HAL
open science

Argumentation techniques for existential rules

Bruno Yun

To cite this version:

Bruno Yun. Argumentation techniques for existential rules. Other [cs.OH]. Université Montpellier, 2019. English. NNT : 2019MONTS011 . tel-02197405v2

HAL Id: tel-02197405

https://theses.hal.science/tel-02197405v2

Submitted on 27 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR
DE L'UNIVERSITÉ DE MONTPELLIER

En Informatique

École doctorale I2S

Unité de recherche LIRMM

ARGUMENTATION TECHNIQUES FOR EXISTENTIAL RULES

Présentée par Bruno YUN

Le 11 Juillet 2019

Sous la direction de Madalina CROITORU, Rallou THOMOPOULOS
et Srdjan VESIC

Devant le jury composé de

Madalina CROITORU, MCF HDR, Université de Montpellier, France	Directrice
Rallou THOMOPOULOS, DR, INRA, Montpellier, France	Co-directrice
Srdjan VESIC, CR, CNRS, Lens, France	Co-encadrant
Stefan WOLTRAN, PU, Vienna University of Technology, Vienne, Autriche	Rapporteur
Leon VAN DER TORRE, PU, Faculty of Science, Technology and Communication, Luxembourg	Rapporteur
Sanjay MODGIL, Senior Lecturer, King's College London, Londres, Royaume-Uni	Examinateur
Marie-Christine ROUSSET, PU, University of Grenoble Alpes, Grenoble, France	Présidente du Jury

UNIVERSITÉ
DE MONTPELLIER

Thanks

I want to express my gratitude to those that supervised me during my thesis, namely Madalina CROITORU, Pierre BISQUERT, Srdjan VESIC and Rallou THOMOPOULOS. They have always been the source of fruitful (scientific) discussions and I will never forget the unconditional and endless support that I received from each of them.

I also truly enjoyed being a part of the GRAPHIK team. They all succeeded in awakening my curiosity for computer science and knowledge representation in particular.

I thank all the co-workers and PhD students that I met at LIRMM and INRA (especially Abdelraouf, Abdallah, Clément, Fati, Jocelyn, Martin and Stathis) for the working atmosphere, advices and all the good moments that we shared.

Now, I would like to thank everyone unrelated to research for their help. First, I thank my parents and my brother for their trust and the help that they provided me. Second, I thank all of my friends for the entertainment and the laughing moments. Lastly, I apologise to everyone that was not explicitly mentioned.

Abstract

In this thesis, we investigate reasoning techniques with argumentation graphs generated from inconsistent knowledge bases expressed in the existential rules language. The existential rules language is a subset of first-order logic in which a knowledge base is composed of two layers: a fact layer and an ontology layer. The fact layer consists of factual knowledge (usually stored in relational databases) whereas the ontology layer consists of reasoning rules of deduction and negative constraints. Since the classical query answering approaches fail in the presence of inconsistencies, we chose to work with an conflict-tolerant reasoning approach that is based on building graphs with structured arguments and attacks from the deductions of the underlying logical knowledge base.

The three main results are the following. First, we study how argumentation graphs are obtained from knowledge bases expressed in existential rules, the structural properties of such graphs and show several insights as to how their generation can be improved. Second, we propose a framework that generates an argumentation graph with a special feature called sets of attacking arguments instead of the regular binary attack relation and show how it improves upon the current state-of-the-art using an empirical analysis. Third, we interest ourselves to ranking-based approaches in both the context of query answering and argumentation reasoning. In the former, we introduce a framework that is based on ranking maximal consistent subsets of facts (repairs) in order to have a more productive query answering. In the latter, we set up the foundation theory for semantics that rank arguments in argumentation graphs with sets of attacking arguments.

Keywords: Argumentation, Inconsistency, Graphs, Existential rules, *Datalog*±.

Résumé (version courte)

Dans cette thèse, nous étudions les techniques de raisonnement utilisant des graphes d'argumentation générés à partir de bases de connaissances inconsistantes exprimées dans le langage des *règles existentielles*. Les trois principaux résultats sont les suivants. Tout d'abord, nous étudions les propriétés structurelles des graphes obtenus à partir de bases de connaissances exprimées avec des règles existentielles et nous donnons plusieurs indications sur la manière dont leur génération peut être améliorée. Deuxièmement, nous proposons une technique pour générer un graphe d'argumentation où plusieurs arguments peuvent attaquer collectivement, remplaçant ainsi la relation d'attaque binaire classique et montrons expérimentalement les avantages de cette technique. Troisièmement, nous nous intéressons aux approches fondées sur les classements pour le raisonnement en logique et en argumentation.

Mots clés: Argumentation, Inconsistance, Graphes, Règles existentielles, *Datalog*±.

Résumé (version longue)

Cette thèse présente un travail de recherche original dans le domaine de la représentation des connaissances et du raisonnement, l'un des principaux sous-domaines de l'intelligence artificielle. Le langage de représentation des connaissances que nous abordons est appelé règles existentielles, une famille de langages logiques correspondant au fragment existentiel conjonctif positif de la logique du premier ordre. Le domaine de la représentation des connaissances s'intéresse à la découverte de formalismes fournissant une description du monde pouvant être utilisée efficacement pour créer des applications "intelligentes". Dans ce contexte, le terme "intelligent" désigne la capacité d'un système à trouver les conséquences implicites des connaissances explicitement représentées. Dans sa forme la plus simple, les données peuvent être stockées explicitement par des expressions sous différentes formes, par exemple dans une base de données relationnelle [Codd, 1970] ou un triplet RDF. Cependant, afin d'obtenir cette "connaissance implicite" des données stockées, les communautés de base de données et de représentation des connaissances ont reconnu la nécessité de structurer les données en informations et en connaissances. Ainsi, des bases de données déductives avec des ontologies ont été créées pour déduire des données implicites, palliant ainsi l'incomplétude des bases de données classiques. L'introduction d'ontologies a également permis l'enrichissement et l'unification de vocabulaires appartenant à plusieurs sources de données. Dans le Web sémantique, les connaissances ontologiques sont souvent représentées par des formalismes

basés sur les logiques de description [Baader et al., 2005]. Cependant, les logiques de description ont une expressivité limitée: outre le fait qu'elles ne prennent en charge que des prédicats unaires et binaires, la connaissance ontologique ne peut être décrite qu'en matière de structure arborescente (aucun cycle n'est autorisé). Dans un même temps, le langage de base de données déductives Datalog [Gallaire and Nicolas, 1987; Ceri et al., 1989], qui est un sous-ensemble syntaxique de Prolog [Colmerauer and Roussel, 1996], a été élu comme langage par défaut pour les bases de données de requêtes. Cependant, dans ce langage, il est uniquement possible de produire des connaissances sur des individus déjà connus et il n'est pas possible de déduire l'existence d'individus inconnus. Ceci est une caractéristique cruciale car nous ne pouvons pas supposer que tous les individus soient connus à l'avance.

Le langage *Datalog+* a été proposé pour répondre à ces deux exigences, c'est-à-dire la capacité d'invention et la capacité à exprimer des structures de haut niveau telles que des prédicats n-aires ou des ontologies plus complexes. Le langage *Datalog+* est une extension plus expressive du langage Datalog [Calì et al., 2013] avec la possibilité de déduire des informations sur des individus inconnus. Cependant, cette nouvelle capacité posa de nombreux problèmes de calculabilité pour le traitement des requêtes. Afin d'éviter ces problèmes, la famille de langages *Datalog±* [Calì et al., 2013, 2009] a été introduite. La famille *Datalog±* correspond à l'ensemble des langages qui restreignent la syntaxe des règles de *Datalog+* afin d'assurer la calculabilité. De plus, *Datalog±* a également apporté une nouvelle fonctionnalité: *les contraintes négatives* permettant d'interdire certaines combinaisons de faits. Veuillez noter que le nom **règles existentielles** fait référence au même formalisme que *Datalog±* et nous utiliserons les deux noms de manière interchangeable.

L'introduction de contraintes négatives engendre des conflits dans les bases de connaissances. La source de ces conflits est soit l'ensemble de faits, soit l'ensemble de règles. Dans le premier cas, nous disons qu'un ensemble de faits est *inconsistant* si l'application de l'ensemble de règles sur cet ensemble de faits spécifiques génère un conflit. Dans le deuxième cas, un ensemble de règles est dit *incohérent* si l'application de toutes les règles à un ensemble de faits conduira toujours à un conflit. La présence de conflits est problématique, car dans la logique classique, une fois la contradiction affirmée, toute proposition (ainsi que sa négation) peut en être déduite. Ceci est connu sous le nom d'explosion déductive (également appelée *ex falso quodlibet*). Pour résoudre les incohérences, Pollock [1987] a introduit le raisonnement "défaisable" dans lequel les faits et les règles peuvent être défaits et des "règles empiriques" ainsi que des préférences suffisent afin de rétablir des capacités de raisonnement satisfaisantes [García and Simari, 2004; Antoniou et al., 2000]. Dans cette thèse, nous nous limitons aux inconsistances et supposons que nous n'avons pas d'incohérences,

c'est-à-dire que l'ensemble des règles est cohérent. Afin de raisonner en présence d'inconsistances avec des règles existentielles, les deux approches principales sont les sémantiques de réparation [Lembo and Ruzzi, 2007] et l'argumentation basée sur la logique [García and Simari, 2004; Modgil and Prakken, 2014]. Les deux approches consistent à raisonner avec les "mondes" cohérents possibles, i.e. des sous ensembles de faits de la base logique qui n'engendrent pas de conflits.

D'une part, les approches se basant sur les sémantiques de réparation considèrent généralement les sous-ensembles maximaux de faits cohérents appelés *réparations* qu'ils manipulent à l'aide d'un modificateur (expansion, scission, etc.) et d'une stratégie d'inférence (intersection, universalité, etc.) pour répondre aux requêtes en présence d'inconsistances [Baget et al., 2016a]. D'autre part, les approches d'argumentation basées sur la logique sont des approches ascendantes qui consistent à instancier un graphe d'argumentation (c'est-à-dire, générer des arguments et des attaques entre eux) à partir de connaissances exprimées dans un langage particulier puis à utiliser une technique de raisonnement sur le graphe d'argumentation obtenu afin de rétablir la consistance [Amgoud, 2014]. Dans cette thèse, nous avons choisi de nous concentrer sur l'argumentation basée sur la logique en raison de son intuitivité pour l'utilisateur. En effet, les explications fournies par l'argumentation sont plus intuitives que celles fournies par les sémantiques de réparation [Arioua, 2016].

Quatre approches majeures ont été étudiées dans la littérature pour l'argumentation basée sur la logique: l'argumentation basée sur les hypothèses (ABA) [Bondarenko et al., 1993], ASPIC/ASPIC + [Modgil and Prakken, 2014], la programmation avec des logiques défaisables (DeLP) [García and Simari, 2004] et l'argumentation déductive [Besnard and Hunter, 2008]. Dans cette thèse, nous nous intéressons à l'argumentation déductive instancié avec des règles existentielles suivant les travaux de Croitoru and Vesic [2013] et de Arioua et al. [2017]. Dans ce contexte, la question de recherche à laquelle nous souhaitons répondre est la suivante:

Question de recherche

Comment peut-on raisonner avec l'argumentation basée sur la logique dans le contexte des règles existentielles?

Les graphes d'argumentation générés par des systèmes d'argumentation basés sur la logique ont été largement étudiés, en particulier pour caractériser leur capacité de réponses aux requêtes et de traitement des inconsistances. Néanmoins, peu de travaux on été menés en ce qui concerne la structure réelle de tels graphes ou sur la manière de construire ces graphes en pratique. Notre problème de recherche peut alors être reformulé en un sous-ensemble de questions de recherche plus précises comme suit:

Questions de recherche

- Quelles sont les caractéristiques particulières des graphes d'argumentation générés dans le contexte des règles existentielles ?
- Pouvons-nous appliquer les techniques basées sur les classements sur les graphes générés ?
- Peut-on générer efficacement des graphes d'argumentation dans le contexte des règles existentielles ? Peut-on fournir des outils à cet effet ?

Notre contribution peut être résumée en trois résultats principaux. Pour les graphes d'argumentation générés à partir de bases de données de règles existentielles, nous avons (1) amélioré la compréhension de leurs propriétés structurelles, (2) leur efficacité et (3) leur expressivité. En ce qui concerne la compréhension, nous avons proposé un ensemble de propriétés qui élargissent notre capacité à comprendre de tels graphes et développé des outils intuitifs pour leur génération. En ce qui concerne leur efficacité, nous avons expliqué comment générer un graphe d'argumentation par rapport à la structure de la base de connaissances sous-jacente et nous avons introduit une nouvelle manière de créer des hypergraphes d'argumentation. Nous montrons que ces hypergraphes d'argumentation permettent d'avoir une représentation des données plus compacte et plus efficace. En ce qui concerne l'expressivité des graphes d'argumentation, nous avons défini comment les sémantiques basées sur les classements devraient être utilisées avec des hypergraphes d'argumentation. Nos résultats sont disséminés dans les différents chapitres:

Dans le chapitre 2, nous présentons la théorie du domaine de la recherche dans lequel nous travaillons, à savoir le formalisme des règles existentielles et de la théorie de l'argumentation. Dans la section 2.1, nous introduisons formellement le cadre des règles existentielles ainsi que le chaînage Skolem, les différentes classes de règles décidables de *Datalog+* et les deux types de conflits (incohérence et inconsistance). Ensuite, dans la section 2.2, nous nous concentrons sur la théorie de l'argumentation et présentons une introduction aux approches les plus récentes en matière d'évaluation des arguments dans le contexte du cadre d'argumentation abstrait de Dung (approches basées sur les extensions, sur l'étiquetage et sur le classement).

Dans le chapitre 3, nous choisissons d'instancier des graphes à partir de bases de connaissances incohérentes exprimées avec des règles existentielles et nous revisitons le cadre d'argumentation de Croitoru and Vesic [2013]. Dans la section 3.1, nous présentons d'abord (1) des résultats généraux pour les graphes générés, tels que l'appartenance à des classes de graphes particulières ou l'existence de caractéristiques particulières (cycles et argu-

ments fondés sur les conflits) et ensuite, (2) des résultats plus précis pour les graphes d'argumentation générés à partir de bases de connaissances sans règles positives, tels qu'une caractérisation du nombre d'arguments factices ou de la composition de composantes fortement connexes. Puis, dans Section 3.2, nous montrons que la méthode naïve de génération des graphes d'argumentation peut être améliorée en utilisant un prétraitement de la base de connaissances ou un filtrage des arguments. Dans la section 3.3, nous présentons DAGGER, une application intuitive permettant de générer, visualiser et exporter un graphe d'argumentation à partir de bases de connaissances logiques. Enfin, dans la section 3.4, nous comparons les solveurs de la communauté d'argumentation et montrons que les graphes générés ont un impact conséquent sur les performances de ces solveurs. Cela montre la nécessité d'utiliser l'argumentation basée sur la logique dans les futures compétitions d'argumentation.

Dans le chapitre 4, nous présentons un système d'argumentation pour les règles existentielles basé sur les hypergraphes d'argumentation comme définis dans les travaux de Nielsen and Parsons [2007]. Dans la section 4.1, nous montrons que ce nouveau système d'argumentation répond à de nombreuses propriétés souhaitables (postulats de rationalité, coïncidence entre plusieurs sémantiques d'argumentation, etc.). Nous montrons aussi que ce système est plus efficace en ce qui concerne le temps de calcul, le nombre d'arguments et d'attaques grâce à une analyse empirique. Ensuite, dans la section 4.2, nous présentons l'outil Java NAKED permettant de générer et de visualiser ce nouveau système d'argumentation.

Dans le chapitre 5, nous décrivons comment les approches basées sur le classement peuvent être implémentées à la fois dans le cadre logique et dans le cadre de l'argumentation. Dans la section 5.1, nous montrons qu'en présence de redondances, les sémantiques basées sur le classement peuvent produire des résultats différents si elles sont utilisées sur un noyau (un sous-graphe où les arguments redondants sont supprimés) par rapport à quand elles sont utilisées sur le graphe d'argumentation d'origine. De plus, nous fournissons des conditions pour que le rang des arguments soit augmenté, diminué ou inchangé. Ensuite, dans la section 5.2, nous fournissons des instructions pour les sémantiques basées sur le classement dans le contexte des hypergraphes d'argumentation et nous présentons la première sémantique basée sur le classement pour ce type de graphes: la sémantique *nh-categoriser*. Enfin, dans la section 5.3, nous utilisons une approche similaire sans argumentation et définissons une approche pour le classement des ensembles de faits. Cette approche permet d'obtenir une réponse aux requêtes plus personnalisée sans prendre en compte l'ensemble des réparations.

Dans le chapitre 6, nous résumons nos contributions et présentons un certain nombre de problèmes de recherche futurs basés sur des extensions possibles de notre travail et de nos travaux publiés.

NOTE

Portions of this work have been published previously in:

- **Journals**
  - ▹ **Bruno Yun**, Pierre Bisquert, Patrice Buche, Madalina Croitoru, Valérie Guillard and Rallou Thomopoulos: Choice of environment-friendly food packagings through argumentation systems and preferences, Ecological Informatics (EI 2018). 24-36 (Impact factor: 1.82).
- **Conferences**
  - Core rank A^*:
    - ▹ **Bruno Yun**, Srdjan Vesic, Madalina Croitoru and Pierre Bisquert: Inconsistency Measures for Repair Semantics in OBDA, the 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence (IJCAI 2018). 1977-1983 (full paper).
    - ▹ **Bruno Yun**, Madalina Croitoru, Srdjan Vesic and Pierre Bisquert: Graph Theoretical Properties of Logic Based Argumentation Frameworks, 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018). 2148-2149 (short paper).
    - ▹ **Bruno Yun**, Madalina Croitoru and Patrice Buche: Are ranking semantics sensitive to the notion of core?, 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017). 943-951 (full paper).
    - ▹ **Bruno Yun**, Madalina Croitoru, Srdjan Vesic and Pierre Bisquert: DAGGER: Datalog+/- Argumentation Graph GEneRator, 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018). 1841-1843 (demo paper).
    - ▹ **Bruno Yun**, Madalina Croitoru and Srdjan Vesic: NAKED: N-Ary graphs from Knowledge bases Expressed in *Datalog*±, 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019). Accepted (demo paper).
  - Core rank A:
    - ▹ **Bruno Yun**, Srdjan Vesic, Madalina Croitoru, Pierre Bisquert and Rallou Thomopoulos: A Structural Benchmark For Logical Argumentation Frameworks, 20th International Symposium on Intelligent Data Analysis (IDA 2017). 334-346 (full paper).
  - Core rank B:
    - ▹ **Bruno Yun**, Rallou Thomopoulos, Pierre Bisquert and Madalina Croitoru: Defining argumentation attacks in practice: an experiment in food packaging consumer expectations, 23rd International Conference on Conceptual Structures (ICCS 2018). 73-87 (full paper).
  - Core rank C:

- Nikos Karanikolas, Madalina Croitoru, Pierre Bisquert, Christos Kaklamanis, Rallou Thomopoulos and **Bruno Yun**: Multi-criteria Decision Making with Existential Rules Using Repair Techniques, the 38th SGAI International Conference on Artificial Intelligence (SGAI 2018). 177-183 (short paper).

