related Inconsistency Measures

In this section, we study relationships between our inconsistency measure and two existing ones. We first consider the MIS based inconsistency measure I_{CC} [21] described in Section 3.2. Comparing I_{MCC} and I_{CC} is motivated by the fact that they satisfy both several fundamental property, namely Consistency, Monotonicity, Free-Formula-Independence, Weak-Dominance and Super-Additivity (see Proposition 4 and Propositions 8-10) They also satisfy the fundamental property introduced in [21], called Independent-MI-Additivity[6]. The aim of this property is to distinguish the inconsistency measures that take into account the distribution of the formulæ among the conflicts. We also consider the MCS based inconsistency measure I_M introduced in [18]. The comparison of I_{MCC} and I_M is motivated by the fact that they are both defined through the maximal consistent subsets.

Definition 11 (Independent-MI-Additivity)**.** *Let I be an inconsistency measure. Then, I satisfies Independent-MI-Additivity iff, for all knowledge bases K and K', if we have $\mathsf{MI}(K \cup K') = \mathsf{MI}(K) \uplus \mathsf{MI}(K')$ and $(\bigcup_{M \in \mathsf{MI}(K)} M) \cap (\bigcup_{M \in \mathsf{MI}(K')} M) = \emptyset$, then $I(K \cup K') = I(K) + I(K')$.*

Proposition 17. *I_{MCC} measure satisfies Independent-MI-Additivity.*

[6] In the original paper, this property is called *Enhanced Additivity.*

Proof. Let K, K_1 and K_2 be knowledge bases such that $\mathsf{MI}(K) = \mathsf{MI}(K_1) \uplus \mathsf{MI}(K_2)$ and, for all $M \in \mathsf{MI}(K_1)$ and $M' \in \mathsf{MI}(K_2)$, $M \cap M' = \emptyset$. We denote K', K'_1 and K'_2 the sets $\bigcup_{M \in \mathsf{MI}(K)} M$, $\bigcup_{M \in \mathsf{MI}(K_1)} M$ and $\bigcup_{M \in \mathsf{MI}(K_2)} M$ respectively. Let us note that $K' = K'_1 \uplus K'_2$. Using Free-Formula-Independence, we have $\mathsf{I}_{MCC}(K) = \mathsf{I}_{MCC}(K')$, $\mathsf{I}_{MCC}(K_1) = \mathsf{I}_{MCC}(K'_1)$ and $\mathsf{I}_{MCC}(K_2) = \mathsf{I}_{MCC}(K'_2)$. Then, using Super-Additivity, we have $\mathsf{I}_{MCC}(K) \geqslant \mathsf{I}_{MCC}(K_1) + \mathsf{I}_{MCC}(K_2)$. Let $M_1 \in \mathsf{MC}(K'_1)$ and $M_2 \in \mathsf{MC}(K'_2)$. Since $(\bigcup_{M \in \mathsf{MI}(K_1)} M) \cap (\bigcup_{M \in \mathsf{MI}(K_2)} M) = \emptyset$, $M_1 \cup M_2$ is a consistent set in K'. Then, using the fact that $K' = K'_1 \uplus K'_2$, we have $\lambda(K') \geqslant \lambda(K'_1) + \lambda(K'_2)$ and, consequently, $\mathsf{I}_{MCC}(K') \leqslant \mathsf{I}_{MCC}(K'_1) + \mathsf{I}_{MCC}(K'_2)$. Thus, $\mathsf{I}_{MCC}(K) \leqslant \mathsf{I}_{MCC}(K_1) + \mathsf{I}_{MCC}(K_2)$ holds. Therefore, we get $\mathsf{I}_{MCC}(K) = \mathsf{I}_{MCC}(K_1) + \mathsf{I}_{MCC}(K_2)$. □

In the following proposition, we highlight an interesting relationship between I_{MCC} and I_{CC}.

Proposition 18. *Given a knowledge base K, we have $\mathsf{I}_{MCC}(K) \geqslant 2 \times \mathsf{I}_{CC}(K)$.*

Proof. Let $S = (\{K_1, \ldots, K_n\}, K')$ be an MI-decomposition s.t. $\mathsf{I}_{CC}(K) = n$. Using the condition (iv) in the definition of MI-decomposition, we get $K_i \cap K_j = \emptyset$ for every $1 \leqslant i < j \leqslant n$. Thus, we get $\mathsf{I}_{MCC}(\bigcup_{i=1}^{n} K_i) \geqslant \sum_{i=1}^{n} \mathsf{I}_{MCC}(K_i)$ since I_{MCC} satisfies Super-Additivity. Moreover, using the condition (iii), we get $K_i \vdash \bot$ for every $1 \leqslant i \leqslant n$. As a consequence, for all $1 \leqslant i \leqslant n$, there exists $M_i \subseteq K_i$ s.t. M_i is a minimal inconsistent set. Using Proposition 13, we get $\mathsf{I}_{MCC}(M_i) = 2$ for every $1 \leqslant i \leqslant n$. Then using the fact that I_{MCC} satisfies Monotonicity, we get $\mathsf{I}_{MCC}(K_i) \geqslant 2$ for every $1 \leqslant i \leqslant n$. Thus, $\mathsf{I}_{MCC}(\bigcup_{i=1}^{n} K_i) \geqslant 2 \times n$ holds. Finally, using the fact that I_{MCC} satisfies Monotonicity, we get $\mathsf{I}_{MCC}(K' \cup \bigcup_{i=1}^{n} K_i) = \mathsf{I}_{MCC}(K) \geqslant 2 \times n$. □

We now show that I_{MCC} allows to distinguish knowledge bases which are not distinguishable by I_{CC}. Consider, for instance, the two knowledge bases $K_1 = \{p \wedge q, p \wedge r, \neg p\}$ and $K_2 = \{p \wedge q, \neg p\}$. Then, $\mathsf{MI}(K_1) = \{\{p \wedge q, \neg p\}, \{p \wedge r, \neg p\}\}$ and $\mathsf{MI}(K_2) = \{\{p \wedge q, \neg p\}\}$. Hence, we get $\mathsf{I}_{CC}(K_1) = \mathsf{I}_{CC}(K_2) = 1$. Furthermore, $\mathcal{C}_1 = \{\{p \wedge q, p \wedge r\}, \{\neg p\}\}$ and $\mathcal{C}_2 = \{\{p \wedge q\}, \{\neg p\}\}$ are maximum MCS-covers of K_1 and K_2 respectively. As a consequence, $\lambda(K_1) = \lambda(K_2) = 0$. Thus, $\mathsf{I}_{MCC}(K_1) = 3$ and $\mathsf{I}_{MCC}(K_2) = 2$ hold. The previous example shows that I_{MCC} allows to distinguish knowledge bases which are not distinguishable by I_{CC}. More generally, we formally show in the two following propositions that I_{CC} does not distinguish knowledge bases of a certain type, but I_{MCC} does.

Proposition 19. *Given a knowledge base K, if $M \cap M' \neq \emptyset$ for every $M, M' \in \mathsf{MI}(K)$ with $M \neq M'$, then $\mathsf{I}_{CC}(K) = 1$.*

Proof. This property is a consequence of the fact that the maximum number of MISes that can be isolated by removing formulæ is 1, since they share all a non-empty subset of formulæ. □

Proposition 20. *Given a knowledge base K, if (i) $\mathsf{IF}(K) = \emptyset$ and (ii) there exists a formula $\phi \in K$ s.t. $M \cap M' = \{\phi\}$ for every $M, M' \in \mathsf{MI}(K)$ with $M \neq M'$, then $\mathsf{I}_{MCC}(K) = |\mathsf{MI}(K)| + 1$.*

Proof. Using Condition (ii), we know that $K \setminus \{\phi\}$ is an MCS. Moreover, using Condition (i) and Condition (ii), we know that for each MCS M containing ϕ, $|M| = |K| - |\mathsf{MI}(K)|$. As a consequence, $\{K \setminus \{\phi\}, M\}$ is a maximum MCS-cover of K for every $M \in \mathsf{MC}(K)$ with $\phi \in M$. Thus, $\lambda(K) = |K| - |\mathsf{MI}(K)| - 1$ holds. Therefore, we get $\mathsf{I}_{MCC}(K) = |K| - \lambda(K) = |\mathsf{MI}(K)| + 1$. □

It is worth noting that the knowledge bases that are considered in Proposition 20 are particular cases of those considered in Proposition 19.

Conversely, consider the knowledge bases $K_3 = \{p \wedge q_1, p \wedge q_2, \neg p, r, \neg r\}$ and $K_4 = \{p \wedge q_1, p \wedge q_2, p \wedge q_3, p \wedge q_4, \neg p\}$. Then, $\langle\{\{p \wedge q_1, p \wedge q_2, \neg p\}, \{r, \neg r\}\}, \emptyset\rangle$ and $\langle\{\{p \wedge q_1, p \wedge q_2, p \wedge q_3, p \wedge q_4, \neg p\}\}, \emptyset\rangle$ are maximum MI-decompositions of K_3 and K_4 respectively and, consequently, $\mathsf{I}_{CC}(K_3) = 2$ and $\mathsf{I}_{CC}(K_3) = 1$ hold. Moreover, $\{\{p \wedge q_1, p \wedge q_2, r\}, \{\neg p, \neg r\}\}$ and $\{\{p \wedge q_1, p \wedge q_2, p \wedge q_3, p \wedge q_4\}, \{\neg p\}\}$ are maximum MCS-covers of K_3 and K_4 respectively. Thus, we obtain $\mathsf{I}_{MCC}(K_3) = \mathsf{I}_{MCC}(K_4) = 5$. This example shows that I_{CC} allows to distinguish knowledge bases which are not distinguishable by I_{MCC}. As a consequence, we can deduce that I_{MCC} and I_{CC} do not capture the same facets in measuring inconsistency, since the one allows us to distinguish knowledge bases that are not distinguishable by the other.

Let us now consider the inconsistency measure I_M introduced in [18]. It is defined as follows:

$$\mathsf{I}_M(K) = |\mathsf{MC}(K)| + |\mathsf{IF}(K)| - 1$$

In our comparison of I_{MCC} and I_M, we first consider the simple case of the knowledge bases containing only inconsistent formulæ. We have the following

property: for all knowledge base K with $\mathsf{IF}(K) = K$, $\mathsf{I}_{MCC}(K) = \mathsf{I}_M(K) = |K|$. Indeed, when a knowledge base K contains only inconsistent formulæ, we get $\lambda(K) = 0$ and $|\mathsf{MC}(K)| = 1$. More generally, using similar arguments, we get the following property: for all knowledge base K with $|M| = 1$ for every $M \in \mathsf{MI}(K)$, $\mathsf{I}_{MCC}(K) = \mathsf{I}_M(K) = |\mathsf{IF}(K)|$. In other words, I_{MCC} and I_M proceed in the same way in the case where the inconsistency of a knowledge base is only the consequence of the presence of inconsistent formulæ.