– Not ranked:

- **Bruno Yun**, Srdjan Vesic and Madalina Croitoru: Toward a More Efficient Generation of Structured Argumentation Graphs, the 7th International Conference on Computational Models of Argument (COMMA 2018). 205-212 (short paper).
- **Bruno Yun**, Srdjan Vesic, Madalina Croitoru and Pierre Bisquert: Viewpoints using ranking based argumentation semantics, the 7th International Conference on Computational Models of Argument (COMMA 2018). 381-392 (full paper).
- **Bruno Yun**, Pierre Bisquert, Patrice Buche and Madalina Croitoru: Arguing about end-of-life of packagings: Preferences to the Rescue, 10th International Conference on Metadata and Semantics Research (MTSR 2016). 119-131 (full paper).
- **Bruno Yun** and Madalina Croitoru: An Argumentation Workflow for Reasoning in Ontology Based Data Access, 6th International Conference on Computational Models of Argument (COMMA 2016). 61-68 (short paper).
- **Bruno Yun**, Madalina Croitoru and Srdjan Vesic: How to generate a benchmark of logical argumentation graphs?, the 7th International Conference on Computational Models of Argument (COMMA 2018). 475-476 (demo paper).

• **Workshops**

- **Bruno Yun**, Madalina Croitoru, Srdjan Vesic and Pierre Bisquert: Graph theoretical properties of logic based argumentation frameworks: proofs and general results, 5th Workshop on Graph Structures for Knowledge Representation and Reasoning (GKR@IJCAI 2018). 118-138 (full paper).

Contents

1 Introduction **1**

2 Preliminaries **13**
2.1 Existential rules framework 14
2.1.1 Logical language 14
2.1.2 Rules and reasoning 17
2.1.3 Chase and finite expansion set 20
2.1.4 Complexity classes 26
2.1.5 Incoherence and inconsistence 27
2.2 Argumentation theory 32
2.2.1 Abstract argumentation semantics 32
2.2.2 Extension-based approaches 33
2.2.3 Labelling approach 38
2.2.4 Ranking-based semantics 40
2.3 Summary 44

3 Using Deductive Argumentation with Existential Rules **47**
3.1 Deductive argumentation frameworks in existential rules 49
3.1.1 Argumentation graphs generated from knowledge bases 52
3.1.2 Argumentation graphs generated from knowledge bases without rules 56
3.2 Improving the argument generation 64
3.2.1 Optimisation for knowledge bases without rules 65
3.2.2 Optimisation for knowledge bases with rules 66
3.3 The DAGGER tool 73
3.3.1 DAGGER's architecture 74
3.3.2 Usability scenarios 74
3.4 Benchmarks on logic-based argumentation frameworks 78
3.4.1 Benchmark generation 79
3.4.2 Results of literature solvers over the benchmark 81
3.5 Summary 86

4 Argumentation Hypergraphs **89**
4.1 Argumentation hypergraphs with the existential rules language . 91

4.1.1 Hypergraph argumentation framework $\mathfrak{F}^\star$ 92
4.1.2 Argumentation framework properties 94
4.1.3 Rationality postulates 99
4.1.4 Empirical analysis 100
4.2 The NAKED tool 106
4.2.1 The argument and attack generation 107
4.2.2 The structure of NAKED 108
4.2.3 Usability scenarios 110
4.3 Summary 112

5 Ranking-Based Reasoning 113
5.1 Ranking with existential rules deductive argumentation framework 116
5.1.1 Core equivalence 117
5.1.2 Characterising ranking changes 126
5.2 Ranking-based semantics with argumentation hypergraphs . . . 131
5.2.1 Properties for ranking-based semantics on hypergraphs . 132
5.2.2 The nh-categoriser 136
5.3 Ranking facts in inconsistent knowledge bases 142
5.3.1 The ranking-based inference framework 143
5.3.2 $\mathcal{RIF}$ results 148
5.4 Summary 156

6 Conclusion 159
6.1 Scope 160
6.2 Summary and contributions 161
6.3 Perspectives 163

7 Appendix i
7.1 Miscellaneous i
7.2 Proofs ii
7.2.1 Chapter 3 ii
7.2.2 Chapter 4 viii
7.2.3 Chapter 5 xii

List of Figures

1.1 Image of a spiny anteater 4
1.2 Representation of the first possible world 5
1.3 Representation of the second possible world 6
1.4 Representation of the third possible world 6

2.1 Abstract and known concrete classes of existential rules . . . 24
2.2 Known concrete FES classes and chases finiteness 25
2.3 Argumentation framework of Example 2.13 32
2.4 Argumentation framework of Example 2.17 37
2.5 Inclusion relations between argumentation semantics 37
2.6 Labelling that corresponds to $\{a, c\}$ of Example 2.17 39
2.7 Labelling that corresponds to $\{a\}$ of Example 2.17 39
2.8 An argumentation framework $\mathfrak{F}$ 41

3.1 Representation of a 2-*copy* graph 58
3.2 Structural properties of argumentation frameworks generated from simple knowledge bases 63
3.3 Approach workflow for optimising the argument generation . 64
3.4 Three steps reconstruction using k-copy graphs 66
3.5 The 3-layer structure of DAGGER 75
3.6 Repair computation module of DAGGER 76
3.7 Argumentation module of DAGGER 77
3.8 Screen capture of the argumentation graph interface of the DAGGER tool 78
3.9 Representation of the argumentation graph corresponding to the knowledge base b_{44} 81

4.1 Generation time comparison between $\mathfrak{F}_{\mathcal{KB}}$ and $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ for set A 101
4.2 Generation time comparison between $\mathfrak{F}_{\mathcal{KB}}$ and $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ for set B 102
4.3 Comparison of the number of arguments between $\mathfrak{F}_{\mathcal{KB}}$ and $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ for set A 102
4.4 Comparison of the number of arguments between $\mathfrak{F}_{\mathcal{KB}}$ and $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ for set B 103
4.5 Comparison of the number of attacks between $\mathfrak{F}_{\mathcal{KB}}$ and $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ for set A 103

4.6 Comparison of the number of attacks between $\mathfrak{F}_{\mathcal{KB}}$ and $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ for set B 104
4.7 An argumentation hypergraph about packagings. 107
4.8 Representation of hyperedges in NAKED 109
4.9 Representation of the areas of interest in NAKED 110
4.10 Argument highlight feature in NAKED 111

5.1 Representation of the core c_1 of $\mathfrak{F}_{\mathcal{KB}}$ using $\bowtie_1$ and displayed in Table 5.1 121
5.2 Representation of the core c_2 of $\mathfrak{F}_{\mathcal{KB}}$ using $\bowtie_2$ and displayed in Table 5.1 122
5.3 Representation of an argumentation framework $\mathfrak{F}$ and one of its cores c' 128
5.4 Argumentation framework with equal h-categoriser scores . . 141
5.5 Another argumentation framework with equal h-categoriser scores 141
5.6 Representation of the $\mathcal{RIF}$ workflow 145
5.7 Representation of the considered packagings 153

List of Tables

1.1 Reading guide by topics 11

2.1 Complexity of CQ entailment for studied Skolem-FES concrete classes 27
2.2 Argumentation semantics with respect to criteria 38

3.1 Classes of argumentation frameworks studied in the literature 52
3.2 Images of the permutation σ on X 60
3.3 Arguments in $\mathfrak{F}_{\mathcal{KB}}$ obtained from the knowledge base of Example 3.23 70
3.4 Characteristics of the $\mathfrak{F}_{\mathcal{KB}}$ and $\mathfrak{F}^*_{\mathcal{KB}}$ generated from the knowledge bases. 73
3.5 Characteristics of the small knowledge bases 80
3.6 Average computational time for small instances 82
3.7 Ranking of solvers for the generated small graphs 83
3.8 Number of timeouts for the generated large graphs 83
3.9 Ranking of solvers for the generated large graphs 83
3.10 Rankings extracted from the ICCMA 2015 website 85
3.11 Normalised Kendall's tau distance between the rankings of the generated graphs and the competition ranking 85

4.1 Comparison of the median number of arguments, attacks and generation time needed between $\mathfrak{F}^\star{}_{\mathcal{KB}}$ and $\mathfrak{F}_{\mathcal{KB}}$ on the sets of knowledge bases A_1, A_2, A_3 and B 101

5.1 Some arguments constructed from the knowledge base of Example 5.12 and two particular cores obtained using $\bowtie_1$ and $\bowtie_2$. 120
5.2 Ranking on arguments of c_1 using the burden-based (and discussion-based) ranking-based semantics 124
5.3 Ranking on arguments of c_2 using the burden-based (and discussion-based) ranking-based semantics 124
5.4 Ranking K using the ranking on arguments of Table 5.2 . . . 124
5.5 Ranking K using the ranking on arguments of Table 5.3 . . . 124
5.6 Ranking on repairs 155

1 Introduction

This thesis presents an original research in the field of *Knowledge Representation and Reasoning* (KRR), one of the main sub-domains in *Artificial Intelligence* (AI). The knowledge representation language we address is the existential rules framework, a family of logical languages corresponding to the positive conjunctive existential fragment of FOL. The contribution of the thesis is the investigation of reasoning techniques using argumentation graphs generated from inconsistent knowledge bases expressed in the existential rules language.

In the reminder of this chapter, we put in context this research vision with respect to our work. The chapter is structured as follows. In Section 1.1, we introduce the general context of the existential rules framework and the types of conflicts that may arise. In Section 1.2, we show the classical techniques for reasoning in the presence of inconsistencies. In Section 1.3, we recall the existing approaches for logic-based argumentation in the literature. Section 1.4 discusses the research question and our contributions. Finally, in Section 1.5, we conclude and highlight the structure of this thesis.

1.1 The existential rules framework

In the field of knowledge representation and reasoning, we thrive for formalisms that provide a high-level description of the world that can be effectively used to build intelligent applications. In this context, "intelligent" refers to the ability of a system to find implicit consequences of its explicitly represented knowledge.

In its simplest form, data can be stored explicitly as statements in various manners, for instance in a relational database [Codd, 1970], an RDF triple store or a graph database. However, in order to get this "implicit knowledge" from the stored data, both the database and knowledge representation communities acknowledged the need to limit complexity and organise data into information and knowledge. Thus, deductive database with ontologies were created to infer implicit data, hence palliating incompleteness in classical databases. The introduction of ontologies also enabled the enrichment

and unification of vocabularies in multiple data sources. In the Semantic Web, ontological knowledge is often represented with formalisms based on Description Logics (DLs), a family of formal KRR languages used for describing and reasoning about the relevant concepts of an application domain [Baader et al., 2005]. However, DLs have a restricted expressivity: beside the fact that they support only unary and binary predicates, the ontological knowledge can only be described in terms of tree structures (i.e. no cycles are allowed). Moreover, conjunctive query answering with classical DLs, such as the $\mathcal{ALCI}$ DL, has appeared to be extremely complex (it is 2ExpTime-complete in combined complexity, and still NP-complete in the size of the data). Hence, there has been a trend to using so-called lightweight DLs for which query answering is tractable and particular attention has been paid to DL-Lite and the $\mathcal{EL}$ DL [Calvanese et al., 2007] which possess the notable property that query answering can be reduced to evaluation of standard database queries. Please note that these DLs form the core of the so-called tractable profiles of the Semantic Web language OWL 2.

Meanwhile, the Deductive Database language Datalog [Gallaire and Nicolas, 1987; Ceri et al., 1989] which is a syntactic subset of Prolog [Colmerauer and Roussel, 1996] have been praised as the default language for query databases. It is not only more expressive than regular relational databases but also possesses desirable features for query answering since Datalog queries on finite sets are guaranteed to terminate. However, in this language, all variables in the rule head necessarily occur in the rule body. Therefore, we can only produce knowledge about already known individuals and cannot infer the existence of unknown individuals. This is a crucial feature since we cannot assume that all individuals are known in advance.

The *Datalog+* language have been proposed to meet these two requirements, i.e. value invention and the ability to express high-level structures such as n-ary predicates or more complex ontologies. The *Datalog+* language is a more expressive extension of the Datalog language [Calì et al., 2013] with the introduction of existential quantifiers in head rules, allowing for the deduction of unknown individuals as in "if x is married then there exists an unknown individual y such that x and y are married". The introduction of existential quantifiers in the head rules led to many problems for the tractability of query answering given the need to deal with large data sets. In order to avoid this tractability problem, the *Datalog±* family of languages [Calì et al., 2013, 2009] was introduced. The *Datalog±* family corresponds to the set of languages that restrict the rule syntax in order to achieve tractability. Moreover, *Datalog±* also brings a new feature: *falsum* ($\perp$) in rule heads in order to forbid certain combinations of facts. Such a rule is called a negative constraint and models some sort of logical negation in the human reasoning like "x cannot be married and unmarried at the same time". These additional features made it possible for *Datalog±* to generalise disparate other KR formalisms such as plain Datalog and a variety of De-

scription Logics families, in particular, DL-Lite and $\mathcal{EL}$[Calì et al., 2010a]. Please note that the name **existential rules framework** refer to the same formalism as *Datalog*± and we will use the two names interchangeably.

The introduction of negative constraints gives rise to conflicts in the knowledge representation. The source of those conflicts is either the set of facts or the set of rules. In the former, we say that a set of facts is *inconsistent* if applying the set of rules on that specific set of facts generates *falsum*. In the latter, a set of rules is said to be *incoherent* if applying all of the rules to any set of facts will always lead to *falsum*.

Example 1.1 (Inconsistency). *Consider the following situation where we want to know if John is married or not. Suppose that there are no means by which we can verify the truthfulness and reliability of the factual knowledge.*

- *Factual knowledge: There is a piece of evidence* e_1 *proving that John is married and there is another piece of evidence* e_2 *proving that John is unmarried.*
- *Rules: If there is an evidence that a person is married than he is married. Likewise, If there is an evidence that a person is unmarried, then he is unmarried.*
- *Negative constraint: It is not possible for a person to be married and unmarried at the same time.*

The set of factual knowledge is inconsistent with respect to the set of rules since we can generate generate both the facts that John is married and unmarried (given evidence e_1 *and* e_2*). However, the set of rules is coherent as we can find a set of facts such that all rules can be applied and no contradiction are generated. For instance, there is an evidence proving that John is married and an evidence proving that Alice is unmarried.*

We now give an example of an incoherent set of rules.

Example 1.2 (Incoherence). *Consider the following set of rules: spiny anteaters are mammals, mammals do not lay eggs, spiny anteaters lay eggs, and one cannot lay eggs and not lay eggs at the same time. Any set of factual knowledge on which all these rules are applicable will always lead to contradiction (lay eggs and not lay eggs).*

Please note that Flouris et al. [2006] showed that inconsistency can be viewed as a form of incoherence. The presence of conflicts is problematic, as in classical logic, once a contradiction has been asserted, any proposition (including their negations) can be inferred from it. This is known as deductive explosion (also called *ex falso quodlibet*). For instance if ϕ and $\neg\phi$ are true then, for any ψ it holds that $\phi \vee \psi$ is true because ϕ is true. But since ϕ is false and $\phi \vee \psi$ is true, then ψ has to be true.

Figure 1.1: Image of a spiny anteater

In order to solve the incoherences, Pollock [1987] introduced Defeasible Reasoning where both facts and rules can be defeated and "rules of thumb" along with preferences are enough to choose the preferred outcome [García and Simari, 2004; Antoniou et al., 2000]. In this thesis, we restrict ourselves to inconsistencies and we make the assumption that we do not have incoherences, i.e. the set of rules is coherent. In the next section, we show how to deal with inconsistencies and give a brief introduction on the main approaches.

1.2 Inconsistency-tolerant reasoning

In order to reason in presence of inconsistencies within the context of the existential rules framework, the two main approaches are Repair Semantics [Lembo and Ruzzi, 2007] and Logic-Based Argumentation [García and Simari, 2004; Modgil and Prakken, 2014; Toni, 2014; Bondarenko et al., 1993; Besnard and Hunter, 2001]. Both approaches consist in reasoning with possible consistent "worlds".

On the one hand, the repair semantics approaches generally consider *Maximally Consistent Subsets* (MCS) of facts called *repairs* that they manipulate using a modifier (expansion, splitting, etc.) and an inference strategy (intersection, universality, etc.) for answering queries in presence of inconsistencies [Baget et al., 2016a]. Please note that the algorithms for repair semantics in practical applications do not compute all of the repairs but are based on rewriting queries [Lembo et al., 2015].

Example 1.3 (Repair semantics). *Consider the following inconsistent situation about the presence of a person in Anthony's house last night. The situation is modelled using a set of facts, a set of rules and a set of negative constraints. We suppose that there are no means by which we can verify the truthfulness and reliability of the factual knowledge. We have the following set of facts:*

f1: Anthony's neighbour heard his washing machine running last night.

f2: Rebecca had dinner with Anthony at his house last night.

f3: Rebecca was depressed and stayed at her house last night.

f4: There was no electricity in Anthony's whole neighbourhood last night.

f5: Last night, Anthony posted a selfie of himself at his house.

f6: Last night, it was a full moon.

And the following set of rules:

r1: If the washing machine was running at X's house, then there was a person at X's house.

r2: If there was no electricity at X's house, then there was no person at X's house.

r3: If there was a selfie of someone at X's house, then there was a person at X's house.

r4: If someone had dinner with another person at X's house, then there was a person at X's house.

And the following set of constraints:

n1: It is not possible that there is a person at X's house and no person at X's house at the same time.

n2: One cannot be at two different houses at the same time.

We can clearly see that the set of factual knowledge is inconsistent since we can deduce that there was no person at Anthony's house (using f4 and r2) and that there was someone at Anthony's house last night (using f1 and r1). In order to cope with the inconsistencies, the repair semantics consider the following set of repairs (the possible worlds).

- *Rebecca was depressed and stayed at her house, there was no electricity at Anthony's house and it was a full moon* $(\{f_3, f_4, f_6\})$.

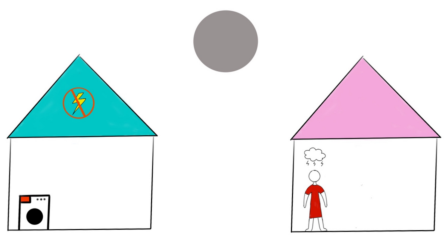

Figure 1.2: Representation of the first possible world

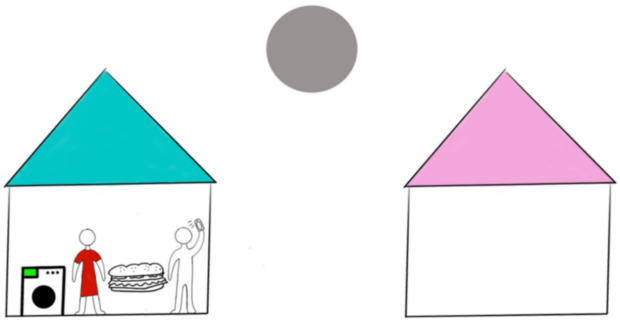

Figure 1.3: Representation of the second possible world

- *At Anthony's house, the washing machine was running, Rebecca had dinner with Anthony, Anthony took a selfie and it was a full moon* $(\{f_1, f_2, f_5, f_6\})$.
- *At Anthony's house, the washing machine was running and Anthony took a selfie of himself at his house, Rebecca was depressed and stayed at her house and it was a full moon* $(\{f_1, f_3, f_5, f_6\})$.

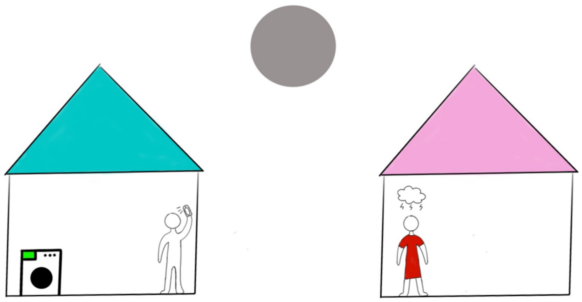

Figure 1.4: Representation of the third possible world

If we consider the repair semantics that answer positively to a query if and only if it is entailed by all the repairs then we will get a negative answer to the query "was there someone in Anthony's house?" because in the first repair, we have no proof that someone was in Anthony's house. However, we will get a positive answer to the query "was there a full moon last night?".

On the other hand, the logic-based argumentation approaches are bottom-up approaches that consist in first instantiating an argumentation framework (i.e. generating arguments and attacks, see next section for an explanation) from knowledge expressed in a particular language and then, applying a reasoning technique on the resulting argumentation graph in order to restore consistency [Amgoud, 2014].

In this thesis, we choose to focus on logic-based argumentation because of its intuitiveness with respect to explanations compared to repair semantics [Arioua, 2016] and its reusability in many domains such as debates [Leite and Martins, 2011], decision-making [Gordon and Karacapilidis, 1997; Bonet and Geffner, 1996] or persuasion [Amgoud et al., 2000; Prakken, 2006; Hadjinikolis et al., 2013].

1.3 Logic-based argumentation

When working with logic-based argumentation, there are two main questions that should be answered:

1. How do we generate the argumentation framework from the data?
2. What are the reasoning techniques that can be used on an argumentation framework?

In order to answer those questions, we have to study how to build arguments from a knowledge base using a given logic. Roughly speaking, a logic has two main components: a logical language (a set of well-formed formulae) and a consequence operator that can draw conclusions from a set of formulae. In the literature, we can distinguish two main families of approaches that are used in logic-based argumentation: The first family contains approaches where arguments are built from Tarskian logics [Tarski, 1936], while the second family of approaches use rule-based logics for constructing arguments. The Tarskian logic is an abstract logic, i.e. it generalises several concrete logics that respect some axioms on the consequence operator (expansion, idempotence, finitude, absurdity and coherence) whereas, rule-based logics [Amgoud et al., 2004] are logics that usually encode two types of rules (strict rules which encode certain knowledge and defeasible rules which encode uncertain knowledge) and the consequence mechanism shows how these rules can be chained. The distinction between the strict and defeasible rules is that a rule is "defeasible if it can be blocked or defeated in some way" [Gabbay et al., 1993]. There are four major logical approaches that have been studied in the literature: *Assumption-Based Argumentation* frameworks (ABA) [Bondarenko et al., 1993], *ASPIC/ASPIC+* [Modgil and Prakken, 2014], *Defeasible Logic Programming* (DeLP) [García and Simari, 2004] and *Deductive argumentation* [Besnard and Hunter, 2008]. The first three approaches are systems based on rule-based logics whereas the fourth is more oriented toward Classical Logics where an arguments is seen as a pair with a hypothesis and a conclusion. We now briefly introduce each of the aforementioned argumentation frameworks and show how they manage to reason in presence of conflicting knowledge.

The ABA framework, as defined by Toni [2014], is composed of a language, a set of rules, a set of assumptions (they are observations or defeasible

premises) and a contrariness function for explaining the reason against each assumptions. The language used by the ABA framework is left abstract (it can be any language of sentences) while the contrariness can be seen as a "handle" to attack the weak point of an assumption. In ABA, the arguments are deductions of claims using the rules and supported by assumptions whereas the attack is defined as follows: an argument a attacks another arguments b if and only if the conclusion of a is the contrary of one of the assumptions of b. Several computational techniques have been defined for ABA [Toni, 2013; Dung et al., 2006], extensions with preferences have been proposed [Bao et al., 2017] and online tools[1] are available. We end this short introduction to ABA by noting that the reasoning techniques for this framework are called *ABA semantics* and can be viewed from two perspectives: the argument and the assumption perspectives. In the first perspective, the standard semantics for abstract argumentation, defined by Dung [1995] in his seminal paper, are applied to ABA. These semantics can compute "acceptable" or "winning" arguments with respect to the attacks in the argumentation graph. In the second perspective, it considers that sets of assumptions can attack each other and sets of "winning" assumptions can be computed in the same fashion. Correspondence between these two perspectives are the results of the work of Dung et al. [2007] and Toni [2012] and enabled a jump in performance.

The ASPIC+ framework was created in the context of the European ASPIC project as a general framework for generating arguments that accommodates a broad range of instantiating logics. As defined by Modgil and Prakken [2014], an argumentation system is composed of a logical language closed under negation, a set of rules (strict and defeasible) and a labelling function that associates a sentence to some defeasible rules. Please note that, inspiring themselves from the work of Bondarenko et al. [1993] on ABA, Modgil and Prakken [2013] lifted the closure under negation condition on the language by further adding a particular contrariness function that associates a set of reasons against each sentences of the language. In the ASPIC+ framework, an argument can either be a sentence of the language or built upon other arguments according to the rule applications. This framework also includes three types of attacks, namely undercutting (attack on a defeasible deduction based on the aforementioned labelling function), rebutting (attacks on a weak conclusion) and undermining (attack on a premise). The framework also accommodates with preferences on arguments using the notion of defeat (or successful attack). As a preference relation on arguments is somewhat harder to obtain than an ordering on elements of the language or rules, the framework comes with ways to "lift" these orderings in order to obtain a preference relation on arguments. An online tool for

[1] Such as `http://www-abaplus.doc.ic.ac.uk`.

ASPIC+, called TOAST, has been developed and is available[2]. We end this short introduction to ASPIC+ by a little discussion on the reasoning techniques for this framework. As stated by Modgil and Prakken [2014], the Dung's extensions of this framework respect the four rationality postulates defined by Caminada and Amgoud [2007] (Sub-argument closure, Closure under Strict Rules, Direct Consistency, Indirect Consistency). This allows ASPIC+ to restore the consistency of the data it is built on. Please note that although some say that the framework violates the consistency in some cases, patches have been proposed by Modgil and Prakken [2014].

The *Defeasible Logic Programming* (DeLP) is an argumentation formalism for deciding between contradictory goals. As defined originally by García and Simari [2004], a *Defeasible Logic Program* is originally composed of a set of fact (ground atoms or negated ground atoms), a set of strict rules and a set of defeasible rules. It has to be noted that this formalism only considers ground rules and that the set of strict rules are considered coherent and consistent with the set of facts. In DeLP, an *argument structure* (or simply argument) is a minimal non-contradictory set of defeasible rules that leads to the deduction of a specific fact. Therefore, a *sub-argument structure* of an argument structure is simply an argument structure with a "smaller" (for set inclusion) set of defeasible rules. Thus, argument structures in DeLP are usually represented with triangles containing smaller triangles representing their sub-argument structures. The attack considered in DeLP is as follows: an argument a that leads to h_1 attacks an other argument b if and only if there is a sub-argument c of b that leads to h_2 and $\{h_1, h_2\}$ are conflicting when put together with the facts and strict rules. This formalism also includes two ways for comparing arguments by using either the specificity criterion, based on the work of Poole [1985], or an explicit preference relation defined among defeasible rules. The comparison criterion among arguments is used in the same fashion as ASPIC+, i.e. using the notion of defeaters (successful attacks). An online tool has been developed and is available[3] and concrete applications have been studied [García et al., 2000]. We end this short introduction by briefly explaining one of the ways for reasoning in the presence of inconsistencies in DeLP. First, the arguments are built and attacks are computed. Second, a comparison criterion for arguments is chosen and defeated arguments are computed. At this point, the result is usually a tree-like graph called "Dialectical tree" where nodes are argument structures and links are defeat relations. Third, the nodes of a dialectical tree are marked in bottom-up process (from the leaves to the root). Fourth, the marking on the root is used for query answering.

The Deductive argumentation, as originally defined by Besnard and Hunter [2008], considers arguments as a pair (H, c) where H is a set of as-

[2]See `http://toast.arg-tech.org`.
[3]See `http://lidia.cs.uns.edu.ar/DeLP`.

sumptions called the premises (or equivalently, the support) of the argument and c is the conclusion (or equivalently, the claim or the consequent) of the argument. In the literature, deductive argumentation have been used in two main approaches: generating argumentation frameworks directly over the abstract Tarskian logic [Amgoud and Besnard, 2009, 2010] or over a concrete Tarskian logic such as propositional logics [Amgoud and Cayrol, 1998; Besnard and Hunter, 2001] or first-order logic [Besnard and Hunter, 2001, 2008, 2014]. In this thesis, we are interested in the second approach. More precisely, we will study deductive argumentation frameworks built over the existential rules framework following the work of Croitoru and Vesic [2013], Arioua et al. [2017] and Arioua [2016]. In these argumentation frameworks, although much effort has been spent to show the desirable behaviour with query answering and inconsistency handling, there is not a lot of work on the actual structure of such graphs or how to practically construct these argumentation frameworks.

1.4 Research problem and contributions

Against this background, the research question we want to answer is:

Research Question

How can we reason using logic-based argumentation in the context of the existential rules framework?

Our research problem can be reformulated into a subset of more precise research questions as follows:

Research Questions

- Are the argumentation frameworks generated with the existential rules particular?
- Can we characterise and apply extension-based and ranking-based techniques on the generated graphs?
- How can we efficiently generate argumentation graphs within the context of existential rules? Can we provide tools?

Our contribution focused on improving (1) the understanding of the structural properties of argumentation graphs generated from existential rules databases, (2) their efficiency and (3) their expressivity. Regarding the understanding, we have a proposed a set of properties that extends our grasp of such graphs and developed intuitive tools for their generation. Re-

Topic	Binary graphs	Hypergraphs
Generation	Chapter 3 Published in [Yun et al., 2018c, 2017b]	Chapter 4 Published in [Yun et al., 2019]
Reasoning with extensions	Chapter 2, Chapter 3 Published in [Yun et al., 2018b]	Chapter 4
Reasoning with rankings	Chapter 5 Published in [Yun et al., 2017a]	Chapter 5

Table 1.1: Reading guide by topics

garding their efficiency, we gave insights as to how one should generate an argumentation graph with respect to the structure of the underlying knowledge base and introduced a more efficient hypergraph-based argumentation framework. Regarding the argumentation graph expressivity, we paved the way by first defining how ranking-based semantics should be used with hypergraph argumentation frameworks.

1.5 Thesis organisation

We chose to present the research that we did in the past three years while at LIRMM by selecting some of our results submitted or published in international conferences. Each chapter is constructed around results that we find important in order to give a relevant representation of our research interests. Table 1.1 summarises the organisation of the thesis. Please note that the presentation of our works was done "a posteriori" and does not coincide with their chronological publication dates.

Chapter 2. In this chapter, we present the theoretical domain of research that we are working in, namely the formalism of the existential rules framework and the argumentation theory. In Section 2.1, we formally introduce the existential rules framework along with the frontier chase, the several decidable classes of rules and the two types of conflicts (incoherence and inconsistence). Then, in Section 2.2, we focus on the argumentation theory and show an introduction to the state-of-the-art existing approaches for evaluating arguments in the context of Dung's abstract argumentation framework (extension-based, labelling-based and ranking-based approaches).

Chapter 3. In this chapter, we chose to instantiate graphs using inconsistent existential rules knowledge bases and to revisit the argumentation framework of Croitoru and Vesic [2013]. In Section 3.1, we first present both (1) general results for the generated graphs such that the membership to particular graph classes or the existence of particular features (cycles and conflict-based arguments) and (2) more focused results for argumen-

tation graphs generated from knowledge bases without positive rules such that a characterisation of the number of dummy arguments or compositions of strongly connected components. Then, in Section 3.2, we show that the naive method for generating the argumentation framework can be improved using either a preprocessing of the knowledge base or a filtration of arguments. In Section 3.3, we introduce DAGGER as an intuitive application for generating, visualising and exporting argumentation graph from logical knowledge bases. Lastly, in Section 3.4, we benchmark state-of-the-art solvers from the argumentation community and show that the generated graphs impact the performance of solvers. This shows the necessity for logic-based argumentation graphs in the upcoming argumentation competitions and why some solvers can outperform other solvers on these graphs. This chapter builds upon the published work of Yun et al. [2017b, 2018c,b,f].

Chapter 4. In this chapter, we present a new argumentation framework for the existential rules framework based on the hypergraph argumentation framework defined in the work of Nielsen and Parsons [2007]. In Section 4.1, we show that this new argumentation framework satisfies many desirable properties (the rationality postulates, coincidence between argumentation semantics, etc.) and is more efficient using an empirical analysis with respect to computational time, number of argument and attacks. Then, in Section 4.2, we introduce the NAKED java tool for generating and visualising this new argumentation framework. This chapter builds upon the published work of Yun et al. [2019].

Chapter 5. In this chapter, we present how ranking-based approaches can be implemented in both the argumentation and the logical setting. In Section 5.1, we show that in the presence of redundancies, ranking-based semantics may yield different results if they are used on a core (a subgraph where redundant arguments are removed) compared to when they are used on the original argumentation framework. Moreover, we provide conditions for the rank of arguments to be increased, decreased or unchanged. Then, in Section 5.2, we provide guidelines for ranking-based semantics in the context of argumentation hypergraphs and introduce the first ranking-based semantics for such type of graphs: *the nh-categoriser*. Finally, in Section 5.3, we use a similar approach without argumentation and define a framework for ranking sets of facts. This framework enables to get a more focused query-answering without considering the set of all repairs. This chapter builds upon the published work of Yun et al. [2017a, 2018g].

Chapter 6. This chapter summarises our contributions, and presents a number of interesting future research problems based on possible extensions of both this work and the published work of Yun et al. [2016, 2018e,a,h].

2
Preliminaries

2.1 Existential rules framework **14**

2.1.1 Logical language 14

2.1.2 Rules and reasoning 17

2.1.3 Chase and finite expansion set 20

2.1.4 Complexity classes 26

2.1.5 Incoherence and inconsistence 27

2.2 Argumentation theory **32**

2.2.1 Abstract argumentation semantics 32

2.2.2 Extension-based approaches 33

2.2.3 Labelling approach 38

2.2.4 Ranking-based semantics 40

2.3 Summary **44**

In this chapter, we discuss two main problems of the knowledge representation field: ensuring the termination of the reasoning mechanism and reasoning in presence of conflicting knowledge.

In knowledge representation, we always thrive to find a good balance between the expressiveness and the computational tractability, since a higher expressiveness might lead to an infinite reasoning. In this section, we provide an introduction to knowledge representation with the existential rules logical framework and to its forward chaining inference mechanism called "chase". We recall the reasoning problem of *query entailment* and the notion of finite classes of existential rules for which the chase is guaranteed to halt. Since knowledge databases might contain conflicts, we introduce the two types of conflicts (inconsistence and incoherence) and discuss how the argumentation theory manages to handle inconsistencies using several evaluation methods.

Research Questions in this Chapter

- *How is knowledge represented using existential rules and how can we reason with this knowledge?*
- *What are the different types of conflict (inconsistence and incoherence) and how can we maintain the ability to reason in presence of conflicts?*
- *How can we handle conflicting knowledge in the context of abstract argumentation theory?*

2.1 Existential rules framework

The goal of knowledge representation and reasoning is to model human-level intelligence and reasoning faculties. According to Levesque and Brachman [1987], the biggest dilemma in this case is the trade-off between expressiveness and computational tractability of a given logical language. *Datalog+* is a first-order logical language that extends plain Datalog with value invention. It has the ability to express knowledge about "unknown" individuals (e.g. "every bird comes from an egg" this specific egg might be unknown but its existence still holds). This level of expressiveness comes at the high cost of undecidability (the reasoning mechanism can be infinite), that is why different decidable fragments of *Datalog+* have been defined under the name of *Datalog±* [Calì et al., 2012] which is a generalisation of Datalog [Ceri et al., 1989] and certain fragments of Description Logics [Baader et al., 2005].

2.1.1 Logical language

We consider a first-order logical (FOL) language $\mathcal{L}$ with no function symbols (except for constants) built with the existential and universal quantifiers $(\exists, \forall)$ and the implication and conjunction connectives $(\rightarrow, \wedge)$ on a vocabulary $\mathcal{V}oc = (\mathcal{P}, \mathcal{C})$ composed of a finite set of predicates $\mathcal{P}$ and a potentially infinite set of constants $\mathcal{C}$. Each predicate $p \in \mathcal{P}$ is associated with a positive integer which is called the *arity* of p. We are also given an infinite set of variables $\mathcal{V}$, and an infinite set of existential "fresh" variables $Nulls$ (called "nulls", which act as placeholders for unknown constants, and can thus be seen as variables). We denote variables by uppercase letters $X, Y, Z, etc.$, constants by lowercase letters $a, b, c, etc.$, and fresh variables (nulls) by $Null_1, Null_2, etc.$

A logical language is a symbolic representation of some knowledge about the world. For these symbols to have meaning, they need to be "mapped" to elements of the world. This is done using an *interpretation* function

which maps predicates and constants symbols to elements of the domain of interpretation.

Definition 2.1 (Interpretation). *An interpretation of a logical language $\mathcal{L}$ built on a vocabulary $\mathcal{V}oc = (\mathcal{P}, \mathcal{C})$ is a pair $(\mathcal{D}, \mathcal{I})$ where $\mathcal{D}$ is a non-empty set called the interpretation domain and $\mathcal{I}$ is an interpretation function of the symbols of $\mathcal{L}$ such that:*

1. *for each constant $c \in \mathcal{C}$, $\mathcal{I}(c) \in \mathcal{D}$.*
2. *for each predicate $p \in \mathcal{P}$ of arity k, $\mathcal{I}(p) \subseteq \mathcal{D}^k$.*
3. *for each pair (c, c') of distinct constants in $\mathcal{C}$, $\mathcal{I}(c) \neq \mathcal{I}(c')$.*

The third item in the above definition is called the *unique name assumption* and indicates that different constants should be interpreted by different elements of the interpretation domain. This assumption is often made in knowledge representation and Baget et al. [2011] showed that as long as equality between constants is not considered (which is the case in this thesis), adopting the unique name assumption or not does not make any difference in the considered reasoning tasks.

Knowledge about the world is expressed using formulas built from the logical language. The basic building blocks are called atomic formulas (or atoms).

Definition 2.2 (Atom). *An atom over the vocabulary $\mathcal{V}oc$ is of the form $p(t_1, \ldots, t_k)$, where $p \in \mathcal{P}$ is a predicate of arity k and $t_i \in \mathcal{V} \cup \mathcal{C} \cup \mathit{Nulls}$ is either a variable, a constant, or a null.*

Given a formula Φ built on a language $\mathcal{L}$, we note $\mathit{terms}(\Phi)$ and $\mathit{vars}(\Phi)$ respectively the terms and variables (including nulls) occurring in Φ. $\top$ (tautology) and $\bot$ (falsity) are allowed and considered themselves atoms. A *ground atom* contains only constants.

Example 2.1 (Atoms, conjunctions, and interpretations). *Consider the following vocabulary $\mathcal{P} = \{p, q\}$, $\mathcal{C} = \{a, b\}$, then "$\exists X\, p(a, X)$" is an atom, "$p(a, b)$" is a ground atom, and "$\exists X\, (p(a, X) \wedge q(X))$" is a conjunction of atoms. An interpretation might map "p" to the concept of parenthood, "$p(a, b)$" might be interpreted as the individual "a" is a parent of "b" (e.g. Adam is a parent of Bruno). "$\exists X\, p(a, X)$" might be interpreted as there exists an unknown individual such that "a" is its parent.*

A basic form of knowledge is factual knowledge which is represented using *facts*. Usually a fact is a ground atom, however to account for knowledge expressing the existence of unknown constants (nulls), the definition of fact is generalised to an atom that contains constants or nulls (existentially quantified variables).

Definition 2.3 (Fact). *A fact on a language $\mathcal{L}$ is an existentially closed atom on $\mathcal{L}$. A closed atom is of the form $\exists \vec{X}\, p(\vec{a}, \vec{X})$ where $p \in \mathcal{P}$ is a predicate, $\vec{a}$ is a finite (potentially empty) set of constants, and $\vec{X}$ is a finite (potentially empty) set of existentially quantified variables.*

Please note that for the purposes of this thesis, *a fact is not a conjunction.* To be able to manipulate conjunctions as sets of facts, existential variables are represented using nulls.