Furthermore, It is worth noting that, for a given knowledge base K, $\mathsf{I}_{MCC}(K) \leqslant |K|$, but $\mathsf{I}_M(K)$ may be exponential in the size of K. Consider, for instance, the following knowledge base:

$$K = \{\sum_{i=1}^{2n} p_i \geqslant n, \neg p_1, \ldots, \neg p_{2n}\}$$

The inequality in K corresponds to an instances of the well-known cardinality constraint. Several polynomial encodings of this kind of constraints into propositional formulæ have been proposed in the literature (e.g. [25, 26]). Clearly, each MCS in K is either $\{\neg p_1, \ldots, \neg p_{2n}\}$ or a subset of n formulæ in $\{\neg p_1, \ldots, \neg p_{2n}\}$ with $\sum_{i=1}^{2n} p_i \geqslant n$. As a consequence, we get $|\mathsf{MC}(K)| = 1 + \binom{2n}{n} = 1 + \frac{2n!}{n!.n!} \geqslant 2^n$.

Consider, for instance, the two knowledge bases $K_1 = \{p \wedge \neg q, q \wedge r_1\}$ and $K_2 = \{p \wedge \neg q, q \wedge r_1, \ldots, q \wedge r_n\}$ for an integer $n \geqslant 2$. In both knowledge bases, there are two disjoint maximal consistent subsets and, consequently, we get $\mathsf{I}_M(K_1) = \mathsf{I}_M(K_2) = 1$. However, we have $\mathsf{I}_{MCC}(K_1) = 2$ and $\mathsf{I}_{MCC}(K_2) = n + 1$. Except the fact that this example shows that I_{MCC} can distinguish knowledge bases that are not distinguishable using I_M, it also shows that I_{MCC} and I_M do not quantify the amount of conflicts in the same way. Indeed, unlike I_{MCC}, the inconsistency measure I_M does not take into account the distribution of formulæ among the maximal consistent subsets.

8. Conclusion and Perspectives

Several approaches for measuring inconsistency have been proposed in the literature. In this paper, we proposed an original approach based on the use of maximal consistent subsets. Using this measure, the amount of conflicts in a knowledge base is defined as the smallest number of pieces of information that are not in the intersection of MCSes covering all the consistent pieces of informations. The main idea is twofold. First, the inconsistency has to

be quantified by considering that all the consistent formulæ of a knowledge are possible. This explains why we use the notion of MCS-cover. Second, a knowledge base with MCSes sharing a lot of formulæ should be assigned a smaller value of inconsistency than a knowledge base with MCSes sharing a small number of formulæ. Intuitively, by taking into account the formulæ shared between MCSes, we aim at capturing the formulæ that are involved in a small number of conflicts. Furthermore, we proposed a multi-agent consensus based interpretation of our inconsistency measure. In this interpretation, we consider that each consistent piece of information is possible according to a distinct agent, and the amount of conflicts is defined from the size of the largest consensus between all the agents. A possible consensus is a subset of consistent pieces of information that are not rejected by any agent. We use an epistemic logic, namely the multimodal logic $\mathsf{S5}$, to describe this interpretation. We then showed that our inconsistency measure satisfies several desired state-of-the-art properties, such as Consistency, Monotonicity, Free-Formula-Independence and Super-Additivity. Moreover, we proposed an encoding in integer linear programming for its computation, which is defined from the set of maximal consistent subsets. We also proposed a polynomial Partial Max-SAT encodings, which allows us to avoid the computation of maximal consistent subsets.

As a future work, we plan to conduct experimental evaluations for the problem of I_{MCC} computation. We also intend to investigate relationships between our approach for measuring inconsistency and existing multi-agent consensus models.

References

[1] A. Hunter, S. Konieczny, On the measure of conflicts: Shapley inconsistency values, Artificial Intelligence 174 (14) (2010) 1007–1026.

[2] J. Grant, Classifications for inconsistent theories, Notre Dame Journal of Formal Logic 19 (3) (1978) 435–444.

[3] K. Knight, Measuring inconsistency, J. Philosophical Logic 31 (1) (2002) 77–98.

[4] S. Konieczny, J. Lang, P. Marquis, Quantifying information and contradiction in propositional logic through test actions, in: IJCAI-03, Pro-

ceedings of the 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico, Morgan Kaufmann, 2003, pp. 106–111.

[5] G. Qi, W. Liu, D. A. Bell, Measuring conflict and agreement between two prioritized belief bases, in: IJCAI-05, Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, Professional Book Center, 2005, pp. 552–557.

[6] K. Mu, W. Liu, Z. Jin, A general framework for measuring inconsistency through minimal inconsistent sets, Knowledge and Information Systems 27 (1) (2011) 85–114.

[7] J. Grant, A. Hunter, Distance-based measures of inconsistency, in: Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 12th European Conference, ECSQARU 2013, Utrecht, The Netherlands, Vol. 7958 of Lecture Notes in Computer Science, Springer, 2013, pp. 230–241.

[8] A. Hunter, S. Parsons, M. Wooldridge, Measuring inconsistency in multi-agent systems, Kunstliche Intelligenz 28 (2014) 169–178.

[9] Q. Chen, C. Zhang, S. Zhang, A verification model for electronic transaction protocols, in: Advanced Web Technologies and Applications, 6th Asia-Pacific Web Conference, APWeb 2004, Hangzhou, China, Vol. 3007 of Lecture Notes in Computer Science, Springer, 2004, pp. 824–833.

[10] A. B. Barragans-Martinez, J. Pazos-Arias, A. Fernandez-Vilas, On measuring levels of inconsistency in multi-perspective requirements specifications, in: Proceedings of the first International Conference on the Principles of Software Engineering, PRISE-04, Buenos Aires, Argentina, 2004, pp. 21–30.

[11] A. Hunter, How to act on inconsistent news: Ignore, resolve, or reject, Data & Knowledge Engineering 57 (3) (2006) 221–239.

[12] J. Grant, A. Hunter, Measuring inconsistency in knowledgebases, Journal of Intelligent Information Systems 27 (2) (2006) 159–184.

[13] M. V. Martinez, A. Pugliese, G. I. Simari, V. S. Subrahmanian, H. Prade, How dirty is your relational database? An axiomatic approach, in: Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 9th European Conference, ECSQARU 2007, Hammamet,

Tunisia, Vol. 4724 of Lecture Notes in Computer Science, Springer, 2007, pp. 103–114.

[14] L. Zhou, H. Huang, G. Qi, Y. Ma, Z. Huang, Y. Qu, Measuring inconsistency in DL-Lite ontologies, in: 2009 IEEE/WIC/ACM International Conference on Web Intelligence, WI-09, Milan, Italy, IEEE Computer Society, 2009, pp. 349–356.

[15] K. McAreavey, W. Liu, P. Miller, K. Mu, Measuring inconsistency in a network intrusion detection rule set based on snort, International Journal of Semantic Computing 5 (3).

[16] P. Besnard, Revisiting postulates for inconsistency measures, in: Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Madeira, Portugal, Vol. 8761 of Lecture Notes in Computer Science, Springer, 2014, pp. 383–396.

[17] M. Thimm, Measuring inconsistency in probabilistic knowledge bases, in: UAI 2009, Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, AUAI Press, 2009, pp. 530–537.

[18] J. Grant, A. Hunter, Measuring consistency gain and information loss in stepwise inconsistency resolution, in: Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 11th European Conference, ECSQARU 2011, Belfast, UK, Vol. 6717 of Lecture Notes in Computer Science, Springer, 2011, pp. 362–373.

[19] G. Priest, Minimally inconsistent LP, Studia Logica 50 (1991) 321–331.

[20] M. Thimm, Inconsistency measures for probabilistic logics, Artificial Intelligence 197 (2013) 1–24.

[21] S. Jabbour, Y. Ma, B. Raddaoui, Inconsistency measurement thanks to MUS decomposition, in: International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, IFAAMAS/ACM, 2014, pp. 877–884.

[22] G. Tseitin, On the complexity of derivations in the propositional calculus, in: H. Slesenko (Ed.), Structures in Constructives Mathematics and Mathematical Logic, Part II, 1968, pp. 115–125.

[23] R. Fagin, J. Y. Halpern, Y. Moses, M. Y. Vardi, Reasoning About Knowledge, MIT Press Books, The MIT Press, 2003.

[24] Z. Fu, S. Malik, On solving the partial MAX-SAT problem, in: Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, Vol. 4121 of Lecture Notes in Computer Science, Springer, 2006, pp. 252–265.

[25] O. Bailleux, Y. Boufkhad, Efficient CNF encoding of boolean cardinality constraints, in: Principles and Practice of Constraint Programming - CP 2003, 9th International Conference, CP 2003, Kinsale, Ireland, Vol. 2833 of Lecture Notes in Computer Science, Springer, 2003, pp. 108–122.

[26] C. Sinz, Towards an optimal CNF encoding of boolean cardinality constraints, in: Principles and Practice of Constraint Programming - CP 2005, 11th International Conference, CP 2005, Sitges, Spain, Vol. 3709 of Lecture Notes in Computer Science, Springer, 2005, pp. 827–831.

A Constructive Argumentation Framework

S. Kaci, and Y. Salhi. A Constructive Argumentation Framework. AAAI Conference on Artificial Intelligence, AAAI 2014, AAAI Press, 1070-1076.

A Constructive Argumentation Framework

Souhila Kaci
LIRMM - UMR 5506, University of Montpellier 2
161 rue ADA F34392 Montpellier Cedex 5, France
kaci@lirmm.fr

Yakoub Salhi
CRIL - UMR 8188, University of Artois
F-62307 Lens Cedex, France
salhi@cril.fr

Abstract

Dung's argumentation framework is an abstract framework based on a set of arguments and a binary attack relation defined over the set. One instantiation, among many others, of Dung's framework consists in constructing the arguments from a set of propositional logic formulas. Thus an argument is seen as a reason for or against the truth of a particular statement. Despite its advantages, the argumentation approach for inconsistency handling also has important shortcomings. More precisely, in some applications what one is interested in are not so much only the conclusions supported by the arguments but also the precise explications of such conclusions. We show that argumentation framework applied to classical logic formulas is not suitable to deal with this problem. On the other hand, intuitionistic logic appears to be a natural alternative candidate logic (instead of classical logic) to instantiate Dung's framework. We develop *constructive argumentation framework*. We show that intuitionistic logic offers nice and desirable properties of the arguments. We also provide a characterization of the arguments in this setting in terms of minimal inconsistent subsets when intuitionistic logic is embedded in the modal logic S4.

Introduction

Argumentation theory is a reasoning process based on constructing arguments, determining conflicts between arguments and determining acceptable arguments. Dung's argumentation framework is an abstract framework based on a set of arguments and a binary attack relation defined over the set (Dung 1995). In this framework, an argument is an abstract entity whose origin and structure are not known. The role of an argument is only described by its conflicts with other arguments. The abstract nature of Dung's framework accounts for the broad range of its applications.