Notation 2.1 (From existential variables to nulls). *An existential variable can be represented as a "fresh" Skolem term by removing the existential quantifier and replacing the variable with a null. This null has to be "fresh" (or "safe") meaning that it has not been used before. For example, $\exists X\, p(a, X)$ can be represented as $p(a, Null_1)$ as long as $Null_1$ is fresh (i.e. $Null_1$ has not been used before).*

Notation 2.2 (From conjunctions to sets of facts). *A conjunction of facts can be represented as a set of facts by removing the existential quantifier and replacing the variables with nulls. For example, $\exists X\, (p(a, X) \wedge q(X))$ can be represented as the set $\{p(a, Null_1),\ q(Null_1)\}$ assuming $Null_1$ is fresh.*

A **model** of a formula built on $\mathcal{L}$ is an interpretation of $\mathcal{L}$ that makes this formula true by considering the classical interpretation of logical connectives and quantifiers.

Definition 2.4 (Logical consequence and equivalence). *Given a language $\mathcal{L}$ and two formulas Φ_1 and Φ_2 on $\mathcal{L}$, Φ_2 is a (logical) consequence of Φ_1 (denoted $\Phi_1 \vDash \Phi_2$) if all models of Φ_1 are models of Φ_2. Φ_1 and Φ_2 are said to be logically equivalent (denoted $\Phi_1 \equiv \Phi_2$) if $\Phi_1 \vDash \Phi_2$ and $\Phi_2 \vDash \Phi_1$.*

One of the relevant problems in knowledge representation is the *entailment* problem, which is asking whether a formula is a consequence of another formula. This can be expressed on facts as follows: *given two facts f_1 and f_2, is it true that f_2 is a consequence of f_1 (i.e. $f_1 \vDash f_2$)?* It is well known that $f_1 \vDash f_2$ if and only if there exists a *homomorphism* from f_2 to f_1 [Baget et al., 2011].

Definition 2.5 (Substitution and homomorphism). *Let $\vec{X}$ be a set of variables and $\vec{T}$ a set of terms. a substitution of $\vec{X}$ to $\vec{T}$ is a mapping form $\vec{X}$ to $\vec{T}$ (notation $\vec{X} \rightarrow \vec{T}$). A homomorphism π from an atom a_1 to an atom a_2 is a substitution of $vars(a_1)$ to $terms(a_2)$ such that $\pi(a_1) = a_2$.*[1] *A homomorphism π from a set of atoms S to a set of atoms S' is a substitution of $vars(S)$ to $terms(S')$ such that $\pi(S) \subseteq S'$.*

[1] By abuse of notation, the resulting atom after the substitution is applied is denoted $\pi(a_1)$. Thus, $\pi(a_1) = a_2$ means that applying the substitution π to the variables of a_1 produces the atom a_2.

Example 2.2 (Homomorphism). *The atom $p(a, Null_1)$ can be mapped to the atom $p(a, b)$ by the homomorphism $\pi = \{Null_1 \rightarrow b\}$ that substitutes $Null_1$ by b. Therefore $p(a, b) \vDash p(a, Null_1)$.*

Notation 2.3 (Homomorphism restriction $\pi|_{\vec{X}}$). *Given a homomorphism π, we denote by $dom(\pi)$ the domain of π. Given a set of variables $\vec{X}$, we denoted the restriction of π to $\vec{X}$ by $\pi|_{\vec{X}} = \{(X, \pi(X)) \mid X \in dom(\pi) \cap \vec{X}\}$.*

The entailment problem is generally expressed using *queries* (query answering problem), specifically conjunctive queries which are an existentially closed conjunctions of atoms. These can be seen as asking if there is a set of constants and nulls that make an existentially closed conjunction of atoms a consequence of the set of facts.

Definition 2.6 (Query). *A Conjunctive Query (CQ) is an existentially closed conjunction of atoms of the form $Q(\vec{X}) = \exists \vec{Y}\, \Phi(\vec{X}, \vec{Y})$, where $\vec{X}$ is a set of variables, $\vec{Y}$ is a set of existential variables (possibly with constants) and Φ is a conjunction of atoms. A Boolean Conjunctive Query (BCQ) is a conjunctive query of the form $Q() = \exists \vec{Y}\, \Phi(\vec{Y})$.*

The answers to a conjunctive query $Q(\vec{X}) = \exists \vec{Y}\, \Phi(\vec{X}, \vec{Y})$ over a set of formulas $\mathcal{F}$ is the set of all tuples (constants and nulls) that if, substituted with $\vec{X}$ and $\vec{Y}$, make Φ a consequence of $\mathcal{F}$. The answer to a boolean conjunctive query is either *true* or *false*, and it is *true* over a set of facts $\mathcal{F}$ if and only if it is a consequence of $\mathcal{F}$, otherwise it is *false*.

Example 2.3 (Conjunctive and boolean queries). *Consider the query $Q(X) = \exists Y\, p(X, Y)$, the answers to this query over the set of facts $\mathcal{F} = \{p(a, b), p(c, Null_1)\}$ are $\{a, c\}$ because there is a homomorphism $\pi_1 = \{X \rightarrow a, Y \rightarrow b\}$ from Q to $p(a, b)$, and there is a homomorphism $\pi_2 = \{X \rightarrow c, Y \rightarrow Null_1\}$ to $p(c, Null_1)$. The answer to the BCQ $Q() = \exists X, Y\, p(X, Y)$ is true (because it can be mapped to $\mathcal{F}$ using π_1 or π_2).*

2.1.2 Rules and reasoning

Rules (or equivalently, positive rules) are formulas that allow the enrichment of a set of facts with new deduced knowledge. These rules generally encode domain-specific implications, for example "if X is a cat then X is an animal". Existential rules [Baget et al., 2011] are general rules that account for unknown individuals, they are also known as Tuple Generating Dependencies (TGD) [Abiteboul et al., 1995], Conceptual Graphs rules [Salvat and Mugnier, 1996; Sowa, 1976], Datalog$^{\exists}$ rules [Calì et al., 2013], etc.

Definition 2.7 (Existential rule). *An existential rule (or simply a rule) r is a formula of the form $\forall \vec{X}, \vec{Y}\ \left(\mathcal{B}(\vec{X}, \vec{Y}) \rightarrow \exists \vec{Z}\ \mathcal{H}(\vec{X}, \vec{Z})\right)$ where $\vec{X}, \vec{Y}$ are tuples of variables, $\vec{Z}$ is a tuple of existential variables, and $\mathcal{B}$, $\mathcal{H}$ are*

finite non-empty conjunctions of atoms respectively called body and head of r *and denoted* $Body(r)$ *and* $Head(r)$*. The frontier of* r *(denoted* $fr(r)$*) is the set of variables occurring in both the body and the head of* r *i.e.* $fr(r) = vars(Body(r)) \cap vars(Head(r))$*.*

Rules are used to *infer* new knowledge starting from an initial set of facts based on the notion of *rule application.*

Definition 2.8 (Rule application). *A rule* r *is said to be applicable to a set of facts* $\mathcal{F}$ *if there is a homomorphism* π *from* $Body(r)$ *to* $\mathcal{F}$*. In that case, the application of* r *to* $\mathcal{F}$ *according to* π *(denoted* $\alpha(\mathcal{F}, r, \pi)$*) adds to* $\mathcal{F}$ *a set of facts* $\pi^{safe}(Head(r))$ *where* π^{safe} *ensures that existential variables are replaced with fresh nulls.*

Example 2.4 (Rule application). *Consider the rule* r *stating that two sisters are the daughters of the same parent. This rules is modelled as:* $\forall X, Y\, sisterOf(X, Y) \rightarrow \exists Z\, daughterOf(X, Z) \wedge daughterOf(Y, Z)$*. This rule* r *is applicable to the set* $\mathcal{F} = \{sisterOf(alice, barbara)\}$ *using the homomorphism* $\pi = \{X \rightarrow alice, Y \rightarrow barbara\}$ *and therefore its application produces* $\alpha(\mathcal{F}, r, \pi) = \mathcal{F} \cup \{daughterOf(alice, Null_1),\ daughterOf(barbara, Null_1)\}$ *assuming* $Null_1$ *is safe.*

Notation 2.4 (Rules with atomic head). *In general, rules might have a conjunction of atoms in their head, however for the purposes of this thesis, we only consider rules with one atom in their head. Baget et al. [2011] showed that any set of rules can be transformed to a set of rules with atomic head. The idea is that conjunctions can be split using intermediary atoms. For example, the rule* $\forall X, Y\, sisterOf(X, Y) \rightarrow \exists Z\, daughterOf(X, Z) \wedge daughterOf(Y, Z)$ *can be transformed to a set of three rules with atomic heads:*

1. $\forall X, Y\, sisterOf(X, Y) \rightarrow \exists Z\, d(X, Y, Z)$
2. $\forall X, Y, Z\, d(X, Y, Z) \rightarrow daughterOf(X, Z)$
3. $\forall X, Y, Z\, d(X, Y, Z) \rightarrow daughterOf(Y, Z)$

A rule is applicable on a set of facts if there is a homomorphism from the body of the rule to this set of facts, furthermore, a rule might not be applicable right away but could become applicable after some new knowledge is generated by another rule, which might make another rule applicable and so on. This sequence of rule applications is called a *derivation.* Normally, a derivation is a sequence of facts generated at each rule application, however, we generalize this notion to include the rule and the homomorphism used at each step.

Definition 2.9 (Derivation). *Given a set of facts $\mathcal{F}$ and a set of rules $\mathcal{R}$, a derivation of $\mathcal{F}$ with respect to $\mathcal{R}$ is a (potentially infinite) sequence δ of D_i s.t. D_i is a tuple $(\mathcal{F}_i, r_i, \pi_i)$ composed of a set of facts $\mathcal{F}_i$, a rule r_i and a homomorphism π_i from $Body(r_i)$ to $\mathcal{F}_i$ where: $D_0 = (\mathcal{F}_0 = \mathcal{F}, \emptyset, \emptyset)$, and $\mathcal{F}_i = \alpha(\mathcal{F}_{i-1}, r_i, \pi_i)$. We denote by $Facts(D_i)$, $Rule(D_i)$, and $Homo(D_i)$ the set of facts, rule and homomorphism of a tuple D_i.*

A derivation can be infinite as a rule can be applied again and again without restrictions as shown in the following Example 2.5.

Example 2.5 (Derivation). *Consider the set of facts $\mathcal{F}$ stating that Bruno is a male human, and the rules $\mathcal{R}$ stating that any human has a parent and that a male human is a man.*

- $\mathcal{F} = \{human(bruno),\ male(bruno)\}$.
- $\mathcal{R} = \{r_1 : \forall X\, human(X) \rightarrow \exists Y\, parentOf(Y, X),$
  $r_2 : \forall X\, male(X) \wedge human(X) \rightarrow man(X)\}$.

A possible derivation of $\mathcal{F}$ w.r.t $\mathcal{R}$ is:

$$\begin{aligned}\delta = \langle &(\mathcal{F}, \emptyset, \emptyset), (\mathcal{F}_1 = \mathcal{F}_0 \cup \{man(bruno)\}, r_2, \pi_1 = \{X \rightarrow bruno\}),\\ &(\mathcal{F}_2 = \mathcal{F}_1 \cup \{parentOf(Null_1, bruno)\}, r_1, \pi_2 = \{X \rightarrow bruno\}),\\ &(\mathcal{F}_3 = \mathcal{F}_2 \cup \{man(bruno)\}, r_2, \pi_3 = \{X \rightarrow bruno\}),\\ &(\mathcal{F}_4 = \mathcal{F}_3 \cup \{parentOf(Null_2, bruno)\}, r_1, \pi_4 = \{X \rightarrow bruno\}),\\ &\ldots\rangle.\end{aligned}$$

As we can see in this example, this derivation apply successively the two rules r_1 and r_2 even though it produces the same atoms using the same homomorphisms.

A derivation for a specific fact f is a finite minimal sequence of rule applications starting from a set of facts $\mathcal{F}$ and ending with a rule application that generates f.

Definition 2.10 (Derivation for a fact). *Given some sets of facts $\mathcal{F}$ and rules $\mathcal{R}$, a derivation for a fact f is a finite derivation $\delta = \langle D_0, \ldots, D_n\rangle$ of $\mathcal{F}' \subseteq \mathcal{F}$ with respect to $\mathcal{R}$ such that:*

1. *$f \in Facts(D_n)$ (i.e. the last rule application contains f).*
2. *δ is minimal i.e. there is no derivation $\delta' = \langle D'_0, \ldots, D'_m\rangle$ for f such that:*
   - *$Facts(D'_0) \subset Facts(D_0)$ and*
   - *$\bigcup_{D' \in \delta'}(Rule(D'), Homo(D')) \subset \bigcup_{D \in \delta}(Rule(D), Homo(D))$.*

Example 2.6 (Derivation for a fact). *Consider the previous Example 2.5, a derivation from $\mathcal{F}$ to $man(bruno)$ is the sequence:*

$$\delta_1 = \langle(\mathcal{F}_0 = \{human(bruno), male(bruno)\}, \emptyset, \emptyset),$$
$$(\mathcal{F}_1 = \mathcal{F}_0 \cup \{man(bruno)\}, r_2, \pi_1)\rangle.$$

Query answering over a set of facts and rules can be done by generating all possible knowledge then finding homomorphisms from the queries to this "saturated" set of facts. In order to generate this knowledge, rules are applied in a breadth first manner. A breadth-first derivation is obtained by considering at each "breadth-first" step all possible rule applications on the current set of facts and applying them all before moving to the next step.

Definition 2.11 (Breadth-first derivation). *Given a set of facts $\mathcal{F}$ and a set of rules $\mathcal{R}$, a breadth-first derivation of $\mathcal{F}$ with respect to $\mathcal{R}$ is a derivation $\delta = \langle(\mathcal{F}_0 = \mathcal{F}, \emptyset, \emptyset), \ldots, (\mathcal{F}_i, r_i, \pi_i), \ldots\rangle$ such that for all $i < j$, if $(\mathcal{F}_{i+1} \backslash \mathcal{F}_i) \cap \pi_j(Body(r_j)) \neq \emptyset$ then for all $k > j$, $\pi_k(Body(r_k)) \nsubseteq \mathcal{F}_i$.*

The above definition ensures that if a rule is applied on some atoms generated by a rule application $i+1$ then no rule application afterwards can use *only* the atoms in $\mathcal{F}_i$. Intuitively, once we go to the next breadth-first step, we cannot apply a rule that could have been applied in a previous step according to the same homomorphism.

An exhaustive breadth-first derivation ensures that all rules have been applied according to all possible homomorphisms. An exhaustive derivation may be infinite and might contain "redundant" rule applications, however removing these "redundant" rule applications might make the exhaustive derivation finite. The role of a *chase* is to remove rule applications that it considers redundant.

2.1.3 Chase and finite expansion set

In order to answer queries over a set of facts and rules, the exhaustive derivation has to be finite. A chase is a mechanism that takes an exhaustive derivation and removes what it considers "redundant" rule applications using a *derivation reducer*. We use the formalization of Rocher [2016] for its simplicity to define a derivation reducer and a chase.

Definition 2.12 (Derivation reducer). *Given a set of facts $\mathcal{F}$ and a set of rules $\mathcal{R}$, a derivation reducer σ is a function that takes a rule application tuple $D_i = (\mathcal{F}_i, r_i, \pi_i)$ in a derivation $\delta = \langle D_0, \ldots, D_i, \ldots\rangle$ of $\mathcal{F}$ with respect to $\mathcal{R}$ and returns a rule applications tuple $\sigma(D_i) = (\mathcal{F}'_i, r_i, \pi_i)$ such that $\mathcal{F}'_i \equiv \mathcal{F}_i$.*

Definition 2.13 (σ-chase). *Given a set of facts $\mathcal{F}$, a set of rules $\mathcal{R}$, a derivation reducer σ, and an exhaustive breadth first derivation $\delta = \langle D_0, \ldots, D_i, \ldots\rangle$ of $\mathcal{F}$ with respect to $\mathcal{R}$: $\sigma\text{-}chase(\mathcal{F}, \mathcal{R}) = \langle\sigma(D_0), \ldots, \sigma(D_i), \ldots\rangle$ and $\sigma(D_i) \in \sigma\text{-}chase(\mathcal{F}, \mathcal{R})$ if and only if $Facts(\sigma(D_i)) \neq Facts(\sigma(D_{i-1}))$.*

The above definition ensures that only non redundant "meaningful" rule applications are kept (i.e. rule applications that generate something new according to the derivation reducer). A chase is finite if there is a breadth-first rule application step k such that for all D_j at step k, no new facts are generated [Baget et al., 2014b].

Applying a chase on a set of facts $\mathcal{F}$ and a set of rules $\mathcal{R}$ generates the saturated set of facts $\mathcal{F}^*$ that contains all initial and generated facts.

Definition 2.14 (Saturated set of facts). *Given a set of facts $\mathcal{F}$ and a set of rules $\mathcal{R}$, the saturation (or equivalently, closure) of $\mathcal{F}$ is $Sat_{\mathcal{R}}(\mathcal{F}) = \bigcup_{D \in \sigma\text{-}chase(\mathcal{F},\mathcal{R})} Facts(D)$. We also refer to $Sat_{\mathcal{R}}(\mathcal{F})$ by $\mathcal{F}^*$ when the set of rules $\mathcal{R}$ is obvious.*

Saturating a set of facts $\mathcal{F}$ with a set of rules $\mathcal{R}$ until no new rule application is possible allows us to obtain the *universal model*. The particularity of this model is that it is representative of all models of $(\mathcal{F} \cup \mathcal{R})$ (we denote the set of models of $(\mathcal{F} \cup \mathcal{R})$ by $models(\mathcal{F}, \mathcal{R})$).

Definition 2.15 (Universal model). *Given a set of facts $\mathcal{F}$ and a set of rules $\mathcal{R}$, a universal model M of $(\mathcal{F} \cup \mathcal{R})$ is a model of $(\mathcal{F} \cup \mathcal{R})$ such that for all models M' of $(\mathcal{F} \cup \mathcal{R})$, there is a homomorphism from M to M'.*

It is not always possible to obtain the universal model (the saturated set of facts might be infinite), however if the chase is finite then the model of the saturated set of facts is a universal model [Baget et al., 2011]. Therefore query entailment can be expressed using the notion of chase.

Theorem 2.1 (Query entailment and chase [Baget et al., 2011]). *Let us consider a set of facts $\mathcal{F}$, a set of rules $\mathcal{R}$ and a Boolean conjunctive query Q. If $\sigma\text{-}chase(\mathcal{F}, \mathcal{R})$ is finite then, $(\mathcal{F} \cup \mathcal{R}) \models Q$ if and only if $Facts(\sigma\text{-}chase(\mathcal{F}, \mathcal{R})) \models Q$.*

Different kinds of chases can be defined using different derivation reducers. Each derivation reducer ensures a universal model if its chase is finite. The most common chase is the *Frontier chase* [Baget et al., 2011], it yields equivalent results as the well-known *Skolem chase* [Marnette, 2009] that relies on a "skolemisation" of the rules by replacing each occurrence of an existential variable Y with a functional term $f_Y^r(\vec{X})$, where $\vec{X} = fr(r)$ are the frontier variables of r. Frontier chase and skolem chase yield isomorphic results [Baget et al., 2014a], in the sense that they generate exactly the same atoms, up to a bijective renaming of nulls by skolem terms.

The frontier chase considers two rule applications redundant if their mapping of the frontier variables are the same for the same rule.