We distinguish between two main trends for extending/instantiating Dung's framework: those argumentation frameworks which use (as in Dung's framework) abstract arguments, e.g. (Bench-Capon 2003), and those which take into account the internal structure of the arguments, e.g., (Simari and Loui 1992; Besnard and Hunter 2008). The present paper follows the second trend. In particular we focus on the case where the arguments are built from a set of classical propositional logic formulas. Therefore an argument is seen as a reason for or against the truth of a particular statement. In particular, an argument is a pair of (1) a set of classical propositional formulas obeying some conditions, called the support of the argument, and (2) a logical formula inferred from the set, called the conclusion of the argument. Given a set of arguments (constructed from a set of classical propositional formulas), the classical treatment of argumentation framework consists in determining conflicts over the arguments and determining acceptable arguments. Conclusions supported by acceptable arguments are consistent and considered reliable. Although works around argumentation frameworks have shown great promises for reasoning with inconsistent knowledge, the argumentation approach also has important shortcomings in this setting. More precisely, in some applications what one is interested in are not so much only the conclusions supported by the acceptable arguments but also the explications of such conclusions. Unfortunately, in some situations the internal structure of the acceptable arguments is not sufficient to provide such information. Let us consider the following example to illustrate the problem:

- If John has taken his medication (M), then his situation is stable (S).
- If John has not taken his medication, then a doctor administers the medication to him (D).
- If a doctor administers the medication to John, then the situation of John is stable.

From the previous statements, we can deduce that the situation of John is stable (S) using classical reasoning. Indeed, this can be easily obtained using the law of excluded middle ($M \vee \neg M$): on one hand, if M is true then S is true. On the other hand, if $\neg M$ is true then D is true which allows us to conclude that S is true too. It goes without saying that this situation is extremely confusing as, although we know that the situation of John is stable, we are not able to provide the precise reason for such a conclusion if we don't know whether John has taken his medication or not! For instance, if John lost consciousness, we, including the doctor, cannot determine which of M or $\neg M$ is true in $M \vee \neg M$. Clearly we need an argumentation framework over a logic

which, contrary to classical logic, goes beyond the classical deduction of the conclusion of an argument from the support of the argument. This consideration calls for more sophisticated logics. In other words, what we need is not only the information from which a certain conclusion is derived but also that information which permit to construct the conclusion at hand. This consideration is formally known in the literature as "constructivism".

In mathematics, *constructivism* is a philosophical approach which consists in requiring that in order to prove that a mathematical object exists, it is necessary to be able to construct it. In this context, intuitionistic propositional logic can be seen as a formalization of constructive mathematics of Brouwer and Heyting (Beeson 1984; Troelstra and van Dalen 1988). It is defined starting from classical logic by excluding certain standard forms of reasoning, in particular, the law of excluded middle and double-negation elimination. A statement in intuitionistic logic is considered as true if we are able to provide a constructive proof, and as false if we are able to prove that it does not have a constructive proof. In a sense, the truth value of a statement is more related to our knowledge than in classical logic. For instance, the formula $M \vee \neg M$ is true in intuitionistic logic only if we provide a constructive proof of M or a constructive proof of $\neg M$, whereas this formula is true in classical logic without such proofs. Thus, if we consider the previous example, it is not possible to derive S in intuitionistic logic from the given statements.

In the setting of argumentation framework for handling inconsistent logical formulas, intuitionistic logic appears to be a suitable tool to address the above considerations. In this paper, we develop a *constructive argumentation framework* which builds on Dung's framework and intuitionistic logic.

Intuitionistic Propositional Logic

We first define the syntax of intuitionistic propositional logic (IPL for short). Let Prop be a denumerable set of propositional variables whose elements are denoted by p, q, r, etc. The symbols $\vee, \wedge, \rightarrow$ denote disjunction, conjunction and implication, respectively. The set of formulæ of IPL, denoted Form, is defined from Prop and the constant $\bot$, denoting absurdity, by using the following grammar:

$$A ::= p \mid \bot \mid A \wedge A \mid A \vee A \mid A \rightarrow A.$$

The logical connective of negation $\neg$ is defined by using $\bot$ and the connective $\rightarrow$: $\neg A \equiv A \rightarrow \bot$. The logical connective of equivalence, denoted $\leftrightarrow$, is defined as usual.

We provide here a possible world semantics, called Kripke semantics, of intuitionistic logic. In this semantics, we use a universe of worlds where each propositional variable has a truth value in each world.

Definition 1 (Kripke Model)**.** *Let W be the universe of worlds. A Kripke model is defined as a triple $(W, \preceq, V)$, where $\preceq$ is a preorder over W and $V : W \rightarrow 2^{\mathsf{Prop}}$ is an interpretation such that, for all w and w' in W with $w \preceq w'$, $V(w) \subseteq V(w')$.*

We associate to each Kripke model $\mathcal{M} = (W, \preceq, V)$ a forcing relation, denoted $\Vdash_{\mathcal{M}}$, between W and Form. It is defined by induction on formula structure as follows:

$w \Vdash_{\mathcal{M}} p$ iff $p \in V(w)$;

$w \Vdash_{\mathcal{M}} \bot$ never holds;

$w \Vdash_{\mathcal{M}} A \wedge B$ iff $w \Vdash_{\mathcal{M}} A$ and $w \Vdash_{\mathcal{M}} B$;

$w \Vdash_{\mathcal{M}} A \vee B$ iff $w \Vdash_{\mathcal{M}} A$ or $w \Vdash_{\mathcal{M}} B$;

$w \Vdash_{\mathcal{M}} A \rightarrow B$ iff, for all $w' \in W$ with $w \preceq w'$, if $w' \Vdash_{\mathcal{M}} A$ then $w' \Vdash_{\mathcal{M}} B$.

Note that $\Vdash_{\mathcal{M}}$ satisfies Kripke monotonicity property:

Proposition 1. *Let A be a formula, $\mathcal{M} = (W, \preceq, V)$ and $w, w' \in W$. If $w \Vdash_M A$ and $w \preceq w'$, then $w' \Vdash A$.*

A formula A is satisfiable in IPL if there exists a Kripke model $\mathcal{M} = (W, \preceq, V)$ and a world w in W such that $w \Vdash_M A$. A is valid in IPL if, for all Kripke model $\mathcal{M} = (W, \preceq, V)$ and for all w in W, $w \Vdash_M A$. Satisfiability and validity in IPL are Polynomial-Space Complete (Statman 1979).

For logical consequence concept, we use the turnstile symbol "$\vdash_i$" (i for intuitionistic propositional logic), i.e., $\{A_1, \ldots, A_n\} \vdash_i B$ and $(A_1 \wedge \cdots \wedge A_n) \rightarrow B$ are equivalent. Similarly to classical logic, we have the deduction property in intuitionistic logic:

Proposition 2. *Let Γ be a set of formulæ, and A and B two formulæ. Then, $\Gamma \vdash_i A \rightarrow B$ iff $\Gamma \cup \{A\} \vdash_i B$.*

Classical logic is stronger than intuitionistic logic. Therefore a valid formula in the latter is also valid in the former.

Classical vs Constructive Arguments

In this section, we first present the usage of Dung's framework for handling inconsistency in classical propositional logic knowledge bases (Besnard and Hunter 2008). We then discuss how this instantiation can or can not be directly applicable with intuitionistic propositional logic. Essentially, a logic-based argumentation framework operates in the following steps:

1. constructing arguments (in favor of/against a conclusion) from knowledge bases,
2. determining the conflicts, called an attack relation, between the arguments,
3. and determining the acceptable arguments from which justified conclusions are concluded.

Dung's argumentation framework is a pair $\langle \mathbb{A}, Att \rangle$, where $\mathbb{A}$ is the set of arguments and Att is the attack relation over $\mathbb{A} \times \mathbb{A}$. Acceptability semantics define sets of arguments that should satisfy some conditions in order to represent a justifiable point of view on the acceptance of the arguments. Due to the lack of space we do not recall these semantics and refer the reader to (Dung 1995). The notion of argument is defined on the basis of the underlying logical language and its associated logical consequence.

Classical Argumentation Framework

When Dung's argumentation framework is used to deal with inconsistent knowledge encoded in classical propositional logic, an argument is defined in the following way:

Definition 2 (Argument). *Let Γ be a set of classical propositional formulas. An argument over Γ is a pair $\mathcal{A} = \langle \Delta, C \rangle$ such that $\Delta \subseteq \Gamma$, $\Delta \nvdash_c \bot$, $\Delta \vdash_c C$ (c for classical propositional logic) and, for all $\Delta' \subset \Delta$, $\Delta' \nvdash_c C$.*

The set Δ is called the support of the argument and C its conclusion. We say that $\mathcal{A}$ is an argument for C. In this setting, $\langle \Delta, C \rangle$ is called a classical argument. For instance, consider $\Gamma = \{A \to C, \neg A \to B, B \to C\}$. The pair $\langle \Delta, C \rangle$, with $\Delta = \Gamma$, is a classical argument. Given two arguments $\langle \Delta, C \rangle$ and $\langle \Delta', C' \rangle$, we say that $\langle \Delta, C \rangle$ undercuts $\langle \Delta', C' \rangle$ iff for some $\phi \in \Delta'$, $C \equiv \neg\phi$. $\langle \Delta, C \rangle$ rebuts $\langle \Delta', C' \rangle$ iff $C \equiv \neg C'$. Then, $\langle \Delta, C \rangle$ attacks $\langle \Delta', C' \rangle$ iff $\langle \Delta, C \rangle$ rebuts/undercuts $\langle \Delta', C' \rangle$[1].

The practical computation of the arguments is far from being a straightforward problem. Given the setting of classical propositional logic and the properties of an argument, the authors of (Besnard et al. 2010) have proposed an approach based on insights from SAT problem (Grégoire, Mazure, and Piette 2009). More precisely, let $\langle \Delta, C \rangle$ be a classical argument. From Definition 2 we have that Δ is minimal. Moreover we have $\Delta \vdash_c C$ which is equivalent to $\Delta \cup \{\neg C\}$ being inconsistent, i.e., $\Delta \cup \{\neg C\} \vdash_c \bot$. Therefore $\Delta \cup \{\neg C\}$ is minimally inconsistent. The subset is inconsistent but all its proper subsets are consistent. From the previous equivalence, a classical argument is characterized in the following way (Besnard and Hunter 2008):

$\langle \Delta, C \rangle$ is a classical argument if and only if $\Delta \cup \{\neg C\}$ is a minimal inconsistent subset.

In SAT terminology, a minimal inconsistent subset is called MUS (for Minimally Unsatisfiable Subset). An algorithm is provided in (Besnard et al. 2010) to generate the set of arguments in an efficient way.

Constructive Argumentation Framework

Let us now focus our attention on our main problem, namely argumentation for intuitionistic propositional logic. We define a constructive argumentation framework as an instantiation of Dung's framework over intuitionistic propositional logic. It also operates in the three steps previously described. The main difference however consists in the logical consequence. More precisely due to the use of intuitionistic (instead of classical) propositional logic, the logical consequence $\vdash_c$ in Definition 2 is replaced with $\vdash_i$. The argument is then called constructive. The definitions of the attack relation and acceptability semantics remain unchanged.

Having defined what constructive argumentation framework is, we naturally come to the question "How do classical and constructive argumentation frameworks relate to each other?" While the definition of the attack relation and acceptability semantics are identical, we will show that the notion of argument and its characterization with a minimally inconsistent subset fall down in the setting of intuitionistic propositional logic. For this purpose, readers need not to have a strong background on IPL. What they need to know in this section are the following properties[2]: *(i)* $\{\neg\neg A\} \vdash_i A$ does not hold, *(ii)* $A \to \neg\neg A$ is a theorem of IPL and *(iii)* $A \vee \neg A$ is not a theorem of IPL.

With this in mind, we can first state that:

a classical argument is not necessarily a constructive argument.

For example $\langle \{A \to C, \neg A \to B, B \to C\}, C \rangle$ is a classical argument but not a constructive one.

Moreover the characteristic property of a classical argument stating that $\langle \Delta, C \rangle$ is a classical argument if and only if $\Delta \cup \{\neg C\}$ is a minimal inconsistent subset does not hold with constructive arguments. For instance, the set $\{\neg\neg p, \neg p\}$ is a minimal inconsistent subset, but $\langle \{\neg\neg p\}, p \rangle$ is not a constructive argument. This is because of $\{\neg\neg p\} \nvdash_i p$. Indeed, the formula $\neg\neg p \to p$ has as a counter-model $\mathcal{M} = (\{w, w'\}, \preceq, V)$ with $w \preceq w'$, $V(w) = \emptyset$ and $V(w') = \{p\}$. In this Kripke model, we have $w \vDash_{\mathcal{M}} \neg\neg p$, since $w' \vDash_{\mathcal{M}} p$, and $w \nvDash_{\mathcal{M}} p$.

However, we have the following property:

Proposition 3. *If $\langle \Delta, C \rangle$ is a constructive argument, then $\Delta \cup \{\neg C\}$ is an inconsistent subset.*

Proof. We have $\Delta \vdash_i C$, since the pair $\langle \Delta, C \rangle$ is a constructive argument. Moreover, we have $C \vdash_i \neg\neg C$ ($C \to \neg\neg C$ is a valid formula in IPL). Thus, using $\Delta \vdash_i C$ and $C \vdash_i \neg\neg C$, we obtain $\Delta \vdash_i \neg\neg C$. From Prop. 2, we deduce that $\Delta \cup \{\neg C\}$ is an inconsistent set, since $\neg\neg C \equiv \neg C \to \bot$. □

The inconsistent subset in the previous proposition is not necessarily minimal. Indeed, the pair $\langle \{\neg\neg p, \neg\neg p \to p\}, p \rangle$ is a constructive argument, but $\{\neg\neg p, \neg\neg p \to p, \neg p\}$ is not a minimal inconsistent set because $\{\neg\neg p, \neg p\}$ is an inconsistent set.

Nonetheless, constructive arguments can be characterized by means of minimal inconsistent subsets when the conclusion of the argument has the form $\neg C$. Formally, we have:

Proposition 4. *The pair $\langle \Delta, \neg C \rangle$ is a constructive argument iff $\Delta \cup \{C\}$ is a minimal inconsistent subset.*

Proof.
Part $\Rightarrow$. Using Prop. 2, we have if $\Delta \vdash_i \neg C$ then $\Delta \cup \{C\} \vdash_i \bot$. Hence, $\Delta \cup \{C\}$ is an inconsistent subset. If $\Delta \cup \{C\}$ is not a minimal inconsistent subset then there exists $\Delta' \subset \Delta$ such that $\Delta' \vdash_i \neg C$. We get a contradiction since Δ is minimal from Def. 2, i.e., no proper subset of Δ deduces $\neg C$. Therefore, $\Delta \cup \{C\}$ is a minimal inconsistent subset.
Part $\Leftarrow$. If $\Delta \cup \{C\}$ is a minimal inconsistent subset, then we have $\Delta \nvdash_i \bot$, $\Delta \vdash_i \neg C$ (Prop. 2) and, for all $\Delta' \subset \Delta$, $\Delta' \nvdash_i \neg C$. Thus, the pair $\langle \Delta, \neg C \rangle$ is a constructive argument. □

Proposition 4 comes from the fact that constructing the negation of a formula corresponds to having a contradiction ($\bot$) from this formula considered as an hypothesis, since $\neg C$ is seen as the formula $C \to \bot$ in IPL.

Lastly, let us emphasize that a constructive argument can be seen as a classical argument augmented with additional

[1] In some argumentation systems, called preference-based argumentation frameworks, the attack relation is derived from a conflict relation and a preference relation over the arguments. However this is not the main focus in this paper.

[2] A formal exposition of IPL and its consequence in argumentative reasoning will be given in the next section.

information that allow us to comply with the principles of constructivism.

Proposition 5. *If $\langle \Delta, C\rangle$ is a constructive argument, then there exists $\Delta' \subseteq \Delta$ s.t. $\langle \Delta', C\rangle$ is a classical argument.*

The proof of the previous proposition relies on the fact that each valid formula in IPL is also valid in classical logic.

Besides the fact that constructive arguments provide a way to construct a given conclusion, they also prevent some undesirable justifications when classical arguments are dealt with. By undesirability we mean here that the justification is misleading. In order to illustrate this point, we consider a simple medical diagnostic framework. We use rules of the form $S_1 \wedge \cdots \wedge S_n \rightarrow D$, where $S_1, \ldots, S_n$ denote symptoms and D a disease. A diagnosis consists in searching for a rule for which the left part matches symptoms of a patient. Consider the following rules and symptoms of a patient: $S_1 \rightarrow D$, $(\neg S_1 \wedge S_2) \rightarrow D$, S_1, S_2. We fix Γ as the set of the previous formulæ. The pair $\mathcal{A} = \langle \{S_1 \rightarrow D, (\neg S_1 \wedge S_2) \rightarrow D, S_2\}, D\rangle$ is a classical argument over Γ. Clearly this argument cannot be considered as such. This is because it is based on the law of excluded middle. Indeed, by using the validity of $S_1 \vee \neg S_1$, if S_1 then D is true because of the rule $S_1 \rightarrow D$; otherwise, $\neg S_1$ is true and, by using the rule $(\neg S_1 \wedge S_2) \rightarrow D$, D is also true, since we have S_2 in the support. Therefore this argument suggests that D is true because S_1 is true or $\neg S_1 \wedge S_2$ is true. However only S_1 is true and $\neg S_1 \wedge S_2$ cannot by no mean considered as a possible justification of D. The argument $\mathcal{A}$ is not constructive because it is based on the law of excluded middle. There exists a single constructive argument over Γ having D as conclusion which is the pair $\langle \{S_1, S_1 \rightarrow D\}, D\rangle$. This argument provides the justification explaining why the patient has the disease D, i.e., the patient has the symptom S_1. Indeed we can say that constructive arguments get rid of imprecision (S_1 or $\neg S_1 \wedge S_2$) and provide arguments with a precise justification, namely S_1 in the previous example.

Properties of Intuitionistic Logic: Application to Constructive Arguments

We have shown in the previous section that instantiating Dung's framework with intuitionistic logic defines a new argumentation framework in which the notion of argument and its characterization with a minimal inconsistent set completely differs from classical argumentation framework. Not only does the new framework compute an argument in favor of a statement but it also builds the precise justification for that statement. Given the virtues of IPL, in this section we go into a more detailed exposition of this logic. We illustrate its applicability on argumentation reasoning.

Non-Interdefinability of Connectives

In classical logic, it is possible to define the logical connective $\wedge$ (resp. $\vee$) by using $\vee$ and $\neg$ (resp. $\wedge$ and $\neg$) which is not possible in intuitionistic logic. Indeed in intuitionistic logic, it is not possible to reformulate the logical connectives $\wedge$, $\vee$ and $\rightarrow$. The most of de Morgan laws are not valid in this logic. For instance, the formula $A \rightarrow B$ is not equivalent to $\neg A \vee B$. Moreover, double-negation elimination is excluded from IPL, i.e., A is not equivalent to $\neg\neg A$.

Intuitively reasoning in intuitionistic logic is not carried out in terms of "true" and "false", but in terms of "proof" and "contradiction". For instance, the formula $\neg(A \wedge B)$ means that we can prove from both A and B that we get a contradiction. However it does not mean that we can prove that (i) we get a contradiction from A or (ii) we get a contradiction from B as it may be suggested by $(\neg A \vee \neg B)$.

For instance, consider the formula $\neg(p \wedge q) \rightarrow (\neg p \vee \neg q)$ which is valid in classical logic. This formula is not valid in intuitionistic logic because it admits the countermodel $\mathcal{M} = (\{w, w', w''\}, \preceq, V)$ where $\preceq$ is defined by $w \preceq w'$ and $w \preceq w''$, and $V(w) = \emptyset$, $V(w') = \{p\}$ and $V(w') = \{q\}$. Indeed, we have $w \Vdash_{\mathcal{M}} \neg(p \wedge q)$, because of $w' \nVdash_{\mathcal{M}} p \wedge q$ and $w'' \nVdash_{\mathcal{M}} p \wedge q$; and we have $w \nVdash_{\mathcal{M}} \neg p$ because of $w' \Vdash_{\mathcal{M}} p$, and $w \nVdash_{\mathcal{M}} \neg q$ because of $w'' \Vdash_{\mathcal{M}} q$.

Let us consider the following statements:

1. Peter cannot be an owner and a tenant of an apartment: $\neg(O \wedge T)$.
2. Peter is an owner of an apartment: O.

We put $\Gamma = \{\neg(O \wedge T), O\}$. The pair $\mathcal{A} = \langle \{\neg(O \wedge T)\}, \neg O \vee \neg T\rangle$ is a classical argument over Γ. This means that from $\neg(O \wedge T)$ we obtain that Peter is not an owner of an apartment or he is not a tenant of an apartment. Note that the conclusion of $\mathcal{A}$ is obtained by using one of the following instances of the law of excluded middle: $(O \vee \neg O)$ and $(T \vee \neg T)$. Indeed, if we have O (resp. T) then, by using $\neg(O \wedge T)$, we have $\neg T$ (resp. $\neg O$). Otherwise, we have $\neg O$ (resp. $\neg T$). Thus, the argument $\mathcal{A}$ allows us to know $\neg O \vee \neg T$ without precisely knowing whether Peter is not an owner of an apartment ($\neg O$) or he is not a tenant of an apartment ($\neg T$). The unique constructive argument over Γ having $\neg O \vee \neg T$ as conclusion is $\langle \{O, \neg(O \wedge T)\}, \neg O \vee \neg T\rangle$. This argument is constructive because we know that Peter is not a tenant ($\neg T$) and, *a fortiori*, we have $\neg O \vee \neg T$.

As a second example, consider the formula $(p \rightarrow q) \rightarrow (\neg p \vee q)$ which is not valid in intuitionistic logic. A countermodel, among others, of this formula is $\mathcal{M} = (\{w, w'\}, \preceq, V)$ where $V(w) = \emptyset$, $V(w') = \{p, q\}$ and $\preceq$ is defined by $w \preceq w'$. This is obtained from $w \Vdash_{\mathcal{M}} p \rightarrow q$, $w \nVdash_{\mathcal{M}} \neg p$ because of $w' \Vdash_{\mathcal{M}} p$, and $w \nVdash_{\mathcal{M}} p$.