Definition 2.16 (Frontier/Skolem chase). *The frontier chase σ_{fr}-chase (equivalent to the Skolem chase relies on the frontier derivation reducer (denoted by σ_{fr}) defined as follows. For any derivation δ, $\sigma_{fr}(D_0) = D_0$ and for*

every $D_i = (\mathcal{F}_i, r_i, \pi_i) \in \delta$*:*

$$Facts(\sigma_{fr}(D_i)) = \begin{cases} \mathcal{F}_{i-1} \cup \pi_i^{safe}(Head(r_i)) & \text{if for every } j < i \text{ with } r_i = r_j, \\ & \pi_j|_{fr(r_j)}(Body(r_j)) \neq \pi_i|_{fr(r_i)}(Body(r_i)) \\ \mathcal{F}_{i-1} & \text{otherwise} \end{cases}$$

Example 2.7 (Frontier chase). *Consider the following set of facts* $\mathcal{F}$ *and set of rules* $\mathcal{R}$*, inspired from the thesis of Hecham [2018], stating that an* ***animal shelter*** *would keep a dog found alone if it has an owner. If it has a collar or a microchip then it has an owner. A dog named "Rex" with a collar and a microchip is found alone.*

- $\mathcal{F} = \{alone(rex),\ hasCollar(rex),\ hasMicrochip(rex)\}$
- $\mathcal{R} = \{r_1 : \forall X, Y\ hasOwner(X, Y) \rightarrow keep(X),$
  $r_2 : \forall X\ alone(X) \wedge hasCollar(X) \rightarrow \exists Y\ hasOwner(X, Y),$
  $r_3 : \forall X\ alone(X) \wedge hasMicrochip(X) \rightarrow \exists Y\ hasOwner(X, Y)\}$

A possible frontier chase of $\mathcal{F}$ *and* $\mathcal{R}$ *is:*

$$\begin{aligned} \sigma_{fr}\text{-}chase(\mathcal{F}, \mathcal{R}) = \langle (\mathcal{F}, \emptyset, \emptyset), \\ & (\mathcal{F}_1 = \mathcal{F} \cup \{hasOwner(rex, Null_1)\}, r_2, \pi_1 = \{X \rightarrow rex\}), \\ & (\mathcal{F}_2 = \mathcal{F}_1 \cup \{hasOwner(rex, Null_2)\}, r_3, \pi_2 = \{X \rightarrow rex\}), \\ & (\mathcal{F}_3 = \mathcal{F}_2 \cup \{keep(rex)\}, r_1, \pi_3 = \{X \rightarrow rex, Y \rightarrow Null_1\})\rangle. \end{aligned}$$

First, r_2 *is applied on* $\{alone(rex),\ hasCollar(rex)\}$ *and generates* $\exists Y\ hasOwner(rex, Y)$ *which is not redundant since* r_2 *has never been applied before, therefore* $\mathcal{F}_1 = \mathcal{F}_0 \cup \{hasOwner(rex, Null_1)\}$*. Then* r_3 *is applied on* $\{alone(rex),\ hasMicrochip(rex)\}$ *and generates* $\exists Y\ hasOwner(rex, Y)$ *which is also not redundant because* r_3 *has never been applied before (even if it generates the same atom as* r_2*), therefore* $\mathcal{F}_2 = \mathcal{F}_1 \cup \{hasOwner(rex, Null_2)\}$*.*

Afterwards, r_1 *is applied on* $\{hasOwner(rex, Null_1)\}$ *and generates* $\{keep(rex)\}$ *which is not redundant as* r_1 *has never been applied before, therefore* $\mathcal{F}_3 = \mathcal{F}_2 \cup \{keep(rex)\}$*. Finally,* r_1 *is applied on the set of facts* $\{hasOwner(rex, Null_2)\}$ *with the homomorphism* $\pi_4 = \{X \rightarrow rex, Y \rightarrow Null_2\}$ *and generates* $\{keep(rex)\}$ *which is redundant since this rule application reuses the same rule and the same frontier mapping as the rule application on* $\{hasOwner(rex, Null_1)\}$ *(i.e.* $\pi_4|_{fr(r_1)} = \pi_3|_{fr(r_1)} = \{X \rightarrow rex\}$*). Since any additional rule application would be redundant (all rules have been applied with all possible homomorphisms) the frontier chase stops.*

Even if the frontier reducer removes some redundant rule applications, the frontier chase might be infinite as shown in the following Example 2.8.

Example 2.8 (Infinite frontier chase). *Consider the set of fact* $\mathcal{F}$ *and the set of rules* $\mathcal{R}$ *containing one fact and one rule.*

- $\mathcal{F} = \{p(a)\}$
- $\mathcal{R} = \{r_1 : \forall X\, p(X) \rightarrow \exists Y\, p(Y)\}$

A possible frontier chase of $\mathcal{F}$ and $\mathcal{R}$ is:

$$\begin{aligned}\sigma_{fr}\text{-}chase(\mathcal{F},\mathcal{R}) = \langle &(\mathcal{F},\emptyset,\emptyset), (\mathcal{F}_1 = \mathcal{F} \cup \{p(Null_1)\}, r_1, \pi_1 = \{X \rightarrow a\}),\\ &(\mathcal{F}_2 = \mathcal{F}_1 \cup \{p(Null_2)\}, r_1, \pi_2 = \{X \rightarrow Null_1\}),\\ &(\mathcal{F}_3 = \mathcal{F}_2 \cup \{p(Null_3)\}, r_1, \pi_2 = \{X \rightarrow Null_2\}), \ldots \rangle.\end{aligned}$$

First, r_1 is applied using π_1 and generates $\exists Y\, p(Y)$ which is not redundant since r_1 has never been applied before, therefore $\mathcal{F}_1 = \mathcal{F}_0 \cup \{p(Null_1)\}$. Then r_1 is applied on $\{p(Null_1)\}$ using π_2 and generates $\exists Y\, p(Y)$ which is not redundant since $\pi_2|_{fr(r_1)} = \{X \rightarrow Null_1\} \neq \pi_1|_{fr(r_1)} = \{X \rightarrow a\}$, therefore $\mathcal{F}_2 = \mathcal{F}_1 \cup \{p(Null_2)\}$, and so on infinitely.

Some derivation reducers are "stronger" than others, this implies that their chase might stop in cases where others do not. This is known as the reducer order relation.

Definition 2.17 (Reducer order relation [Rocher, 2016]). *Given two derivation reducers σ_1 and σ_2, we say that σ_1 is weaker than σ_2 (denoted by $\sigma_1 \leq \sigma_2$) if for any set of rules $\mathcal{R}$ and set of facts $\mathcal{F}$, if σ_1-chase is finite then σ_2-chase is also finite. Furthermore, we say that σ_1 is strictly weaker than σ_2 if $\sigma_1 \leq \sigma_2$ and $\sigma_2 \nleq \sigma_1$.*

In the literature, there are four well-known types of chase: the Oblivious chase (σ_{obl}-*chase*) [Calì et al., 2013], the Skolem/Frontier chase (σ_{fr}-*chase*) [Marnette, 2009; Baget et al., 2011], the Restricted chase (σ_{res}-*chase*) [Fagin et al., 2005], and the Core chase (σ_{core}-*chase*) [Deutsch et al., 2008].

Proposition 2.1 (Chases finiteness order [Onet, 2013; Rocher, 2016]). *The following relations hold: $\sigma_{obl} \leq \sigma_{fr} \leq \sigma_{res} \leq \sigma_{core}$.*

It is well-known that query entailment using a chase is undecidable (the chase might be infinite) [Beeri and Vardi, 1981] even under strong restrictions such as using a single rule or restricting to binary predicates with no constants. However, some restrictions on the set of rules can ensure decidability for a specific type of chase. These restrictions are classified into three big categories known as "*abstract classes*". The first one is "*Finite Expansion Set*" (FES) [Baget et al., 2014b] that ensures that a finite universal model of the knowledge base exists and can be generated using a chase. For each chase we can define its FES class: *oblivious-FES*, *skolem-FES*, *restricted-FES*, *and core-FES*. The second class is called "*Finite Unification Set*" (FUS) [Baget et al., 2011] which guarantees that some backward chaining method halts. Finally, the class called "*Greedy Bounded Treewidth Set*"

(GBTS) [Baget et al., 2011] ensures that the potentially infinite universal model of a knowledge base has a bounded treewidth. Each abstract class has a set of "*concrete classes*" that classifies rules based on their syntactic properties e.g. the concrete class *Datalog* describes rules that do not contain existentially quantified variables. The following Figure 2.1 shows the most studied concrete classes in the literature and the relation between them: an upward edge going from a class C to a class C' means that any set of rules in class C is also in class C'.

In this thesis we rely mainly on the frontier chase to reason with existential rules, for simplicity we will only give examples and intuitions about concrete classes of skolem-FES.[2] Restricting ourselves to the frontier chase and subsequently to the skolem-FES classes of rules is not a very restrictive constraint since most studied concrete FES classes are skolem-FES (cf. Figure 2.2).

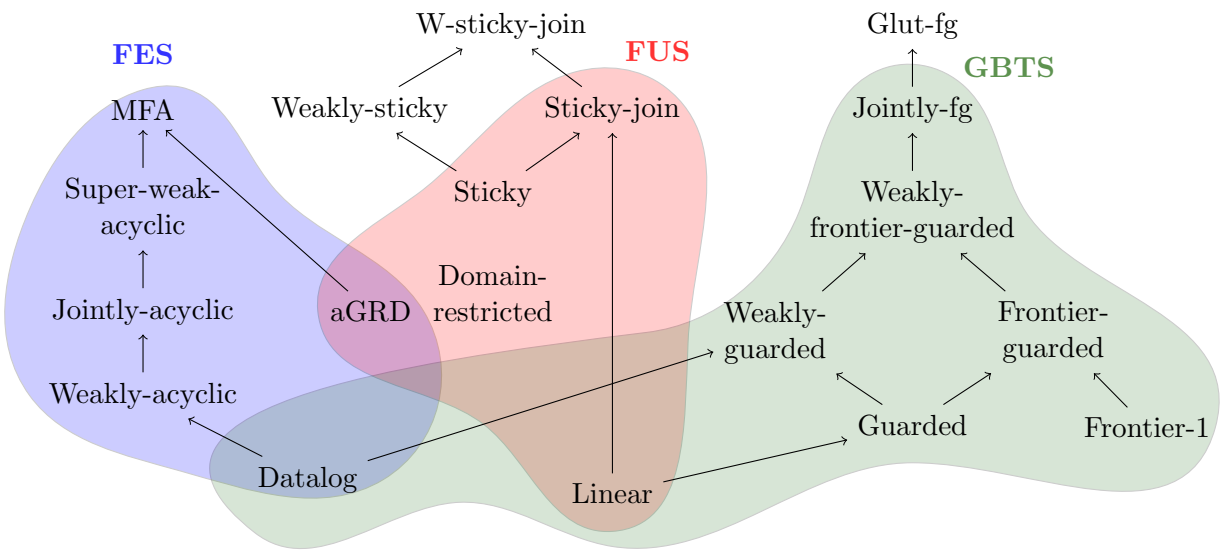

Figure 2.1: Abstract and known concrete classes of existential rules [Baget et al., 2011; Rocher, 2016]

A concrete class is simply a syntactic distinction of rules. The most basic skolem-FES concrete class is the **Datalog class** (also known as Range Restricted [Abiteboul et al., 1995]) which are rules without the existential quantifier. Another simple class is the **aGRD class** (Acyclic Graph of Rule Dependency) [Baget et al., 2014a]. A Graph of Rule Dependency is a directed graph that encodes possible interactions between rules: the

[2]For more information about these concrete classes, see the work of Baget et al. [2011]. The online tool Kiabora `http://graphik-team.github.io/graal/downloads/kiabora-online` checks automatically if a set of rules is skolem-FES.

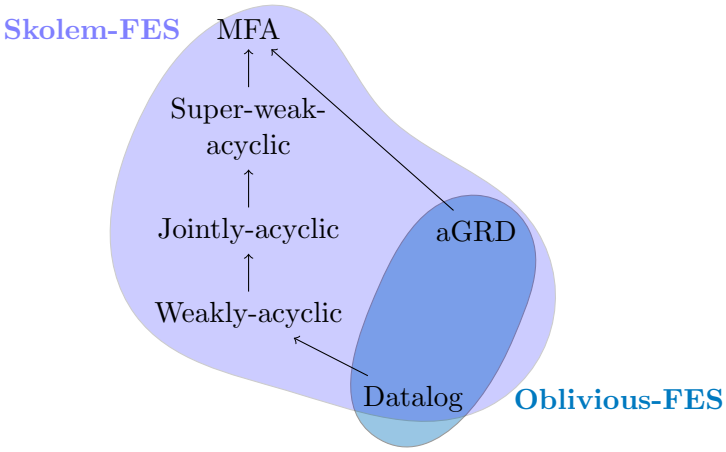

Figure 2.2: Known concrete FES classes and chases finiteness (all skolem-FES concrete classes are restricted-FES and core-FES

nodes represent the rules and there is an edge from a node r_1 to r_2 if and only if an application of the rule r_1 may create a new application of the rule r_2. A GRD is acyclic when it has no circuit. The notions of "*weak acyclicity*" [Marnette, 2009] and "*joint acyclicity*" [Krötzsch and Rudolph, 2011] are based on the position of the predicate and the existential and frontier variables. The **MFA** class (Model Faithful Acyclicity) [Grau et al., 2013] relies on detecting a specific set of facts called *critical instance*. The following Example 2.9 provides some rules that are skolem-FES.

Example 2.9 (Skolem-FES rules). *Consider the following sets of rules:*

- $\mathcal{R}_1 = \{\forall X, Y, Z\, p(X,Z) \wedge p(Z,Y) \rightarrow p(X,Y)\}$ *is range restricted (Datalog).*
- $\mathcal{R}_2 = \{\forall X, Y\, siblingOf(X,Y) \rightarrow \exists Z\, parentOf(Z,X,Y)\}$ *is aGRD.*
- $\mathcal{R}_3 = \{r_1 : \forall X, Y\, p(X,Y) \rightarrow \exists Z\, r(Y,Z),$
  $r_2 : \forall X, Y\, r(X,Y) \rightarrow p(Y,X)\}$. $\{r_1\}$ *is aGRD and* $\{r_2\}$ *is range restricted, however* $\mathcal{R}_3$ *is weakly-acyclic and is neither aGRD nor range-restricted.*
- $\mathcal{R}_4 = \{r_1 : \forall X, Y\, p(X,Y) \rightarrow \exists Z\, r(Y,Z),$
  $r_2 : \forall X, Y\, r(X,Y) \wedge r(Y,X) \rightarrow p(X,Y)\}$. $\{r_1\}$ *is aGRD and* $\{r_2\}$ *is range restricted, however* $\mathcal{R}_4$ *is Jointly-acyclic.*
- $\mathcal{R}_5 = \{r_1 : \forall X q(X) \rightarrow \exists Y\, p(X,Y) \wedge p(Y,X) \wedge p(X,X),$
  $r_2 : \forall X\, p(X,X) \rightarrow r(X),$
  $r_3 : \forall X\, r(X) \rightarrow q(X)\}$. $\{r_1\}$ *alone is aGRD,* $\{r_2, r_3\}$ *is range restricted, however* $\mathcal{R}_5$ *is super-weakly-acyclic.*

- $\mathcal{R}_6 = \{\forall X, Y\, p(X, Y) \rightarrow \exists Z, T\, q(Y, Z) \wedge p(Z, T)\}$ *is model-faithful-acyclic.*

Not all concrete classes are created equal, some might have higher complexity for query answering, and applying a chase on these classes would require more time. In the next section we recall the definitions for some complexity classes and describe the complexity of CQ entailment for the skolem-FES concrete classes.

2.1.4 Complexity classes

Complexity is an indication of a computational problem inherent difficulty. We briefly recall the definitions of the complexity classes by increasing complexity. For more details about complexity theory, the reader is referred to the work of Papadimitriou [1994].

Definition 2.18 ($\mathbf{AC}_0$). *A problem is in* AC_0 *if it can be solved by a boolean circuit of bounded depth with a polynomial number of AND and OR gates.*

Definition 2.19 (Polynomial time (PTime)). *A problem is in* PTIME *if it can be solved by a deterministic Turing machine running in polynomial time in the input.*

Definition 2.20 (NP). *A problem is in* NP *if it can be solved by a non-deterministic Turing machine running in polynomial time in the input.*

Definition 2.21 (coNP). *A problem is in* coNP *if its complement is in the class* NP*, meaning that there is a polynomial-time algorithm that can verify no instances (counterexamples) using a non-deterministic Turing machine.*

Definition 2.22 (Exponential time (ExpTime)). *A problem is in the* EXPTIME *class if it can be solved by a deterministic Turing machine running in simple exponential time ($2^{p(n)}$) in the input.* 2EXPTIME *is running in exponential time* $2^{2^{p(n)}}$ *while* 3EXPTIME *is* $2^{2^{2^{p(n)}}}$.

Furthermore, a problem P is **hard** for a given complexity class C if any instance of a problem from C can be reduced to an instance of P through a reduction (in most cases, this reduction has to be in polynomial time, but for lower classes (PTIME and below), logarithmic space reductions must be used). A problem P is **complete** for a given complexity class C if it belongs to C and is hard for C. For the query entailment problem, two different measures of complexity are considered:

- **Combined complexity:** the input contains the set of rules, the set of facts and the query.
- **Data complexity:** the input contains only the set of facts while the set of rules and the query are assumed to be fixed.

Data complexity is sometimes considered more relevant [Lembo et al., 2010] because the query and the rules are usually far smaller than the set of facts in practical applications, however both complexities can help understand where the difficulties lie. Indeed, for instance, query answering over skolem-FES rules using a frontier chase has in the worst case 2ExpTime-complete combined complexity and PTime-complete data complexity. The following Table 2.1 describes the combined and data complexity of query answering for each studied concrete class of Skolem-FES.

Rule Class	Combined Complexity	Data Complexity
Datalog	ExpTime-complete [Chandra et al., 1981]	PTime-complete [Dantsin et al., 2001]
aGRD	ExpTime-complete [Calì et al., 2010b]	PTime-complete [Calì et al., 2010b]
Jointly-acyclic	2ExpTime-complete [Krötzsch and Rudolph, 2011]	PTime-complete [Krötzsch and Rudolph, 2011]
Weakly-acyclic	2ExpTime-complete [Fagin et al., 2005]	PTime-complete [Fagin et al., 2005]
Super-weakly-acyclic	2ExpTime-complete [Marnette, 2009]	PTime-complete [Marnette, 2009]
MFA	2ExpTime-complete [Zhang et al., 2015]	PTime-complete [Marnette, 2009]

Table 2.1: Complexity of CQ entailment for studied Skolem-FES concrete classes

2.1.5 Incoherence and inconsistence

To represent knowledge about the world one should account for "*negative knowledge*", i.e. information that dictates how things ought not to be, especially since generating new knowledge from seemingly correct information might lead to a contradiction down the line. A basic form of "negative knowledge" is stating that a fact and its negation (or equivalently, complement) should not be both asserted at the same time. While the existential rules language $\mathcal{L}$ is negation-free, the notion of integrity constraint from the database domain can be used to express negative knowledge.

Definition 2.23 (Negative constraint). *A negative constraint (or simply a constraint) is a rule of the form $\forall \vec{X}\, \mathcal{B}(\vec{X}) \rightarrow \bot$ where $\vec{X}$ is a tuples of variables, and $\mathcal{B}$ is a finite non-empty conjunction of atoms.*

In this thesis, we only consider "binary" negative constraints (a.k.a. denial constraints) that express a conflict between two atoms. This restriction simplifies subsequent definitions and does not imply a loss of generality since any negative constraint can be transformed into a set of rules and binary negative constraints [Calì et al., 2012].

Definition 2.24 (Conflicting facts). *A set of facts $Z = \{f_1, f_2, \ldots, f_n\}$ is a conflicting set of facts if and only if the body of a negative constraint can*

be mapped to Z. By abuse of notation, if $Z = \{f_1, f_2\}$, we say that f_1 is in conflict with f_2.

Example 2.10 (Negative constraint and conflicting facts). *Consider the negative constraint stating that it is impossible that a person is married and unmarried at the same time: $\forall X\, \mathit{married}(X) \wedge \mathit{unmarried}(X) \rightarrow \bot$. The fact $\mathit{married}(\mathit{bruno})$ is in conflict with $\mathit{unmarried}(\mathit{bruno})$ (and vice-versa) because the body of the negative constraint can be mapped to these facts.*

Negative constraints are used to ensure that a set of facts is consistent (i.e. it contains no contradictions). This is especially important since in presence of conflicts, query answering becomes trivial due to the principle of explosion (ex falso quodlibet), i.e. "*from falsehood anything follows*".