In order to illustrate the fact that the formula $(p \rightarrow q) \rightarrow (\neg p \vee q)$ is not valid in intuitionistic logic, consider the following statements:

- If Peter is a tenant, then he will buy an apartment: $T \rightarrow B$.
- Peter is a tenant: T.

We put $\Gamma = \{T \rightarrow B, T\}$. In the classical argument $\langle \{T \rightarrow B\}, \neg T \vee B\rangle$ we know that Peter is not a tenant or he will buy an apartment without exactly knowing which of these statements is true. In this context, constructivism requires to know this information in order to obtain the conclusion $\neg T \vee B$. For instance, $\langle \{T, T \rightarrow B\}, \neg T \vee B\rangle$ is a constructive

argument because its support allows us to know that Peter will buy an apartment.

As a third example, consider the following statement: it is note true that if the suspect is guilty then he confesses his crime ($\neg(G \rightarrow C)$). From this statement, one can deduce that the suspect is guilty in classical reasoning. Indeed, the pair $\langle\{\neg(G \rightarrow C)\}, G\rangle$ is a classical argument, since we have $\neg(G \rightarrow C) \equiv \neg(\neg G \vee C) \equiv G \wedge \neg C$. This comes from the interdefinability of connectives in classical logic and the double-negation elimination. However, the previous pair is not a constructive argument, since we are not able to construct G from $\neg(G \rightarrow C)$.

Disjunction Property

The disjunction property is one of the most important properties satisfied in intuitionistic logic. It says that if a formula $A \vee B$ is valid, then A is valid or B is valid. This property is not satisfied in classical logic. Indeed, the formula $p \vee \neg p$ is valid without p and $\neg p$ being individually valid in classical logic. From the point of view of constructivism, the disjunction property says that to construct the object $A \vee B$, it is necessary to be able to construct at least one of the objects A and B.

$$\frac{}{\Gamma, A \vdash A}\ [Id] \qquad \frac{}{\Gamma, \bot \vdash C}\ [\bot]$$

$$\frac{\Gamma, A, B \vdash C}{\Gamma, A \wedge B \vdash C}\ [\wedge_L] \qquad \frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ [\wedge_R]$$

$$\frac{\Gamma \vdash A_i}{\Gamma \vdash A_1 \vee A_2}\ [\vee_R^{i \in \{1,2\}}] \qquad \frac{\Gamma, A \vdash C \qquad \Gamma, B \vdash A}{\Gamma, A \vee B \vdash C}\ [\vee_L]$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}\ [\rightarrow_R]$$

$$\frac{\Gamma, A \rightarrow B \vdash A \qquad \Delta, B \vdash C}{\Gamma, \Delta, A \rightarrow B \vdash C}\ [\rightarrow_L]$$

Figure 1: Sequent Calculus G_{IPL}

Here we use the fact that intuitionistic logic enjoys the disjunction property to show that if a constructive argument has a support Δ which does not contain the disjunction connective, and a conclusion of the form $A \vee B$, then from Δ we can construct one of the formulas A and B, i.e., $\langle\Delta, A\rangle$ or $\langle\Delta, B\rangle$ is a constructive argument. In order to show this property, we use a proof system for IPL in the sequent calculus formalism-style.

Let us recall that an *inference rule* has the following form:

$$\frac{P_1 \quad \cdots \quad P_n}{C}\ [R]$$

where $[R]$ is its name, C its *conclusion* and $P_1, \ldots, P_n$ its *premises*. An *axiom* can be seen as a rule without premises. A *proof system* is defined as a set of inference rules. Proof-search in a sequent calculus corresponds to a bottom-up construction of derivations using its inference rules, i.e., a construction from the conclusion to axioms.

We consider here the sequent calculus G_{IPL} for IPL described in Figure 1 (see (Troelstra and Schwichtenberg 1996)). A sequent $\mathcal{S}$ has a proof in G_{IPL} if it has a finite derivation in G_{IPL} where each leaf node is labeled with an axiom. For instance, we provide a proof of $\{p \rightarrow r, q \rightarrow r\} \vdash p \vee q \rightarrow r$ using G_{IPL} in Figure 2.

We now show a property satisfied by constructive arguments that comes from the disjunction property.

Proposition 6. *If $\langle\Delta, A \vee B\rangle$ is a constructive argument and $\vee$ does not appear in Δ, then $\langle\Delta, A\rangle$ or $\langle\Delta, B\rangle$ is a constructive argument.*

Proof. By induction on the proof of $\Delta \vdash A \vee B$ in the sequent calculus G_{IPL}. Note that $\Delta \vdash A \vee B$ can not be an instance of an axiom ($[\bot]$ and $[Id]$), since $\Delta \nvdash \bot$ and $A \vee B$ is not a subformula of Δ. If the last application rule is a right rule, then it is an instance of $[\vee_R]$. Hence, we have a proof of $\Delta \vdash A$ or $\Delta \vdash B$ in G_{IPL}. Consequently, $\langle\Delta, A\rangle$ is a constructive argument or $\langle\Delta, B\rangle$ is a constructive argument, since $\Delta \nvdash \bot$ and Δ is minimal. We now consider the case where the last application rule is an instance of a left rule. In this case, the last application rule is an instance of either $[\wedge_L]$ or $[\rightarrow_L]$.

In the case of $[\wedge_L]$:

$$\frac{\Delta', C, D \vdash A \vee B}{\Delta', C \wedge D \vdash A \vee B}\ [\wedge_L]$$

where $\Delta = \Delta', A \wedge B$, by applying the induction hypothesis on $\Delta', C, D \vdash A \vee B$, we obtain a proof of $\Delta', C, D \vdash A$ or $\Delta', C, D \vdash B$ in G_{IPL}. Hence, we have a proof of $\Delta', C \wedge D \vdash A$ or $\Delta', C \wedge D \vdash B$ in G_{IPL}. Therefore, (Δ, A) is a constructive argument or (Δ, B) is a constructive argument.

In the case of $[\rightarrow_L]$:

$$\frac{\Delta', C \rightarrow D \vdash C \qquad \Delta', D \vdash A \vee B}{\Delta', C \rightarrow D \vdash A \vee B}\ [\rightarrow_L]$$

where $\Delta = \Delta', A \rightarrow B$, by applying the induction hypothesis on $\Delta', D \vdash A \vee B$, we obtain a proof of $\Delta', D \vdash A$ or $\Delta', D \vdash B$ in G_{IPL}. Hence, using the rule $[\rightarrow_L]$, we have a proof of $\Delta', C \rightarrow D \vdash A$ or $\Delta', C \rightarrow D \vdash B$ in G_{IPL}. Therefore, $\langle\Delta, A\rangle$ or $\langle\Delta, B\rangle$ is a constructive argument. □

Note that Proposition 6 is not satisfied in classical logic. For instance, the pair $\langle\{\neg(p \wedge q)\}, \neg p \vee \neg q\rangle$ is a classical argument where the support does not contain the disjunction connective. However, neither $\langle\{\neg(p \wedge q)\}, \neg p\rangle$ nor $\langle\{\neg(p \wedge q)\}, \neg q\rangle$ are classical arguments.

Computing Constructive Arguments using Modal Logic S4

In this section we borrow from classical argumentation framework the characterization of classical arguments in terms of minimal inconsistent sets. As shown in (Besnard et al. 2010) this characterization offers nice tractability properties for computing classical arguments. We provide such a characterization for constructive arguments using modal logic S4 (Blackburn, de Rijke, and Venema 2001).

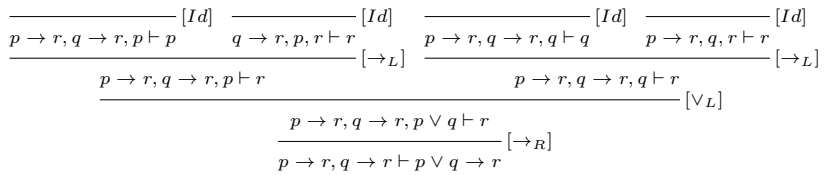


Figure 2: A proof in G_{IPL}

Modal Logic S4

The set of S4 formulæ is obtained by extending the propositional language with the modal connectives $\Box$ et $\Diamond$:

$$A ::= p \mid \bot \mid A \wedge A \mid A \vee A \mid A \rightarrow A \mid \Box A \mid \Diamond A.$$

Similarly to intuitionistic logic, modal logic S4 has a possible world semantics where we use a universe of worlds with an accessibility relation between worlds which is reflexive and transitive. More precisely, an S4 model is a triple $\mathcal{M} = (W, \preceq, V)$, where $\preceq$ is a preorder over W and $V : W \rightarrow 2^{\mathsf{Prop}}$ is an interpretation. Hence, a S4 model can be seen as a Kripke model of intuitionistic logic without Kripke monotonicity property.
The forcing relation, denoted $\vDash^{S4}_{\mathcal{M}}$, is inductively defined on formula structure as follows:

$w \vDash^{S4}_{\mathcal{M}} p$ iff $p \in V(w)$; $\quad w \vDash^{S4}_{\mathcal{M}} \bot$ never holds;
$w \vDash^{S4}_{\mathcal{M}} A \wedge B$ iff $w \vDash^{S4}_{\mathcal{M}} A$ and $w \vDash^{S4}_{\mathcal{M}} B$;
$w \vDash^{S4}_{\mathcal{M}} A \vee B$ iff $w \vDash^{S4}_{\mathcal{M}} A$ or $w \vDash^{S4}_{\mathcal{M}} B$;
$w \vDash^{S4}_{\mathcal{M}} A \rightarrow B$ iff if $w \vDash^{S4}_{\mathcal{M}} A$ then $w \vDash^{S4}_{\mathcal{M}} B$;
$w \vDash^{S4}_{\mathcal{M}} \Diamond A$ iff $\exists w' \in W$ s.t. $w \preceq w'$ and $w' \vDash^{S4}_{\mathcal{M}} A$;
$w \vDash^{S4}_{\mathcal{M}} \Box A$ iff $\forall w' \in W$ with $w \preceq w'$, $w' \vDash^{S4}_{\mathcal{M}} A$;

Embedding Intuitionistic Logic into S4

We describe here Gödel's embedding of intuitionistic logic into modal logic S4 (see, e.g., (Troelstra and Schwichtenberg 1996)). Intuitively, this embedding comes from the fact that the Kripke models of intuitionistic logic are S4 models. The definition of the embedding $(\cdot)^{S4}$ is by induction on formula structure as follows:

$(p)^{S4} = \Box p$; $(\bot)^{S4} = \bot$; $(A \wedge B)^{S4} = (A)^{S4} \wedge (B)^{S4}$;
$(A \vee B)^{S4} = (A)^{S4} \vee (B)^{S4}$;
$(A \rightarrow B)^{S4} = \Box((A)^{S4} \rightarrow (B)^{S4})$.