In the various domains of knowledge representation, conflicts might be inherent to the represented domain or may arise from an incorrect description of the world. When a set of factual knowledge contains no conflicts it is said to be *consistent*, otherwise it is *inconsistent*.

Definition 2.25 (Inconsistence). *A set of facts $\mathcal{F}$ is inconsistent with respect to a set of negative constraints $\mathcal{N}$ if and only if $(\mathcal{F} \cup \mathcal{N})$ has no possible model ($\mathit{models}(\mathcal{F}, \mathcal{N}) = \emptyset$) i.e. $(\mathcal{F} \cup \mathcal{N}) \vDash \bot$. In practice, $\mathcal{F}$ is inconsistent if a negative constraint can be applied i.e. there exists a negative constraint $r \in \mathcal{N}$ such that $\mathcal{F} \vDash \mathit{Body}(r)$.*

An inconsistent set of facts does not necessarily mean an incorrect representation of the factual knowledge of the world. In some cases, the inconsistency of the generated set of facts is unavoidable (i.e. the representation has no model) even with a correct description of factual knowledge. This is due to an *incoherent* set of rules.

Definition 2.26 (Incoherence). *A set of rules $\mathcal{R}$ is incoherent with respect to a set of negative constraints $\mathcal{N}$ if and only if $\mathcal{R} \cup \mathcal{N}$ is unsatisfiable i.e. for any set of facts S such that all rules in $\mathcal{R}$ are applicable, $\mathit{models}(S, \mathcal{R} \cup \mathcal{N}) = \emptyset$. The application of $\mathcal{R}$ on any set of facts S will inevitably lead to an inconsistent saturated set of facts S^* with respect to $\mathcal{N}$.*

Clearly, the notions of incoherence and inconsistence are highly related. In fact, an incoherent set of rules $\mathcal{R}$ will always lead to an inconsistent set of facts $\mathcal{F}^*$ if all rules in $\mathcal{R}$ are applied on $\mathcal{F}$ [Flouris et al., 2006]. The following Examples 2.11 and 2.12 describe the key difference between incoherence and inconsistence.

Example 2.11 (Incoherence). *Consider the following sets of facts $\mathcal{F}$, rules $\mathcal{R}$, and negative constraints $\mathcal{N}$ representing the knowledge that mammals do not lay eggs, platypus are mammals and penguins lay eggs. "Perry" is a platypus, does it lay eggs (i.e. $Q_1() = \mathit{layEggs}(\mathit{perry})$)? Does it not lay eggs (i.e. $Q_2() = \mathit{notLayEggs}(\mathit{perry})$)?*

- $\mathcal{F} = \{platypus(perry)\}$
- $\mathcal{R} = \{\ r_1 : \forall X\, platypus(X) \rightarrow mammal(X),$
  $r_2 : \forall X\, mammal(X) \rightarrow notLayEggs(X),$
  $r_3 : \forall X\, platypus(X) \rightarrow layEggs(X)\}$
- $\mathcal{N} = \{\forall X\, layEggs(X) \wedge notLayEggs(X) \rightarrow \bot\}$

The saturated set of facts resulting from a frontier chase is

- $\mathcal{F}^* = \{platypus(perry), mammal(perry), layEggs(perry), notLayEggs(perry)\}$.

The set of rules $\mathcal{R}$ is ***incoherent*** *because no set of facts (even outside $\mathcal{F}$) that makes all rules in $\mathcal{R}$ applicable prevents the application of the negative constraint, therefore $models(\mathcal{F}, \mathcal{R} \cup \mathcal{N}) = \emptyset$. The answer to the boolean queries Q_1 and Q_2 is "true" (principle of explosion) i.e. Perry lays eggs and does not lay eggs at the same time. The saturated set of facts $\mathcal{F}^*$ is* ***inconsistent*** *because $models(\mathcal{F}, \mathcal{R} \cup \mathcal{N}) = \emptyset$.*

Example 2.12 (Inconsistence and incoherence). *Consider the following sets of facts $\mathcal{F}$, rules $\mathcal{R}$, and negative constraints $\mathcal{N}$ defined by Hecham [2018] about a criminal case. If there is a scientific evidence incriminating a defendant, then he is responsible for the crime, if there is a scientific evidence absolving a defendant then he is not responsible for the crime. A defendant is guilty if responsibility is proven. If a defendant has an alibi then he is innocent. There is a scientific evidence "e1" incriminating a defendant "alice", while another scientific evidence "e2" is absolving her of the crime. She also has an alibi. Is Alice innocent (i.e. $Q_1() = innocent(alice)$)? Is she guilty (i.e. $Q_2() = guilty(alice)$)?*

- $\mathcal{F} = \{incrim(e1, alice),\ absolv(e2, alice),\ alibi(alice)\}$
- $\mathcal{R} = \{r_1 : \forall X, Y\, incrim(X, Y) \rightarrow resp(Y),$
  $r_2 : \forall X, Y\, absolv(X, Y) \rightarrow notResp(Y),$
  $r_3 : \forall X\, resp(X) \rightarrow guilty(X),$
  $r_4 : \forall X\, alibi(X) \rightarrow innocent(X)\}$
- $\mathcal{N} = \{\forall X\, resp(X) \wedge notResp(X) \rightarrow \bot,$
  $\forall X\, guilty(X) \wedge innocent(X) \rightarrow \bot\}$

The saturated set of facts resulting from a frontier chase is

- $\mathcal{F}^* = \{incrim(e1, alice),\ absolv(e2, alice),\ alibi(alice),\ resp(alice),$
  $notResp(alice),\ guilty(alice),\ innocent(alice)\}$.

The set of rules $\mathcal{R}$ is ***coherent*** *because $\mathcal{R} \cup \mathcal{N}$ is satisfiable i.e. there exists a possible set of facts $S = \{incrim(e1, \mathbf{bob}),\ absolv(e2, alice),\ alibi(alice)\}$ such that all rules in $\mathcal{R}$ are applicable and $models(S, \mathcal{R} \cup \mathcal{N}) \neq \emptyset$, the set*

$S^* = \{incrim(e1, \mathbf{bob}), absolv(e2, alice), aibli(alice), resp(\mathbf{bob}), notResp(alice), guilty(\mathbf{bob}), innocent(alice)\}$ *is* ***consistent*** *as no negative constraint is applicable on it.*

However the saturated set of facts $\mathcal{F}^*$ *is* ***inconsistent*** *because a negative constraint is applicable, thus* $models(\mathcal{F}, \mathcal{R} \cup \mathcal{N}) = \emptyset$*. Since the set of rules is coherent, the inconsistence of* $\mathcal{F}^*$ *is due to an erroneous set of intial facts (either one of the evidences, the alibi, or all of them are not valid).*

The classical answer to the boolean queries Q_1 *and* Q_2 *is "true" (i.e. Alice is guilty and innocent), because from falsehood, anything follows.*

Inconsistence and incoherence are problematic for classical query answering. Indeed, as in classical logic, contradictions trivialise query answering since everything follows from a contradiction.

In this thesis, we focus on inconsistency and only consider set of rules that are coherent with respect to the set of negative constraints. This is not a big assumption as the existential rules framework is widely used in Semantic Web and in the so called *Ontology-Based Data Access* (OBDA). In this setting, the rules and negative constraints are given by experts and are used as an ontology to "access" multiple data sources. These sources are prone to inconsistencies whereas the ontology is reliable since it is constructed by field experts. Thus, we use the term *inconsistent knowledge base* to refer to a triple composed of a fact base, a set of negative constraints and a set of coherent rules with respect to the negative constraints.

Definition 2.27 (Inconsistent knowledge base). *An inconsistent knowledge base (or simply knowledge base if it is obvious by the context) is a tuple* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *where* $\mathcal{F}$ *is a set of facts,* $\mathcal{N}$ *is a set of negative constraints and* $\mathcal{R}$ *is a set of existential rules coherent with respect to* $\mathcal{N}$*.*

The set of all possible knowledge bases is denoted by $\mathcal{KB}_s$. We now introduce the notion of $\mathcal{R}$-inconsistency. As opposed to the notion of *conflicting facts* where the body of a negative constraint can be directly mapped to the set of facts, $\mathcal{R}$-inconsistency also consider the facts that can be derived before trying to map the body of negative constraints.

Definition 2.28 ($\mathcal{R}$-inconsistence). *Let us consider a knowledge base* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$*. We say that a set of facts* X *is* $\mathcal{R}$*-inconsistent with respect to* $\mathcal{KB}$ *if and only if* $Sat_{\mathcal{R} \cup \mathcal{N}}(X) \models \bot$*. Otherwise,* X *is said to be* $\mathcal{R}$*-consistent.*

In the knowledge representation and reasoning field, there are several ways to handle inconsistencies. The two main approaches are Consistency-based approaches (first mentioned by Rescher and Manor [1970]) and Dung-style logic-based argumentation approaches. The former consists in computing maximal consistent subsets (MCS) or repairs of the knowledge base and using non-classical consequence relation to infer from the knowledge base. In our context, the highlight is put on subsets of the fact base and repairs

are particular subsets of facts. A MCS or (data) repair [Arenas et al., 1999] of an inconsistent knowledge base $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is a maximal for set inclusion subset of $\mathcal{F}$ that is $\mathcal{R}$-consistent. Thus, any superset of a repair is $\mathcal{R}$-inconsistent.

Definition 2.29 (Repair). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, a set $X \subseteq \mathcal{F}$ is a repair of $\mathcal{KB}$ if and only if X is $\mathcal{R}$-consistent and for every $X \subset X'$, X' is $\mathcal{R}$-inconsistent. The set of all repairs of $\mathcal{KB}$ is denoted by $repairs(\mathcal{KB})$.*

The notion of minimal inconsistent set is similar to the concept of repairs and corresponds to minimal for set inclusion subsets of $\mathcal{F}$ that are $\mathcal{R}$-inconsistent.

Definition 2.30 (Minimal inconsistent set). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, a set $X \subseteq \mathcal{F}$ is a minimal inconsistent set of $\mathcal{KB}$ if and only if X is $\mathcal{R}$-inconsistent and for every $X' \subset X$, X' is $\mathcal{R}$-consistent. The set of all minimal inconsistent sets of $\mathcal{KB}$ is denoted by $MI(\mathcal{KB})$.*

The free facts of a knowledge base $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ are facts of $\mathcal{F}$ that are not "touched" by the inconsistencies. As a result, they are in every repairs or in no minimal inconsistent sets.

Definition 2.31 (Free fact). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, a fact $f \in \mathcal{F}$ is a free fact if and only if for every minimal inconsistent set $m \in MI(\mathcal{KB}), f \notin m$. We denote by $Free(\mathcal{KB})$, the set of free facts of $\mathcal{KB}$.*

Dung-style logic-based argumentation approaches are also widely used to handle inconsistencies. Those approaches are based on instantiating an abstract argumentation framework composed of a set of arguments and a set of attacks among them. The logic-based version of an argumentation framework views an argument as a structured entity built from a knowledge base $\mathcal{KB}$ and the approach proceeds by computing all of the possible arguments and attacks. Then, coalitions of arguments [Bertossi et al., 2005] called extensions (sets of arguments that are conflict-free and are defending themselves) are computed and inconsistency-tolerance can be defined on those extensions. In the next section, we begin by providing an introduction to the argumentation theory by introducing Dung's abstract model of argumentation [Dung, 1995] and the ranking-based semantics approach.

2.2 Argumentation theory

Argumentation represents a major component of human intelligence. The problems of understanding argumentation and its role in the way humans reason have been addressed by many researchers in different fields. In this section, we give an introduction to the argumentation theory as it was defined originally by Dung [1995]. His perception of argumentation was built around the basic principle that "the one who has the last word laughs best". Roughly speaking, Dung gives the idea that a statement is believable if it can be argued successfully against attacking arguments. In other words, whether or not an agent believes in a statement depends on whether or not the argument supporting this statement can be successfully defended against the counterarguments. Although many researchers have been analysing the structure of arguments before Dung [Birnbaum et al., 1980; Birnbaum, 1982; Cohen, 1987], he was the first one (to our knowledge) to clearly propose a simple model for understanding the acceptability of arguments by proposing semantical relations for abstract argumentation frameworks.

2.2.1 Abstract argumentation semantics

An abstract argumentation framework as defined by Dung [1995] takes as input a set of arguments and a pre-constructed binary relation that represents attacks between arguments. In Dung's abstract model, the structure of arguments and the type of attack are not defined and are left unspecified.

Definition 2.32 (Argumentation framework [Dung, 1995]). *An argumentation framework is a pair $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ where $\mathscr{A}$ is a set of arguments and $\mathscr{R}$ is a binary relation over $\mathscr{A}$. Given two arguments $a, b \in \mathscr{A}$, we say that a attacks b if and only if $(a, b) \in \mathscr{R}$.*

Notation 2.5. *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ and $\mathfrak{F}' = (\mathscr{A}', \mathscr{R}')$ be two argumentation frameworks. $\mathfrak{F} \oplus \mathfrak{F}'$ denotes the argumentation frameworks $(\mathscr{A} \cup \mathscr{A}', \mathscr{R} \cup \mathscr{R}')$ representing the merging of the two argumentation frameworks $\mathfrak{F}$ and $\mathfrak{F}'$.*

An argumentation framework can be seen as a directed graph where vertices represent arguments and edges represent attack between argument.

Example 2.13 (Argumentation framework). *Suppose we have three arguments a, b, and c such that a and b attack each other (i.e. $(a, b), (b, a) \in \mathscr{R}$), and c attacks b (i.e. $(c, b) \in \mathscr{R}$). This argumentation framework is shown in Figure 2.3.*

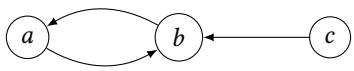

Figure 2.3: Argumentation framework of Example 2.13

Definition 2.33 (Path). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework and $a \in \mathscr{A}$. We say that a sequence $S = (a_1, \ldots, a_n)$ is a path of size n from a_n to a if and only if $a_1 = a$ and for every $i \in \{2, 3, \ldots, n\}, (a_i, a_{i-1}) \in \mathscr{R}$.*

Notation 2.6. *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework and $a, b \in \mathscr{A}$. We say that $b \in \mathscr{R}_n^-(a)$ if and only if there exists a path of size n from b to a. The set $\mathscr{R}_1^-(a)$ is called the set of direct attackers of a whereas $\mathscr{R}_2^-(a)$ are called direct defenders of a. Please note that we will often use the alternative notation $Att_{\mathfrak{F}}^-(a)$ to denote the set of* direct attackers *of a in $\mathfrak{F}$ or simply $Att^-(a)$ if the argumentation framework is clear from the context. Similarly, we also use the notation $Att^+(a)$ to denote the set of arguments directly attacked by a, namely $Att^+(a) = \{b \mid (b, a) \in \mathscr{R}\}$.*

Definition 2.34 (Set attack and defense). *A set of argument S attacks an argument b if there exists an argument $c \in S$ such that $(c, b) \in \mathscr{R}$. If there is an argument $a \in S$ such that $(b, a) \in \mathscr{R}$ and S attacks b then S defends a.*

Argumentation is based on the notion of *acceptability of an argument* in the sense that a rational agent accepts only arguments which she can defend from all possible attacks.

Definition 2.35 (Acceptability of an argument). *Given an argumentation framework $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$. An arguments $a \in \mathscr{A}$ is acceptable with respect to a set of argument $S \subseteq \mathscr{A}$ if and only if S defends a from all its attacks, that is for every $b \in \mathscr{A}$ such that $(b, a) \in \mathscr{R}$, there exists $c \in S$ such that $(c, b) \in \mathscr{R}$.*

Example 2.14 (Example 2.13 cont'd). *a is acceptable with respect to the set $\{c\}$.*

2.2.2 Extension-based approaches

Acceptability of argument is used to define argumentation semantics. Two different methods are proposed to define semantics: *extension-based* [Dung, 1995] and *labeling-based* [Caminada, 2006]. We start by the extension-based approach which defines what an acceptable argument means under some specific semantics. The idea behind the extension-based approach is to identify and select a set of arguments called *extensions* that can survive a conflict together. Thus, an extension is often represented as a reasonable position or viewpoint in a debate. The reader can find examples of the semantics presented in this thesis (the admissible, complete, grounded, preferred and stable) in Dung [1995] and an intuitive introduction to argumentation semantics can be found in Baroni et al. [2011].

Extension-based semantics are defined on the principle of conflict-freeness which translates the idea the arguments in an extension should be able to "stand together", that is, the arguments of the same extension should not attack each other.

Definition 2.36 (Conflict-freeness). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework. A set of arguments $S \subseteq \mathscr{A}$ is conflict-free if and only if there are no $a, b \in S$ such that $(a, b) \in \mathscr{R}$.*

Please note that Definition 2.36 excludes all of the sets containing self-attacking arguments.

Example 2.15 (Example 2.13 cont'd). *$\{a, c\}$ is conflict-free.*

A maximal conflict-free set of arguments is called a *naive extension.*

Definition 2.37 (Naive semantics). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework. A set of arguments $S \subseteq \mathscr{A}$ is a* naive extension *if and only if S is conflict-free and for every $S' \subseteq \mathscr{A}$ such that $S \subset S'$, S' is not conflict-free.*

A set of non-conflicting arguments can be seen as an agent's position in a debate, for this position to hold it has to defend all its argument. This corresponds to the notion of admissibility [Dung, 1995].

Definition 2.38 (Admissibility of a set). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework. A conflict-free set of arguments $S \subseteq \mathscr{A}$ is admissible if and only if every argument $a \in S$ is acceptable with respect to S.*

An admissible set of arguments is a set of non-conflicting arguments that defends all its elements, such set is called an *admissible extension.* Every argumentation framework has at least one admissible set: the empty set.

Example 2.16 (Example 2.13 cont'd). *The admissible extensions are: $\emptyset$, $\{a\}$, $\{c\}$, and $\{a, c\}$. Note that $\{b\}$ is not an admissible set since it does not defend itself from c.*

A preferred set of arguments is a maximal set of arguments that is admissible. The idea of the preferred semantics is that one wants to accept as many arguments as reasonably possible to have the largest viewpoint on a debate.

Definition 2.39 (Preferred semantics). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework. A preferred extension is a maximal (for set inclusion) admissible set of arguments $S \subseteq \mathscr{A}$.*

A stable set of arguments S is a conflict-free set that attacks all of the arguments outside of S. The idea of the stable semantics is that an argument can only be for or against a viewpoint in a debate and that neutrality is not allowed.

Definition 2.40 (Stable semantics). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework. A stable extension is a conflict-free set of arguments $S \subseteq \mathscr{A}$ such that for every $b \in (\mathscr{A} \setminus S)$ there exists an argument $a \in S$ such that $(a, b) \in \mathscr{R}$.*

A complete set of arguments is an admissible set that contains all of the arguments that it defends. The complete semantics refines the admissibility in the sense that one should always accept an argument if it can be defended.

Definition 2.41 (Complete semantics). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework. An admissible set of arguments $S \subseteq \mathscr{A}$ is a complete extension if and only if for every $a \in \mathscr{A}$, if S defends a then $a \in S$.*

Grounded semantics is the most skeptical (or least committed) of argumentation semantics, it is defined based on the notion of complete extension. It is the admissible extension that includes all the arguments it can defend from all attacks.

Definition 2.42 (Grounded semantics). *The grounded extension of an argumentation framework is the least (with respect to set-inclusion) complete extension.*

In some cases, the stable semantics yields no extensions at all (not even the empty set). That is why a more refined approach was defined: the semi-stable semantics [Caminada et al., 2012]. Please note that this semantics is equivalent to the admissible stage semantics defined by Verheij [1999].

Definition 2.43 (Semi-stable semantics [Caminada et al., 2012]). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework. A semi-stable extension is a complete extension S such that $S \cup \{b \in \mathscr{A} \mid$ there exists $a \in S$ such that $(a, b) \in \mathscr{R}\}$ is maximal (with respect to set inclusion) amongst all complete extensions.*

Contrary to the stable extensions, the existence of the semi-stable extensions is always guaranteed. Furthermore, a stable semantics is a semi-stable extension and semi-stable extensions coincide with stable extensions when the set of stable extensions is not empty.

An ideal extension is a maximal for set inclusion set of argument that is a subset of each preferred extension. It was shown by Caminada and Pigozzi [2011] that the ideal extension is also a complete extension and thus it is a superset of the grounded extension.

Definition 2.44 (Ideal semantics [Caminada and Pigozzi, 2011]). *Given an argumentation framework $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$. An admissible set S is called ideal if and only if it is a subset of each preferred extension. The ideal extension of $\mathfrak{F}$ is a maximal (with respect to set inclusion) ideal set.*

For our purposes, we require some further formal notions. An argumentation framework is strongly connected if and only if there is a path from any argument a to any argument a'.

Definition 2.45 (Strongly connected). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework. We say that $\mathfrak{F}$ is strongly connected if and only if for every $a, a' \in \mathscr{A}$ such that $a \neq a'$, there is a path from argument a to argument a'.*

The nodes of an arbitrary directed graph can be partitioned such that the subgraphs, induced by each set of nodes, are maximal strongly connected subgraphs. Each set of such a partition is called a strongly connected components of this graph. In the rest of this thesis, we will denote by $SCC(\mathfrak{F})$, this particular partition of the set of arguments of $\mathfrak{F}$.

Definition 2.46 (Component-defeated [Gaggl and Woltran, 2013]). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework and $S \subseteq \mathscr{A}$ a set of arguments. An argument $b \in \mathscr{A}$ is component-defeated by S if there exists $a \in S$ such that $(a, b) \in \mathscr{R}$ and a is not in same the strongly connected component than b. The set of arguments component-defeated by S in $\mathfrak{F}$ is $D_{\mathfrak{F}}(S)$.*

All of the above mentioned argumentation semantics are admissibility-based, i.e. the extension returned are admissible sets. Moreover, in the multiple-status semantics (such as complete, preferred, stable and semi-stable), we can notice that odd-length unidirectional attack cycles are handled badly. However, in some applications, cycles need to be treated equally independently of their length [Pollock, 2001]. The stage semantics conforms with this idea of "equal cycles treatment" but loses its proximity with the grounded semantics as it was shown that even non attacked arguments can be rejected in some cases [Baroni et al., 2011]. Against this background, the cf2 semantics was designed as a multiple-status semantics that is not admissibility-based, treats cycles equally and which accepted arguments are a superset of those accepted by the grounded semantics.

Definition 2.47 (Cf2 semantics [Gaggl and Woltran, 2013]). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework and $S \subseteq \mathscr{A}$ be a set of arguments. S is a cf2 extension of $\mathfrak{F}$ if and only if:*

- *in case $|SCC(\mathfrak{F})| = 1$, then S is a maximal conflict free set of $\mathfrak{F}$,*
- *otherwise, for every $C \in SCC(\mathfrak{F}), (S \cap C)$ is a cf2 extension of $(A \cap Y, R \cap (Y \times Y))$ where $Y = C \setminus D_{\mathfrak{F}}(S)$.*

Notation 2.7. *Let $\mathfrak{F}$ be an argumentation framework, we will denote by $Ext_x(\mathfrak{F})$ the set of extensions with respect to the argumentation semantics x for $\mathfrak{F}$. We use the abbreviations cf, a, p, s, c, g, ss, i and $cf2$ for respectively conflict-free, admissible, preferred, stable, complete, grounded, semi-stable, ideal and cf2.*

Definition 2.48 (Sceptically accepted, credulously accepted and rejected arguments). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework and $Ext_x(\mathfrak{F})$ be the set of extensions with respect to the argumentation semantics x for $\mathfrak{F}$. We say that:*

- *a is sceptically accepted with respect to x if and only if for every $\varepsilon \in Ext_x(\mathfrak{F}), a \in \varepsilon$.*

- *a is credulously accepted with respect to x if and only if for there exists $\varepsilon_1, \varepsilon_2 \in Ext_x(\mathfrak{F})$, such that $a \in \varepsilon_1$ and $a \notin \varepsilon_2$.*
- *a is rejected with respect to x if and only if for every $\varepsilon \in Ext_x(\mathfrak{F}), a \notin \varepsilon$.*

Example 2.17 (Argumentation semantics). *Consider the argumentation framework $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ such that $\mathscr{A} = \{a, b, c, d\}$ and $\mathscr{R} = \{(e, e), (d, e), (d, c), (c, d), (b, c), (a, b)\}$, represented in Figure 2.4. We make the following observations:*

- *The admissible extensions are $\{d\}$, $\{a\}$, $\{a, d\}$, $\{a, c\}$ and $\emptyset$.*
- *The complete extensions are $\{a\}$, $\{a, c\}$, $\{a, d\}$ and $\emptyset$.*
- *The preferred and cf2 extensions are $\{a, c\}$ and $\{a, d\}$.*
- *The stable and semi-stable extension is $\{a, d\}$.*
- *The ideal extension is $\{a\}$.*
- *The least complete extension is $\{a\}$ which is the ground extension.*

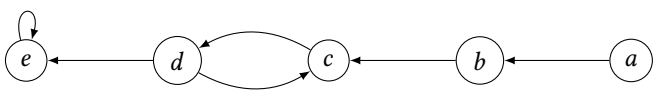

Figure 2.4: Argumentation framework of Example 2.17

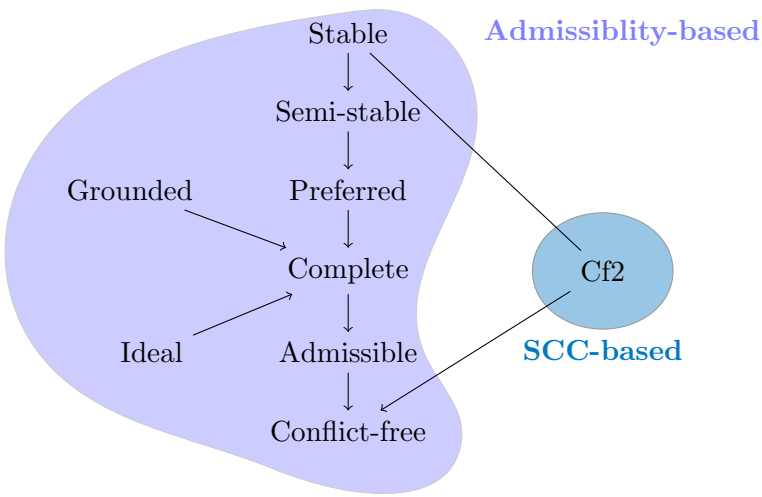

Figure 2.5: Inclusion relations between the several argumentation semantics used in this thesis.

Semantics	CF	DF	ADM	INCDF	MAX	AGR	UNIQ	EXIST
Admissible	✓	✓	✓					✓
Complete	✓	✓	✓	✓				✓
Stable	✓	✓	✓	✓		✓		
Semi-stable	✓	✓	✓	✓	✓			✓
Preferred	✓	✓	✓	✓	✓			✓
Grounded	✓	✓	✓	✓			✓	✓
Cf2	✓				✓			✓
Ideal	✓	✓	✓	✓	✓		✓	✓

Table 2.2: Argumentation semantics with respect to criteria. ✓ means the criterion is satisfied

In Figure 2.5, we show the inclusion relations between the several argumentation semantics used in this thesis. An arrow from the node A to the node B means that an extension for semantics A is also an extension for semantics B. In Table 2.2, we summarise the semantics and their essential criteria. The criteria are as follows: CF means that the extensions are conflict-free, DF means that they defend all their elements, INCDF means that they include what they defend, MAX means that they are maximal with respect to inclusion, AGR means that they attack all arguments that are outside of the extension, UNIQ means that there is always one extension and EXIST means that there is always at least one extension. The table is only an illustration and the criteria are not completely dependent as some of them are derivable from others.

2.2.3 Labelling approach

The labelling approach consists in mapping arguments with labels.

Definition 2.49 (Labelling). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework and Λ be a set of labels. A labelling L with respect to Λ is a total function $L : \mathscr{A} \to \Lambda$.*

A sensible choice for the labels (but not the only one possible) is **in**, **out**, and **undec** in order to represent that an argument is accepted, rejected and undecided respectively. However, such a mapping does not have much sense if made arbitrary. Thus, the notion of reinstatement labelling was introduced as constraints to ensure the meaning of the mapping.

Definition 2.50 (Reinstatement labelling). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework. A labelling L is a reinstatement labelling if and only if all of the following items are satisfied:*

- *for every $a \in \mathscr{A}$, $L(a) = in$ if and only if for every $b \in \mathscr{A}$ such that $(b, a) \in \mathscr{R}$, $L(b) = out$*
- *for every $a \in \mathscr{A}$, $L(a) = out$ if and only if there exists $b \in \mathscr{A}$ such that $(b, a) \in \mathscr{R}$ and $L(b) = in$*
- *for every $a \in \mathscr{A}$, $L(a) = undec$ if and only if $L(a) \neq in$ and $L(a) \neq out$*

Example 2.18 (Grounded labelling). *The argumentation framework in Example 2.4 can have different reinstatement labelings. Figure 2.6 corresponds to the complete extension $\{a, c\}$ and Figure 2.7 corresponds to the grounded extension $\{a\}$.*

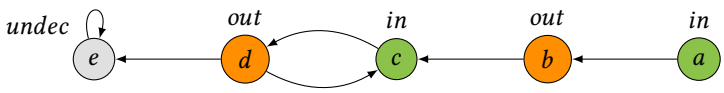

Figure 2.6: Labelling that corresponds to $\{a, c\}$ of Example 2.17

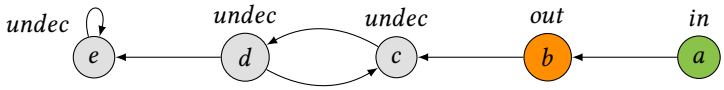

Figure 2.7: Labelling that corresponds to $\{a\}$ of Example 2.17

Every extension can be translated into a reinstatement labelling: the arguments of the extension are in, those attacked by an argument of the extension are out, and the others are undec. Similarly, one can build an extension from a reinstatement labelling just by considering the arguments that are labeled "in". Moreover, Caminada [2006] proved that the reinstatement labelings and the complete extensions can be mapped in a bijective way and that other Dung's argumentation semantics can be obtained from particular reinstatement labelling. In this thesis, we will not delve into the labelling representation and only use the extension-based approach.

In the next section, we introduce a similar approach for analysing and detecting the most attacked arguments. *Ranking-based semantics* [Amgoud and Ben-Naim, 2013; Bonzon et al., 2016; Besnard and Hunter, 2001] were developed for ranking arguments with respect to their acceptability. The added value of this approach is that contrary to the usual three value statuses (accepted, rejected or undecided) offered by the argumentation semantics, the ranking-based semantics offer a more gradual acceptability range which can be more useful for many applications such as debate platforms on the web (see the work of Leite and Martins [2011]).

2.2.4 Ranking-based semantics

Amgoud and Ben-Naim [2013] give the three main properties of extension-based (and labelling-based) semantics: *Killing*, *Existence* and *Flatness*.

- *Killing.* An attack from an argument a to b is drastic and it is no longer possible for b to be in the same extension as a.
- *Existence.* One successful attack against an argument has the same effect on an argument as any number of successful attacks.
- *Flatness.* All arguments with the same status (accepted, rejected, undecided) have the same level of acceptability and cannot be distinguished.

Although those intuitions may have been understandable in the context of paraconsistent logics [Besnard and Hunter, 2008] because they will ensure the consistency of a set of formulas by killing any contradiction between arguments, these considerations are arguable when given in the context of decision-making application [Yun et al., 2016, 2018a] or online debate platforms [Leite and Martins, 2011]. In those applications, it is understandable that many successful attacks should have a more negative impact than just one successful attack. Thus, although many arguments can have the same status, they should not be undistinguishable.

In the rest of this section, we formally define the ranking-based semantics and scoring semantics.

Definition 2.51 (Ranking-based semantics σ). *A ranking-based semantics σ associates to any argumentation framework $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ a ranking $\geq^{\sigma}_{\mathfrak{F}}$ on $\mathscr{A}$ where $\geq^{\sigma}_{\mathfrak{F}}$ is a total preorder (reflexive and transitive relation) on $\mathscr{A}$. The notation $a \geq^{\sigma}_{\mathfrak{F}} b$ means that a is at least as acceptable as b.*

Notation 2.8. *We use the notation $a \approx^{\sigma}_{\mathfrak{F}} b$ if and only if $a \geq^{\sigma}_{\mathfrak{F}} b$ and $b \geq^{\sigma}_{\mathfrak{F}} a$. Moreover, we say that $a \succ^{\sigma}_{\mathfrak{F}} b$ if and only if $a \geq^{\sigma}_{\mathfrak{F}} b$ and $b \not\geq^{\sigma}_{\mathfrak{F}} a$. Likewise, we say that $a \preceq^{\sigma}_{\mathfrak{F}} b$ if and only if $a \not\succ^{\sigma}_{\mathfrak{F}} b$. Finally, we say that $a \prec^{\sigma}_{\mathfrak{F}} b$ if and only if $a \preceq^{\sigma}_{\mathfrak{F}} b$ and $b \not\preceq^{\sigma}_{\mathfrak{F}} a$*

A scoring function assign to each argument, in an argumentation framework, a score based on different criteria. The score can be chosen in an interval $[0, 1], [-1, 1], \mathbb{N}$ or even $\mathbb{R}$. Please note that the score given by a scoring function should not be confused with the inner weight of arguments in weighted argumentation frameworks [Dunne et al., 2011; Coste-Marquis et al., 2012]. Indeed, weights are given by external sources (preferences, inner strength) whereas scores are computed with respect to the intrinsic structure of the argumentation framework.

Definition 2.52 (Scoring function). *A scoring semantics is a function which associates to any argumentation framework $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ a scoring S on $\mathscr{A}$, where S is a function from $\mathscr{A}$ to $\mathbb{R}$.*

Please note that scoring function and ranking-based semantics are not independent notions. Indeed, one can use the scores from a scoring function to rank arguments, thus obtaining a ranking-based semantics. However, it does not mean that every ranking-based semantics is based on an underlying scoring function. Indeed, some ranking-based semantics can directly compare arguments using methods such as the lexicographical order without using scores.

Definition 2.53 (Lexicographical order). *Let $V = (V_1, V_2, \ldots, V_n)$ and $V' = (V'_1, V'_2, \ldots, V'_n)$ be two vectors of real numbers of size n. The lexicographical order between V and V' is defined as $V \succ_{lex} V'$ if and only if there exists $i \in \{1, \ldots, n\}$ such $V_i < V'_i$ and for every $j < i, V_j = V_i$.*

Notation 2.9. *We use the notation $V \approx_{lex} V'$ if and only if $V \not\succ_{lex} V'$ and $V' \not\succ_{lex} V$. Moreover, we say that $V \succeq_{lex} V'$ if and only if $V' \not\succ_{lex} V$.*

2.2.4.1 Existing ranking-based semantics

In this section, we introduce the ranking-based semantics from the literature that will be used in throughout the thesis, i.e. h-categoriser, burden-based and discussion-based. In order to correctly illustrate the ranking outputted by each of these ranking-based semantics, we use the argumentation framework in Figure 2.8 proposed by Delobelle [2017]

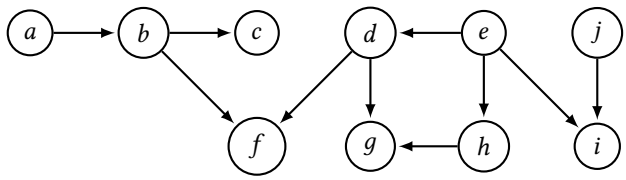

Figure 2.8: An argumentation framework $\mathfrak{F}$

Besnard and Hunter [2001] proposed the h-categoriser function as a scoring function that gives the strength of an argument based on the strengths of its attackers.

Definition 2.54 (H-categoriser function [Besnard and Hunter, 2001]). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an abstract argumentation framework. The h-categoriser function is $C' : \mathscr{A} \to [0, 1]$ defined as, for all $a \in \mathscr{A}$:*

$$C'(a) = \begin{cases} 1 & \text{if } \mathscr{R}_1^-(a) = \emptyset \\ \frac{1}{1 + \sum_{b \in \mathscr{R}_1^-(a)} C'(b)} & \text{otherwise} \end{cases}$$

Pu et al. [2014] showed the existence and uniqueness of the values returned by the h-categoriser function for any argumentation frameworks. The following definition shows how the h-categoriser ranking-based semantics is constructed from the scores returned by the h-categoriser function.

Definition 2.55 (H-categoriser ranking-based semantics [Pu et al., 2014]). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an abstract argumentation framework. The h-categoriser ranking-based semantics on $\mathfrak{F}$ returns a ranking $\succeq^{hcat}_{\mathfrak{F}}$ on $\mathscr{A}$ such that for every $a, b \in \mathscr{A}$:*

$$b \succeq^{hcat}_{\mathfrak{F}} a \text{ if and only if } C'(a) \leq C'(b)$$

Example 2.19 (H-categoriser). *We consider the argumentation framework $\mathfrak{F}$ depicted in Figure 2.8. The values returned by the h-categoriser function for each argument are $C'(a) = C'(e) = C'(j) = 1, C'(c) \approx 0.667, C'(i) \approx 0.333$ and $C'(b) = C'(d) = C'(f) = C'(g) = C'(h) = 0.5$. Thus, the ranking on $\mathscr{A}$ outputted by the h-categoriser ranking-based semantics is:*

$$a \approx^{hcat}_{\mathfrak{F}} e \approx^{hcat}_{\mathfrak{F}} j \succ^{hcat}_{\mathfrak{F}} c \succ^{hcat}_{\mathfrak{F}} b \approx^{hcat}_{\mathfrak{F}} d \approx^{hcat}_{\mathfrak{F}} f \approx^{hcat}_{\mathfrak{F}} g \approx^{hcat}_{\mathfrak{F}} h \succ^{hcat}_{\mathfrak{F}} i$$

Amgoud and Ben-Naim [2013] introduced the Discussion-based ranking-based semantics which compares the arguments with respect to the number of paths leading to them. The intuition behind this ranking-based semantics is that an even path to a should increase the score of a whereas an odd path should reduce its score.

Definition 2.56 (Discussion count). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework, $a \in \mathscr{A}$ and $i \in \mathbb{N} \setminus \{0\}$.*

$$Dis_i(a) = \begin{cases} -|\mathscr{R}^-_i(a)| & \text{if } i \text{ is odd} \\ |\mathscr{R}^-_i(a)| & \text{otherwise} \end{cases}$$

The discussion count of a is the vector $Dis(a) = (Dis_1(a), Dis_2(a), \dots)$.

The discussion-based ranking-based semantics is computed using the lexicographical order on the discussion counts of the arguments.

Definition 2.57 (Discussion-based ranking-based semantics [Amgoud and Ben-Naim, 2013]). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework. The discussion-based ranking-based semantics on $\mathfrak{F}$ returns a ranking $\succeq^{dis}_{\mathfrak{F}}$ on $\mathscr{A}$ such that for every $a, b \in \mathscr{A}$:*

$$b \succeq^{dis}_{\mathfrak{F}} a \text{ if and only if } Dis(a) \succeq_{lex} Dis(b)$$

.