We have the following property:

Proposition 7. *$\Gamma \vdash_i C$ is valid in intuitionistic logic iff $(\Gamma)^{S4} \vdash (C)^{S4}$ is valid in S4.*

Hence, since the logical consequence concept in S4 is classical, we obtain the following proposition:

Proposition 8. *The pair $\langle \Delta, C \rangle$ is a constructive argument iff $(\Delta)^{S4} \cup \{\neg(C)^{S4}\}$ is a minimal inconsistent subset in S4.*

Prop. 8 states that the use of $S4$ allows to provide a simple characterization of being a constructive argument similar to that of being a classical argument. Such a characterization can be used in a constructive argument generation in the same way as in (Besnard et al. 2010). However, notice that the computation of minimal inconsistent subsets in modal logics is much less studied than in classical propositional logic. An interesting idea would be exploring how the MUS computation methods in classical propositional logic could benefit to modal logics. This is left for future work.

Conclusion and Future works

Constructivism is an approach which requires that to prove the existence of an object, it is necessary to be able to construct it. So far the main application of intuitionistic logic in computer science is using the Curry-Howard correspondence (Howard 1980) which corresponds to a direct relationship between constructive proofs and computer programs (Nordström, Petersson, and Smith 1990; Paulin-Mohring and Werner 1993).

In this paper we show the benefits of using intuitionistic logic to reason about inconsistency in argumentation theory. In particular Dung's framework is instantiated with this logic. In this setting, not only does the support of an argument deduce the conclusion of that argument but also constructs that conclusion. The present paper comes to complete existing works studying the validity of the logic-based instantiations of Dung's framework (Amgoud and Besnard 2013). While the focus of these works has been on the quality of the output of the logic-based argumentation frameworks (in terms of postulates), no attention has been paid on the argument itself, in particular the support of the argument. In our setting, a set of formulas which deduces a conclusion would not be a support of an argument for that conclusion if the reasons for such a deduction are not exactly identified. For example $\langle \{\neg(O \wedge T)\}, \neg O \vee \neg T \rangle$ is not an argument in our setting while $\langle \{O, \neg(O \wedge T)\}, \neg O \vee \neg T \rangle$ is. In addition, we provided a characterization of being a constructive argument in terms of a minimal inconsistent subset using Gödel's embedding of intuitionistic logic into the modal logic S4. Our work should be useful in diagnosis-based applications and law reasoning, to cite few.

As a future work, we intend to investigate the use of the Curry-Howard correspondence in encoding the proofs of the constructive arguments. Indeed, we know that each proof of a constructive argument can be encoded as a λ-term in a typed λ-calculus (Howard 1980). In this context, we plan to consider a constructive argumentation framework where we associate to each constructive argument a λ-term encoding a method used in the argument to construct its conclusion from its support. In this case, one of the perspectives consists in defining a new type of attack relations over the λ-terms.

References

Amgoud, L., and Besnard, P. 2013. Logical limits of abstract argumentation frameworks. *Journal of Applied Non-Classical Logics* 23(3):229–267.

Beeson, M. J. 1984. *Foundations of Constructive Mathematics*. Modern Surveys in Mathematics. Springer-Verlag.

Bench-Capon, T. 2003. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation* 13(3):429–448.

Besnard, P., and Hunter, A. 2008. *Elements of Argumentation*. MIT Press.

Besnard, P.; Grégoire, E.; Piette, C.; and Raddaoui, B. 2010. Mus-based generation of arguments and counter-arguments. In *IEEE International Conference on Information Reuse and Integration (IRI'10)*, 239–244.

Blackburn, P.; de Rijke, M.; and Venema, Y. 2001. *Modal Logic*. Cambridge University Press.

Caminada, M., and Amgoud, L. 2007. On the evaluation of argumentation formalisms. *Artificial Intelligence* 171(5-6):286–310.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77:321–357.

Grégoire, E.; Mazure, B.; and Piette, C. 2009. Using local search to find msses and muses. *European Journal of Operational Research* 199(3):640–646.

Howard, W. A. 1980. The formulæ-as-types notion of Construction. Academic Press. 479–490.

Nordström, B.; Petersson, K.; and Smith, J. M. 1990. *Programming in Martin-Löf's Type Theory, an introduction*, volume 7 of *Monographs on Computer Science*. Oxford Press.

Paulin-Mohring, C., and Werner, B. 1993. Synthesis of ml programs in system coq. *Journal of Symbolic Computation* 15:607–640.

Simari, G., and Loui, R. 1992. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence* 53:125–157.

Statman, R. 1979. Intuitionistic propositional logic is polynomial-space complete. *Theor. Comput. Sci.* 9:67–72.

Troelstra, A. S., and Schwichtenberg, H. 1996. *Basic Proof Theory*, volume 43 of *Cambridge tracks in Theoretical Computer Science*. Cambridge University Press.

Troelstra, A. S., and van Dalen, D. 1988. *Constructivism in Mathematics, an Introduction*. Studies in Logic and the foundations of Mathematics. North-Holland.

Bibliographie

[AdS13] L. Amgoud and F. Dupin de Saint-Cyr. An axiomatic approach for persuasion dialogs. In *25th IEEE International Conference on Tools with Artificial Intelligence, ICTAI*, pages 618–625, Herndon, VA, USA, 2013. IEEE Computer Society.

[AG17] C. Ansótegui and J. Gabàs. WPM3 : an (in)complete algorithm for weighted partial maxsat. *Artificial Intelligence*, 250 :37–57, 2017.

[Agg15] C. C. Aggarwal. *Data Mining - The Textbook*. Springer, 2015.

[AIS93] R. Agrawal, T. Imieliński, and A. Swami. Mining Association Rules Between Sets of Items in Large Databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, pages 207–216, New York, NY, USA, 1993. ACM.

[AMP00] L. Amgoud, N. Maudet, and S. Parsons. Modeling dialogues using argumentation. In *4th International Conference on Multi-Agent Systems, ICMAS*, pages 31–38, Boston, MA, USA, 2000. IEEE Computer Society.

[APM00] L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. In *ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence*, pages 338–342, Berlin, Germany, 2000. IOS Press.

[AR13] C. C. Aggarwal and C. K. Reddy. *Data clustering : algorithms and applications*. CRC Press, 2013.

[ARSO15] M. Ammoura, B. Raddaoui, Y. Salhi, and B. Oukacha. On Measuring Inconsistency Using Maximal Consistent Sets. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 13th European Conference, ECSQARU 2015, Compiègne, France*, pages 267–276, 2015.

[ASOR17] M. Ammoura, Y. Salhi, B. Oukacha, and B. Raddaoui. On an MCS-based inconsistency measure. *International Journal of Approximate Reasoning*, 80 :443–459, 2017.

[AU07] H. Arimura and T. Uno. An efficient polynomial space and polynomial delay algorithm for enumeration of maximal motifs in a sequence. *Journal of Combinatorial Optimization*, 13(3) :243–262, 2007.

[Bay98] R. J. Bayardo Jr. Efficiently Mining Long Patterns from Databases. In *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Mana-*

gement of Data, June 2-4, 1998, Seattle, Washington, USA., pages 85–93, 1998.

[BB03] O. Bailleux and Y. Boufkhad. Efficient CNF Encoding of Boolean Cardinality Constraints. In *9th International Conference on Principles and Practice of Constraint Programming (CP 2003), Kinsale, Ireland*, pages 108–122, 2003.

[BBR09] O. Bailleux, Y. Boufkhad, and O. Roussel. New Encodings of Pseudo-Boolean Constraints into CNF. In *Theory and Applications of Satisfiability Testing (SAT'09), Swansea, UK*, pages 181–194, 2009.

[BD12] L. Billard and E. Diday. *Symbolic Data Analysis : Conceptual Statistics and Data Mining*. John Wiley & Sons, 2012.

[BDP95] S. Benferhat, D. Dubois, and H. Prade. How to Infer from Inconsisent Beliefs without Revising ? In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, 1995, 2 Volumes*, pages 1449–1457, 1995.

[BDP96] S. Benferhat, D. Dubois, and H. Prade. Reasoning in Inconsistent Stratified Knowledge Bases. In *26th IEEE International Symposium on Multiple-Valued Logic, ISMVL 1996, Santiago de Compostela, Spain, 1996, Proceedings*, pages 184–191, 1996.

[Ber06] P. Berkhin. A Survey of Clustering Data Mining Techniques. In Jacob Kogan, Charles K. Nicholas, and Marc Teboulle, editors, *Grouping Multidimensional Data - Recent Advances in Clustering*, pages 25–71. Springer, 2006.

[Bes06] C. Bessiere. Constraint Propagation. In *Handbook of Constraint Programming*, pages 29–83. 2006.

[Bes14] P. Besnard. Revisiting Postulates for Inconsistency Measures. In *Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal*, pages 383–396. Springer, 2014.

[BH08] P. Besnard and A. Hunter. *Elements of Argumentation*. MIT Press, 2008.

[BHvW09] A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.

[BJSS14] B. Benhamou, S. Jabbour, L. Sais, and Y. Salhi. A Generic and Declarative Method for Symmetry Breaking in Itemset Mining. In *Knowledge Discovery, Knowledge Engineering and Knowledge Management - 6th International Joint Conference, IC3K 2014, Rome, Italy, Revised Selected Papers*, pages 143–160, 2014.

[BJSS17a] A. Boudane, S. Jabbour, L. Sais, and Y. Salhi. Clustering Complex Data Represented as Propositional Formulas. In *Advances in Knowledge Discovery and Data Mining - 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea*, pages 441–452, 2017.

[BJSS17b] A. Boudane, S. Jabbour, L. Sais, and Y. Salhi. Enumerating Non-redundant Association Rules Using Satisfiability. In *Advances in Knowledge Discovery and Data Mining - 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea*, pages 824–836, 2017.

[BJSS17c] A. Boudane, S. Jabbour, L. Sais, and Y. Salhi. Une approche logique pour la fouille de règles d'association. In *17ème Journées Francophones Extraction et Gestion des Connaissances, EGC 2017, Grenoble, France*, pages 357–362, 2017.

[BJSS18] A. Boudane, S. Jabbour, L. Sais, and Y. Salhi. SAT-Based Data Mining. *International Journal on Artificial Intelligence Tools*, 27(1) :1–24, 2018.

[BJSY16] A. Boudane, S. Jabbour, L. Sais, and Y.Salhi. A SAT-Based Approach for Mining Association Rules. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA*, pages 2472–2478, 2016.

[BK18] A. Biere and D. Kröning. SAT-Based Model Checking. In *Handbook of Model Checking*, pages 277–303. Springer, 2018.

[BLM18] C. Bessiere, N. Lazaar, and M. Maamar. User's Constraints in Itemset Mining. In *Principles and Practice of Constraint Programming - 24th International Conference, CP 2018, Lille, France*, pages 537–553, 2018.

[BM11] E. Bonzon and N. Maudet. On the outcomes of multiparty persuasion. In *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Volume 1-3*, pages 47–54, Taipei, Taiwan, 2011. IFAAMAS.

[Boc00] H. H. Bock. *Analysis of Symbolic Data : Exploratory Methods for Extracting Statistical Information from Complex Data*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2000.

[BPT+00] Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, and L. Lakhal. Mining Minimal Non-redundant Association Rules Using Frequent Closed Itemsets. In *Computational Logic - CL 2000, First International Conference, London, UK, 24-28 July, 2000, Proceedings*, pages 972–986, 2000.

[BS18] Y. Boumarafi and Y. Salhi. Tractable Classes in Exactly-One-SAT. In *Artificial Intelligence : Methodology, Systems, and Applications - 18th International Conference, AIMSA 2018, Varna, Bulgaria*, pages 197–206, 2018.

[BSS17] Y. Boumarafi, L. Sais, and Y. Salhi. From SAT to Maximum Independent Set : A New Approach to Characterize Tractable Classes. In *LPAR-21, 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Maun, Botswana*, pages 286–299, 2017.

[Cal08] T. Calders. Itemset frequency satisfiability : Complexity and axiomatization. *theoretical Computer Science*, 394(1-2) :84–111, 2008.

[CHH+18] L. A. Chalaguine, E. Hadoux, F. Hamilton, A. Hayward, A. Hunter, S. Polberg, and H. W. W. Potts. Domain modelling in computational persuasion for behaviour change in healthcare. *CoRR*, 2018.

[CJSS12] E. Coquery, S. Jabbour, L. Saïs, and Y. Salhi. A SAT-Based Approach for Discovering Frequent, Closed and Maximal Patterns in a Sequence. In *ECAI 2012 - 20th European Conference on Artificial Intelligence, Montpellier, France*, pages 258–263, 2012.

[CKS16] J.-F. Condotta, S. Kaci, and Y. Salhi. Optimization in temporal qualitative constraint networks. *Acta Informatica*, 53(2) :149–170, 2016.

[CRS16] J.-F. Condotta, B. Raddaoui, and Y. Salhi. Quantifying Conflicts for Spatial and Temporal Information. In *Principles of Knowledge Representation and Reasoning : Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa*, pages 443–452, 2016.

[CZZ04] Q. Chen, C. Zhang, and S. Zhang. A Verification Model for Electronic Transaction Protocols. In *Advanced Web Technologies and Applications, 6th Asia-Pacific Web Conference, APWeb 2004, Hangzhou, China*, volume 3007 of *Lecture Notes in Computer Science*, pages 824–833. Springer, 2004.

[DB11] J. Davies and F. Bacchus. Solving MAXSAT by solving a sequence of simpler SAT instances. In *Principles and Practice of Constraint Programming - CP 2011 - 17th International Conference, CP 2011, Perugia, Italy*, pages 225–239, 2011.

[dCCL09] F. de A.T. de Carvalho, Marc Csernel, and Y. Lechevallier. Clustering constrained symbolic data. *Pattern Recognition Letters*, 30(11) :1037–1045, 2009.

[DE03] E. Diday and F. Esposito. An introduction to symbolic data analysis and the SODAS software. *Intelligent Data Analysis*, 7(6) :583–601, 2003.

[DGHK18] G. De Bona, J. Grant, A. Hunter, and S. Konieczny. Towards a Unified Framework for Syntactic Inconsistency Measures. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA*, 2018.

[DGN08] L. De Raedt, T. Guns, and S. Nijssen. Constraint Programming for Itemset Mining. In *ACM SIGKDD*, pages 204–212, 2008.

[DMW19] E. Demirovic, N. Musliu, and F. Winter. Modeling and solving staff scheduling with partial weighted maxSAT. *Annals OR*, 275(1) :79–99, 2019.

[dSdC04] R. M.C.R. de Souza and F. de A.T. de Carvalho. Clustering of interval data based on city–block distances. *Pattern Recognition Letters*, 25(3) :353–365, 2004.

[Dun95] P. M. Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence*, 77(2) :321–358, 1995.

[ES06] N. Eén and N. Sörensson. Translating Pseudo-Boolean Constraints into SAT. *JSAT*, 2(1-4) :1–26, 2006.

[FGG+14] P. Fournier-Viger, A. Gomariz, T. Gueniche, A. Soltani, C.-W. Wu, and V. S. Tseng. SPMF : a Java open-source pattern mining library. *Journal of Machine Learning Research*, 15(1) :3389–3393, 2014.

[FM06a] Z. Fu and S. Malik. On solving the partial MAX-SAT problem. In *Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA*, pages 252–265, 2006.

[FM06b] Z. Fu and S. Malik. Solving the minimum-cost satisfiability problem using SAT based branch-and-bound search. In *2006 International Conference on Computer-Aided Design, ICCAD 2006, San Jose, CA, USA*, pages 852–859, 2006.

[Fog98] B. J. Fogg. Persuasive computers : Perspectives and research directions. In *Proceeding of the CHI '98 Conference on Human Factors in Computing Systems*, pages 225–232, Los Angeles, California, USA, 1998. ACM.

[Gär92] P. Gärdenfors. *Belief Revision*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1992.

[GD94] K. C. Gowda and E. Diday. *New Approaches in Classification and Data Analysis*, chapter Symbolic Clustering Algorithms using Similarity and Dissimilarity Measures. Springer Berlin Heidelberg, 1994.

[GDN+17] T. Guns, A. Dries, S. Nijssen, G. Tack, and L. De Raedt. Miningzinc : A declarative framework for constraint-based mining. *Artificial Intelligence*, 244 :6–29, 2017.

[GGQ+16] M. Gebser, T. Guyet, R. Quiniou, J. Romero, and T. Schaub. Knowledge-Based Sequence Mining with ASP. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA*, pages 1497–1504, 2016.

[GH13] J. Grant and A. Hunter. Distance-Based Measures of Inconsistency. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 12th European Conference, ECSQARU 2013, Utrecht, The Netherlands*, pages 230–241. Springer, 2013.

[GND11] T. Guns, S. Nijssen, and L. De Raedt. Itemset mining : A constraint programming perspective. *Artificial Intelligence*, 175(12-13) :1951–1983, 2011.

[Gun15] T. Guns. Declarative pattern mining using constraint programming. *Constraints*, 20(4) :492–493, 2015.

[HH18] E. Hadoux and A. Hunter. Learning and updating user models for subpopulations in persuasive argumentation using beta distributions. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS*, pages 1141–1149, Stockholm, Sweden, 2018. IFAAMAS/ACM.

[HJSS17] S. Hattad, S. Jabbour, L. Sais, and Y. Salhi. Enhancing Pigeon-Hole based Encoding of Boolean Cardinality Constraints. In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence, ICAART 2017, Volume 2, Porto, Portugal*, pages 299–307, 2017.

[HK10] A. Hunter and S. Konieczny. On the measure of conflicts : Shapley Inconsistency Values. *Artifificial Intelligence*, 174(14) :1007–1026, 2010.

[HP17a] A. Hunter and S. Polberg. Empirical methods for modelling persuadees in dialogical argumentation. In *29th IEEE International Conference on Tools with Artificial Intelligence, ICTAI*, pages 382–389, Boston, MA, USA, 2017. IEEE Computer Society.

[HP17b] A. Hunter and N. Potyka. Updating probabilistic epistemic states in persuasion dialogues. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 14th European Conference, ECSQARU, Proceedings*, pages 46–56, Lugano, Switzerland, 2017. Springer.

[HPW14] A. Hunter, S. Parsons, and M. Wooldridge. Measuring Inconsistency in Multi-Agent Systems. *Kunstliche Intelligenz*, 28 :169–178, 2014.

[HT16] A. Hunter and M. Thimm. Optimization of dialectical outcomes in dialogical argumentation. *International Journal of Approximate Reasoning*, 78 :73–102, 2016.

[Hun14] A. Hunter. Opportunities for argument-centric persuasion in behaviour change. In *Logics in Artificial Intelligence - 14th European Conference, JELIA. Proceedings*, pages 48–61, Funchal, Madeira, Portugal, 2014. Springer.

[Hun15] A. Hunter. Modelling the persuadee in asymmetric argumentation dialogues for persuasion. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*, pages 3055–3061, Buenos Aires, Argentina, 2015. AAAI Press.

[Hun16] A. Hunter. Computational persuasion with applications in behaviour change. In *Computational Models of Argument - Proceedings of COMMA*, pages 5–18, Potsdam, Germany, 2016. IOS Press.

[Hun18] A. Hunter. Towards a framework for computational persuasion with applications in behaviour change. *Argument & Computation*, 9(1) :15–40, 2018.

[Jac01] P. Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37 :547–579, 1901.

[JKS+13] S. Jabbour, M. Khiari, L. Sais, Y. Salhi, and K. Tabia. Symmetry-Based Pruning in Itemset Mining. In *25th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2013, Herndon, VA, USA*, pages 483–490, 2013.

[JKSS16] S. Jabbour, S. Kaci, L. Sais, and Y. Salhi. Itemset Mining with Penalties. In *28th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2016, San Jose, CA, USA*, pages 962–966, 2016.

[JLSS14] S. Jabbour, J. Lonlac, L. Sais, and Y. Salhi. Extending modern SAT solvers for models enumeration. In *Proceedings of the 15th IEEE International Conference on Information Reuse and Integration, IRI 2014, Redwood City, CA, USA*, pages 803–810, 2014.

[JMR+15] S. Jabbour, Y. Ma, B. Raddaoui, L. Sais, and Y. Salhi. On Structure-Based Inconsistency Measures and Their Computations via Closed Set Packing. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey*, pages 1749–1750, 2015.

[JMR+16] S. Jabbour, Y. Ma, B. Raddaoui, L. Sais, and Y. Salhi. A MIS Partition Based Framework for Measuring Inconsistency. In *Principles of Knowledge Representation and Reasoning : Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa*, pages 84–93. AAAI Press, 2016.

[JMSS14] S. Jabbour, J. Marques-Silva, L. Sais, and Y. Salhi. Enumerating Prime Implicants of Propositional Formulae in Conjunctive Normal Form. In *Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal*, pages 152–165, 2014.

[JRSS15] S. Jabbour, S. Roussel, L. Sais, and Y. Salhi. Mining to Compress Table Constraints. In *27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2015, Vietri sul Mare, Italy*, pages 405–412, 2015.

[JRSS16] S. Jabbour, B. Raddaoui, L. Sais, and Y. Salhi. On the Computation of Top-k Extensions in Abstract Argumentation Frameworks. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, The Hague, The Netherlands*, pages 913–920, 2016.

[JSS13a] S. Jabbour, L. Sais, and Y. Salhi. A Pigeon-Hole Based Encoding of Cardinality Constraints. *TPLP*, 13(4-5-Online-Supplement), 2013.

[JSS13b] S. Jabbour, L. Sais, and Y. Salhi. Boolean satisfiability for sequence mining. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA*, pages 649–658, 2013.

[JSS13c] S. Jabbour, L. Sais, and Y. Salhi. The top-k frequent closed itemset mining using top-k SAT problem. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic*, pages 403–418, 2013.

[JSS14] S. Jabbour, L. Sais, and Y. Salhi. A Pigeon-Hole Based Encoding of Cardinality Constraints. In *International Symposium on Artificial Intelligence and Mathematics, ISAIM 2014, Fort Lauderdale, FL, USA*, 2014.

[JSS15] S. Jabbour, L. Sais, and Y. Salhi. Decomposition Based SAT Encodings for Itemset Mining Problems. In *Advances in Knowledge Discovery and Data Mining - 19th Pacific-Asia Conference, PAKDD 2015, Ho Chi Minh City, Vietnam*, pages 662–674, 2015.

[JSS17] S. Jabbour, L. Sais, and Y. Salhi. Mining Top-k motifs with a SAT-based framework. *Artificial Intelligence*, 244 :30–47, 2017.

[JSST12] S. Jabbour, L. Sais, Y. Salhi, and K. Tabia. Symmetries in Itemset Mining. In *ECAI 2012 - 20th European Conference on Artificial Intelligence, Montpellier, France*, pages 432–437, 2012.

[JSSU13] S. Jabbour, L. Sais, Y. Salhi, and T. Uno. Mining-based compression approach of propositional formulae. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA*, pages 289–298, 2013.

[Kac11] S. Kaci. *Working with Preferences : Less Is More*. Cognitive Technologies. Springer, 2011.

[Kaz09] P. Kazienko. Mining Indirect Association Rules for Web Recommendation. *Applied Mathematics and Computer Science*, 19(1) :165–186, 2009.

[KBC10] M. Khiari, P. Boizumault, and B. Crémilleux. Constraint Programming for Mining n-ary Patterns. In *Principles and Practice of Constraint Programming - CP 2010 - 16th International Conference, CP 2010, St. Andrews, Scotland, UK*, pages 552–567, 2010.

[KLM03] S. Konieczny, J. Lang, and P. Marquis. Quantifying information and contradiction in propositional logic through test actions. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico*, pages 106–111. Morgan Kaufmann, 2003.

[Kry98] M. Kryszkiewicz. Representative Association Rules and Minimum Condition Maximum Consequence Association Rules. In *Principles of Data Mining and Knowledge Discovery, Second European Symposium, PKDD '98, Nantes, France, September 23-26, 1998, Proceedings*, pages 361–369, 1998.

[KS14] S. Kaci and Y. Salhi. A Constructive Argumentation Framework. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, Québec City, Québec, Canada*, pages 1070–1076, 2014.

[Lig13] G. Ligozat. *Qualitative Spatial and Temporal Reasoning*. ISTE. Wiley, 2013.

[LM06] I. Lynce and J. Marques-Silva. SAT in Bioinformatics : Making the Case with Haplotype Inference. In *Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA*, pages 136–141, 2006.

[Mic80] R. S. Michalski. Knowledge Acquisition Through Conceptual Clustering : A Theoretical Framework and an Algorithm for Partitioning Data into Conjunctive Concepts. *Journal of Policy Analysis and Information Systems*, 4(3) :219–244, 1980.

[MP08] J. Marques-Silva and J. Planes. Algorithms for maximum satisfiability using unsatisfiable cores. In *Design, Automation and Test in Europe, DATE 2008, Munich, Germany*, pages 408–413, 2008.

[MPS+07] M. V. Martinez, A. Pugliese, G. I. Simari, V. S. Subrahmanian, and H. Prade. How Dirty is your Relational Database ? An Axiomatic Approach. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 9th European Conference, ECSQARU 2007, Hammamet, Tunisia*, volume 4724 of *Lecture Notes in Computer Science*, pages 103–114. Springer, 2007.

[NB14] N. Narodytska and F. Bacchus. Maximum satisfiability using core-guided maxsat resolution. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, Québec, Canada.*, pages 2717–2723, 2014.

[OJS+15] I. Ouled Dlala, S. Jabbour, L. Sais, Y. Salhi, and B. Ben Yaghlane. Parallel SAT based closed frequent itemsets enumeration. In *12th IEEE/ACS International Conference of Computer Systems and Applications, AICCSA 2015, Marrakech, Morocco*, pages 1–8, 2015.

[PBTL99] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering Frequent Closed Itemsets for Association Rules. In *Database Theory - ICDT '99, 7th International Conference, Jerusalem, Israel*, pages 398–416, 1999.

[PCGS05] N. Pisanti, M. Crochemore, R. Grossi, and M.-F. Sagot. Bases of Motifs for Generating Repeated Patterns with Wild Cards. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(1) :40–50, 2005.

[Pra05] H. Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15(6) :1009–1040, 2005.

[PRF+00] L. Parida, I. Rigoutsos, A. Floratos, D. E. Platt, and Y. Gao. Pattern discovery on character sets and real-valued data : linear bound on irredundant motifs and an efficient polynomial time algorithm. In *Proceedings of the*

Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, USA, pages 297–308, 2000.

[RM70] N. Rescher and R. Manor. On inference from inconsistent premisses. *Theory and Decision*, 1(2) :179–217, 1970.

[Sal18] Y. Salhi. Approaches for Enumerating All the Essential Prime Implicants. In *Artificial Intelligence : Methodology, Systems, and Applications - 18th International Conference, AIMSA 2018, Varna, Bulgaria*, pages 228–239, 2018.

[Sal19a] Y. Salhi. Entailment Functions and Reasoning Under Inconsistency. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada*, pages 2183–2185, 2019.

[Sal19b] Y. Salhi. Entailment Functions and Reasoning Under Inconsistency. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada*, pages 2183–2185, 2019.

[Sal19c] Y. Salhi. On an Argument-centric Persuasion Framework. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada*, pages 1279–1287, 2019.

[SCS+15] M. Sioutis, J.-F. Condotta, Y. Salhi, B. Mazure, and D. A. Randell. Ordering Spatio-Temporal Sequences to Meet Transition Constraints : Complexity and Framework. In *Artificial Intelligence Applications and Innovations - 11th IFIP WG 12.5 International Conference, AIAI 2015, Bayonne, France*, pages 130–150, 2015.

[SCSM14] M. Sioutis, J.-F. Condotta, Y. Salhi, and B. Mazure. A Qualitative Spatio-Temporal Framework Based on Point Algebra. In *Artificial Intelligence : Methodology, Systems, and Applications - 16th International Conference, AIMSA 2014, Varna, Bulgaria*, pages 117–128, 2014.

[SCSM15] M. Sioutis, J.-F. Condotta, Y. Salhi, and B. Mazure. Generalized Qualitative Spatio-Temporal Reasoning : Complexity and Tableau Method. In *Automated Reasoning with Analytic Tableaux and Related Methods - 24th International Conference, TABLEAUX 2015, Wrocław, Poland*, pages 54–69, 2015.

[Sin05] C. Sinz. Towards an Optimal CNF Encoding of Boolean Cardinality Constraints. In *11th International Conference on Principles and Practice of Constraint Programming (CP'05), Sitges, Spain*, pages 827–831, 2005.

[SJS12] Y. Salhi, S. Jabbour, and L. Sais. Graded Modal Logic GS5 and Itemset Support Satisfiability. In *Information Search, Integration and Personalization - International Workshop, ISIP 2012, Sapporo, Japan. Revised Selected Papers*, pages 131–140, 2012.

[SL07] J. P. Marques Silva and I. Lynce. Towards Robust CNF Encodings of Cardinality Constraints. In *In Proceedings of 13th International Conference on Principles and Practice of Constraint Programming (CP'07), Providence, RI, USA*, pages 483–497, 2007.

[SMV+07] S. Safarpour, H. Mangassarian, A. G. Veneris, M. H. Liffiton, and K. A. Sakallah. Improved Design Debugging Using Maximum Satisfiability. In *Formal Methods in Computer-Aided Design, 7th International Conference, FMCAD 2007, Austin, Texas, USA*, pages 13–19, 2007.

[SS96] J. P. Marques Silva and K. A. Sakallah. GRASP - a new search algorithm for satisfiability. In *ICCAD*, pages 220–227, 1996.

[SS99] J. P. Marques Silva and K.A. Sakallah. GRASP : A Search Algorithm for Propositional Satisfiability. *IEEE Transactions on Computers*, 48(5) :506–521, 1999.

[SS15] Y. Salhi and M. Sioutis. A Resolution Method for Modal Logic S5. In *Global Conference on Artificial Intelligence, GCAI 2015, Tbilisi, Georgia*, pages 252–262, 2015.

[SSC15a] M. Sioutis, Y. Salhi, and J.-F. Condotta. On the use and effect of graph decomposition in qualitative spatial and temporal reasoning. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain*, pages 1874–1879, 2015.

[SSC15b] M. Sioutis, Y. Salhi, and J.-F. Condotta. A Simple Decomposition Scheme for Large Real World Qualitative Constraint Networks. In *Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2015, Hollywood, Florida, USA*, pages 119–122, 2015.

[SSC17] M. Sioutis, Y. Salhi, and J.-F. Condotta. Studying the use and effect of graph decomposition in qualitative spatial and temporal reasoning. *The Knowledge Engineering Review*, 32 :e4, 2017.

[Sza06] L. Szathmary. *Symbolic Data Mining Methods with the Coron Platform. (Méthodes symboliques de fouille de données avec la plate-forme Coron)*. PhD thesis, Henri Poincaré University, Nancy, France, 2006.

[TBMP13] K. Tanaka, F. Berto, E. D. Mares, and F. Paoli, editors. *Paraconsistency : Logic and Applications*, volume 26 of *Logic, Epistemology, and the Unity of Science*. Springer, 2013.

[Thi13] Matthias Thimm. Inconsistency measures for probabilistic logics. *Artificial Intelligence*, 197 :1–24, 2013.

[Thi16] M. Thimm. On the expressivity of inconsistency measures. *Artificial Intelligence*, 234 :120–151, 2016.

[Thi18] M. Thimm. On the Evaluation of Inconsistency Measures. In John Grant and Maria Vanina Martinez, editors, *Measuring Inconsistency in Information*, volume 73 of *Studies in Logic*. College Publications, February 2018.

[TKS00] P.-N. Tan, V. Kumar, and J. Srivastava. Indirect Association : Mining Higher Order Dependencies in Data. In *Principles of Data Mining and Knowledge Discovery, 4th European Conference, PKDD 2000, Lyon, France, September 13-16, 2000, Proceedings*, pages 632–637, 2000.

[TKS02] P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada*, pages 32–41, 2002.

[TS96] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, New York, NY, USA, 1996.

[Tse68] G.S. Tseitin. On the complexity of derivations in the propositional calculus. In H.A.O. Slesenko, editor, *Structures in Constructives Mathematics and Mathematical Logic, Part II*, pages 115–125, 1968.

[UBC+17] W. Ugarte, P. Boizumault, B. Crémilleux, A. Lepailleur, S. Loudni, M. Plantevit, C. Raïssi, and A. Soulet. Skypattern mining : From pattern condensed representations to dynamic constraint satisfaction problems. *Artificial Intelligence*, 244 :48–69, 2017.

[UBLC15] W. Ugarte, P. Boizumault, S. Loudni, and B. Crémilleux. Modeling and Mining Optimal Patterns Using Dynamic CSP. In *27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2015, Vietri sul Mare, Italy*, pages 33–40, 2015.

[WHLT05] J. Wang, J. Han, Y. Lu, and P. Tzvetkov. TFP : An Efficient Algorithm for Mining Top-K Frequent Closed Itemsets. *IEEE Transactions on Knowledge and Data Engineering*, 17(5) :652–664, 2005.

[Zak04] M. J. Zaki. Mining Non-Redundant Association Rules. *Data Mining and Knowledge Discovery*, 9(3) :223–248, 2004.