Example 2.20 (Discussion-based ranking-based semantics). *We consider the argumentation framework $\mathfrak{F}$ depicted in Figure 2.8. We obtain the following discussion counts:*

- *Dis(a) = Dis(e) = Dis(j) = (0, 0, 0)*
- *Dis(c) = (1, -1, 0)*
- *Dis(b) = Dis(d) = Dis(h) = (1, 0, 0)*
- *Dis(f) = Dis(g) = (2, -2, 0)*
- *Dis(i) = (2, 0, 0)*

Here, it is enough to only compute discussion counts of size three as we can see that every discussion count finishes with a zero (it means that there are no paths of size more than two). Using the lexicographical order, we get the following ranking on $\mathscr{A}$:

$$a \approx^{dis}_{\mathfrak{F}} e \approx^{dis}_{\mathfrak{F}} j \succ^{dis}_{\mathfrak{F}} c \succ^{dis}_{\mathfrak{F}} b \approx^{dis}_{\mathfrak{F}} d \approx^{dis}_{\mathfrak{F}} h \succ^{dis}_{\mathfrak{F}} f \approx^{dis}_{\mathfrak{F}} g \succ^{dis}_{\mathfrak{F}} i$$

The last ranking-based semantics considered in this thesis is the Burden-based ranking-based semantics also defined by Amgoud and Ben-Naim [2013]. This ranking-based semantics uses some intuition from the discussion-based ranking-based semantics as it first considers the direct attackers since it is based on the lexicographical order. However, instead of computing the number of paths for a specific argument, it updates the "burden" of each argument with respect to the burden of its direct attackers.

Definition 2.58 (Burden vector). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework, $a \in \mathscr{A}$ and $i \in \mathbb{N} \setminus \{0\}$.*

$$Bur_i(a) = \begin{cases} 1 & \text{if } i = 0 \\ 1 + \sum_{b \in \mathscr{R}_1^-(a)} \frac{1}{Bur_{i-1}(b)} & \text{otherwise} \end{cases}$$

The burden vector of an argument a is $Bur(a) = (Bur_0(a), Bur_1(a), \dots)$.

The burden-based ranking-based semantics is then computed using the lexicographical order on the burden vectors of the arguments.

Definition 2.59 (Burden-based ranking-based semantics). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework. The burden-based ranking-based semantics on $\mathfrak{F}$ returns a ranking $\succeq^{bur}_{\mathfrak{F}}$ on $\mathscr{A}$ such that for every $a, b \in \mathscr{A}$:*

$$b \succeq^{bur}_{\mathfrak{F}} a \text{ if and only if } Bur(a) \succeq_{lex} Bur(b)$$

.

Example 2.21 (Burden-based ranking-based semantics). *We consider the argumentation framework $\mathfrak{F}$ depicted in Figure 2.8. We obtain the following burden vectors:*

- *Bur(a) = Bur(e) = Bur(j) = (1, 1, 1, 1)*
- *Bur(c) = (1, 2, 1.5, 1.5)*
- *Bur(b) = Bur(d) = Bur(h) = (1, 2, 2, 2)*
- *Bur(f) = Bur(g) = (1, 3, 2, 2)*
- *Bur(i) = (1, 3, 3, 3)*

Using the lexicographical order, we get the following ranking on $\mathscr{A}$:

$$a \approx^{bur}_{\mathfrak{F}} e \approx^{bur}_{\mathfrak{F}} j \succ^{bur}_{\mathfrak{F}} c \succ^{bur}_{\mathfrak{F}} b \approx^{bur}_{\mathfrak{F}} d \approx^{bur}_{\mathfrak{F}} h \succ^{bur}_{\mathfrak{F}} f \approx^{bur}_{\mathfrak{F}} g \succ^{bur}_{\mathfrak{F}} i$$

Although the ranking on arguments is the same for the burden-based and discussion-based ranking-based semantics in Example 2.20 and 2.21, the equality of the two ranking-based semantics is not true. The reader is invited to read the original paper by Amgoud and Ben-Naim [2013] for an intuitive counter-example.

2.3 Summary

In this chapter we presented the existential rule logical fragment along with the *frontier chase* forward chaining inference mechanism. We showed that allowing the presence of existential quantifiers in the head of rules might lead to infinite rule applications, that is why a derivation reducer is needed to *remove redundant rule applications*. We presented the *frontier derivation reducer* and showed the types of rules (*Skolem-FES*) for which it is guaranteed to stop. Then we defined the different types of conflicts, namely, *inconsistence* when a negative constraint is applicable, and *incoherence* when the set of rules is unsatisfiable. In the rest of this thesis, we only work with inconsistent knowledge bases with a coherent set of rules.

Afterwards, we presented the *Argumentation Theory* which is a conflict-tolerant form of reasoning that is based on argumentation frameworks with arguments and attacks among them. These argumentation frameworks can then be used to evaluate arguments using different approaches:

1. The extensions-based approaches are semantics that are able to select sets of non-conflicting arguments called extensions (or equivalently,

coalitions) that can survive a conflict together. We first recalled the some admissible-based argumentation semantics such as Dung's argumentation semantics (complete, preferred, stable and grounded), the semi-stable and the ideal semantics. Second, we recalled an non admissible-based semantics based on strongly-connected components: the cf2 semantics. Lastly, we presented how all of those argumentation semantics are connected with respect to extensions inclusion.

2. The labelling approach is a semantics that is based on giving labels to arguments. In the literature, we usually restrict ourselves to reinstatement labelings in order to ensure the meaning of the labelling. Since it was proven by Caminada [2006] that reinstatement labelings straightforwardly coincide with the complete extensions, we can get all of the usual Dung's argumentation extensions by picking specific labelings in the set of reinstatement labelings. In this thesis, we will not delve into this labelling approach.

3. The last approach is called ranking-based semantics. These semantics were developed following different intuitions than extension-based approaches. Indeed, ranking-based semantics are more gradual because they do not directly "kill" an argument when it is attacked. Ranking-based semantics return a ranking on arguments for any argumentation framework. In this thesis, we only consider three ranking-based semantics: h-categoriser, burden-based and discussion-based ranking-based semantics. The first ranking-based semantics is defined upon a scoring function whereas the last two are defined around the lexicographical order on vectors of values (burden vector and discussion count).

We discussed about the existential rule logical fragment and approaches for evaluating arguments when given an abstract argumentation framework. In the next chapter, we show how the two notions can be combined using logic-based argumentation and instantiation an argumentation framework with an inconsistent knowledge base in the existential rules language.

Chapter 2 in a Nutshell

- *Reasoning with existential rules requires a derivation reducer to become decidable. Frontier/Skolem chase is the most used forward chaining inference mechanism and has decidable classes (types) of rules called Skolem-FES.*
- *There are two types of conflicts: inconsistency when a negative constraint is applicable, and incoherence when the set of rules is unsatisfiable. We will only work with coherent set of rules.*
- *A Dung's abstract argumentation framework is composed of arguments and attacks among them. Reasoning with such a framework consists in evaluating arguments using several approaches: extensions-based, labelling-based and ranking-based.*

3

Using Deductive Argumentation with Existential Rules

3.1 Deductive argumentation frameworks in existential rules **49**
3.1.1 Argumentation graphs generated from knowledge bases 52
3.1.2 Argumentation graphs generated from knowledge bases without rules 56
3.2 Improving the argument generation **64**
3.2.1 Optimisation for knowledge bases without rules . . 65
3.2.2 Optimisation for knowledge bases with rules 66
3.3 The DAGGER tool **73**
3.3.1 DAGGER's architecture 74
3.3.2 Usability scenarios 74
3.4 Benchmarks on logic-based argumentation frameworks **78**
3.4.1 Benchmark generation 79
3.4.2 Results of literature solvers over the benchmark . . 81
3.5 Summary **86**

Logic-based argumentation considers constructing arguments from inconsistent knowledge bases and computing attacks between them. The result of such a workflow is usually an *argumentation graph* (also called *argumentation framework*) where nodes are arguments and directed edges are attacks between them. In this thesis, we focus on instantiating a specifically crafted deductive argumentation framework using the existential rules language. The reason why we did not use any of the existing frameworks is because they are not *directly and straightforwardly applicable* in the context of the existential rules language. Indeed, none of the aforementioned frameworks can be applied to an inconsistent existential rules knowledge base without modifying it beforehand.

In the case of ABA, although it is abstract enough to function with a language that has neither implication nor negation, it needs a contrariness function that returns a single contrary sentence for each formula of the language. This is not enough in the case where a fact appears in multiple

conflicts and the language does not allow for the disjunction. We only say that ABA cannot be applied in a straightforward manner and not that it cannot be applied at all. Toni [2014] proposes a fix to the aforementioned problem that consists in adding new facts and rules in a specific way in order to encode a single contrary per assumption. In the case of ASPIC+ framework, we cannot instantiate it since the definition of the contrariness relation is not general enough to account for the existential rules negative constraints. Let us illustrate this on an example. Suppose we are given three facts: the biscuit has a square shape, the biscuit has a round shape and the biscuit is sweet; no rules and one negative constraint: the biscuit cannot have a square and round shape at the same time. As the fact "the biscuit is sweet" is a *free-fact* (i.e. it is not involved in any minimal conflict), there is no way to define its contrary in an intuitive manner (without modifying the knowledge base). In the work of Modgil and Prakken [2014], the third item of Definition 5.1 specifies that each formula of the language must have at least one contradictory, which is not the case for the latter fact in our example. Of course, workarounds exist but would necessitate the addition of multiple facts, positive rules and negative constraints. In the case of DeLP, we cannot instantiate it since the original work only consider ground rules which cannot encompass existential rules. Last not but least, the approach of Besnard and Hunter [2001] cannot be used directly as it was defined originally for classical propositional or full first-order logic [Besnard and Hunter, 2008].

The chapter is organized as follows: In Section 3.1, we revisit the particular deductive argumentation framework for existential rules proposed by Arioua et al. [2017] and show various properties results. In Section 3.2, we provide various optimisation for the arguments generation with respect to the knowledge base structure. In Section 3.3, we showcase the first tool in the literature for automatically generating the argumentation graph from an inconsistent knowledge base expressed in existential rules. In Section 3.4, we benchmark the top solvers of the ICCMA competition on the generated graphs and show that the structure of the generated graphs have an impact on argumentation solvers.

Research Questions in this Chapter

- *How can one generate a deductive argumentation framework from an inconsistent knowledge base expressed in existential rules?*
- *How can we get an efficient argument generation?*
- *Do these generated argumentation graphs possess some particular structure? If yes, does it have an impact on solvers?*

3.1 Deductive argumentation frameworks in existential rules

A deductive argumentation framework as described by Besnard and Hunter [2001] is composed of arguments with a support (or equivalently, hypothesis) and a conclusion that is derived from the support using the rule applications. The first deductive argumentation framework for the existential rules language was defined in the work of Croitoru and Vesic [2013] where arguments correspond to sequences of rule applications. It was proven that the aforementioned framework possesses several desirable properties such as the equivalence between repairs and preferred (respectively stable) extensions, the equivalence between intersection of extensions and some inconsistency tolerant semantics but it also satisfies argumentation properties defined by Caminada and Amgoud [2007]. As such, this particular deductive argumentation framework was the first link between the semantics used in inconsistent ontological knowledge base query answering and those from argumentation theory.

However, in practice, representing arguments as derivations is often redundant as some atoms can be derived from the same set of facts in different ways. We illustrate this intuition in the following example.

Example 3.1 (Multiple derivations for a fact). *Let us consider the knowledge base* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *such that:*

- $\mathcal{F} = \{p(a)\}$
- $\mathcal{R} = \{r_1 = \forall X(p(X) \rightarrow q(X)),$
  $r_2 = \forall X(p(X) \rightarrow r(X)),$
  $r_3 = \forall X(r(X) \rightarrow q(X))\}$
- $\mathcal{N} = \emptyset$

We obtain two following derivations for the fact $q(a)$*:*

$$\delta_1 = \langle(\mathcal{F}_0 = \{p(a)\}, \emptyset, \emptyset), (\mathcal{F}_1 = \mathcal{F}_0 \cup \{q(a)\}, r_1, \pi_1 = \{X \rightarrow a\})\rangle.$$

$$\delta_2 = \langle(\mathcal{F}_0 = \{p(a)\}, \emptyset, \emptyset), (\mathcal{F}_2 = \mathcal{F}_0 \cup \{r(a)\}, r_2, \pi_2 = \{X \rightarrow a\}),$$
$$(\mathcal{F}_3 = \mathcal{F}_2 \cup \{q(a)\}, r_3, \pi_3 = \{X \rightarrow a\})\rangle.$$

If we use the framework of Croitoru and Vesic [2013], we will get two arguments with the same support $\{p(a)\}$ *and conclusion* $\{q(a)\}$*.*

Although remembering which rule application is useful for many purposes such as debates or explanation, some devices with limited memory capacity cannot afford to keep all of those derivations. That is why, in both

the work of Arioua et al. [2017] and Yun et al. [2017a], a new deductive argumentation framework for the existential rules framework was proposed where the support of arguments are now composed of a minimal set of $\mathcal{R}$-consistent facts and the conclusion is only a set of atoms that is derived from the support. This new argumentation framework is more compact as it does not keep which rule applications led to the conclusion from the support.

Definition 3.1 (Deductive argument [Arioua et al., 2017; Yun et al., 2017a]). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base. An argument a is a tuple (H, C) with H a non-empty $\mathcal{R}$-consistent subset of $\mathcal{F}$ and C a set of facts such that:*

- *$H \subseteq \mathcal{F}$ and $Sat_{\mathcal{R}\cup\mathcal{N}}(H) \not\models \bot$* (consistency)
- *$C \subseteq Sat_{\mathcal{R}}(H)$* (entailment)
- *there is no $H' \subset H$ such that $C \subseteq Sat_{\mathcal{R}}(H')$* (minimality)

The support H of an argument a is denoted by $Supp(a)$ and the conclusion C by $Conc(a)$. If X is a set of arguments, we denote by $Base(X) = \bigcup_{a \in X} Supp(a)$.

Example 3.2 (Deductive argument). *Consider the following knowledge base $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ describing the situation: If an animal is a dog then it has an owner, if an animal can make cat sounds then it is not a dog.*

- $\mathcal{F} = \{animal(tom), miaow(tom), dog(tom)\}$
- $\mathcal{R} = \{r_1 = \forall X(animal(X) \wedge dog(X) \rightarrow \exists Y ownerOf(Y, X)),$
  $r_2 = \forall X(animal(X) \wedge miaow(X) \rightarrow notDog(X))\}$
- $\mathcal{N} = \{\forall X(dog(X) \wedge notDog(X)) \rightarrow \bot\}$

There are fourteen arguments that can be created from $\mathcal{KB}$:

$a_1 = (\{animal(tom)\}, \{animal(tom)\})$

$a_2 = (\{miaow(tom)\}, \{miaow(tom)\})$

$a_3 = (\{animal(tom), miaow(tom)\}, \{animal(tom), miaow(tom)\})$

$a_4 = (\{animal(tom), miaow(tom)\}, \{notDog(tom)\})$

$a_5 = (\{animal(tom), miaow(tom)\}, \{animal(tom), notDog(tom)\})$

$a_6 = (\{animal(tom), miaow(tom)\}, \{miaow(tom), notDog(tom)\})$

$a_7 = (\{animal(tom), miaow(tom)\}, \{animal(tom), miaow(tom), notDog(tom)\})$

$a_8 = (\{dog(tom)\}, \{dog(tom)\})$

$$
\begin{aligned}
a_9 &= (\{animal(tom), dog(tom)\}, \{animal(tom), dog(tom)\}) \\
a_{10} &= (\{animal(tom), dog(tom)\}, \{owner(Null_0, tom)\}) \\
a_{11} &= (\{animal(tom), dog(tom)\}, \{animal(tom), owner(Null_0, tom)\}) \\
a_{12} &= (\{animal(tom), dog(tom)\}, \{dog(tom), owner(Null_0, tom)\}) \\
a_{13} &= (\{animal(tom), dog(tom)\}, \{animal(tom), dog(tom), owner(Null_0, tom)\}) \\
a_{14} &= (\{miaow(tom), dog(tom)\}, \{miaow(tom), dog(tom)\})
\end{aligned}
$$

Please note that we restrict ourselves to recognisable FES classes of existential rules where the chase is guaranteed to stop [Baget et al., 2011]. In this case, $Sat_{\mathfrak{R}}(H)$ is a finite set. In order to capture inconsistencies between arguments, we consider the binary attack relation of Croitoru and Vesic [2013] where an argument a attacks an argument b if and only if the union of the conclusion of a and an element of the support of b is $\mathfrak{R}$-inconsistent. Roughly speaking, Croitoru and Vesic [2013] give the intuition that this particular binary attack relation is enough to capture all of the conflicts since we work in the OBDA setting where all the inconsistency "comes from the fact".

Definition 3.2 (Attack relation). *An argument a attacks an argument b denoted by $(a, b) \in \mathscr{R}$ (or $a\mathscr{R}b$) if and only if there exists $\phi \in Supp(b)$ such that $Conc(a) \cup \{\phi\}$ is $\mathfrak{R}$-inconsistent.*

Example 3.3 (Example 3.2 cont'd). *We have an attack from argument a_4 to a_{10} since the set $\{notDog(tom)\} \cup \{dog(tom)\}$ is $\mathfrak{R}$-inconsistent. Please note that this attack relation is not symmetric. Here, we can see that a_{10} does not attack a_4.*

Now that we defined the structure of arguments and attacks, the argumentation graph corresponding to a knowledge base consists simply of all arguments and attacks that can be generated.

Definition 3.3 (Argumentation graph). *The argumentation graph instantiated over a knowledge base $\mathcal{KB}$ is denoted by $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$, where the set of arguments $\mathscr{A}$ and the set of attacks $\mathscr{R}$ follow from Definition 3.1 and Definition 3.2 respectively.*

Example 3.4 (Example 3.3 cont'd). *In our example, $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ where $\mathscr{A} = \{a_1, \ldots, a_{14}\}$ and $\mathscr{R}$ contains the 60 possible attacks on $\mathscr{A}$.*

Note that if the fact base of the knowledge base is $\mathfrak{R}$-consistent, then there will be only one extension which will contain all of the arguments (independently of the argumentation semantics) since there will be no attacks amongst the arguments. We thus restrict ourselves to the study of argumentation graphs that are generated from inconsistent knowledge bases, i.e. knowledge bases that have at least one minimal conflict of size at least two.

Name	Description
Finite	$\mathfrak{F}$ has a finite set of arguments
Non-trivial	$Ext_p(\mathfrak{F}) \neq \{\emptyset\}$
Coherent	$Ext_p(\mathfrak{F}) = Ext_s(\mathfrak{F})$
Relatively grounded	$GE = \bigcap Ext_p(\mathfrak{F})$
Well-founded	$Ext_c(\mathfrak{F}) = Ext_g(\mathfrak{F}) = Ext_p(\mathfrak{F}) = Ext_s(\mathfrak{F})$

Table 3.1: Classes of argumentation frameworks studied in the literature

The rest of this section is organised as follows. In Section 3.1.1, we focus on the argumentation graphs generated from general knowledge bases and study their structural properties. Likewise, in Section 3.1.2, we study argumentation graphs generated from knowledge bases without positive rules.

3.1.1 Argumentation graphs generated from knowledge bases

We first recall, in Table 3.1, the different classes of argumentation graphs defined in the literature [Coste-Marquis et al., 2005].

The next proposition shows that an argumentation graph generated from an inconsistent knowledge base has a finite number of arguments.

Proposition 3.1 (Finiteness [Arioua et al., 2017]). *Let $\mathcal{KB}$ be a finite inconsistent knowledge base. Then, $\mathfrak{F}_{\mathcal{KB}}$ is finite.*

Proposition 3.2 shows that if an argument has a repair as its support and the same repair as its conclusion, then the set containing only this argument is an admissible set. Please note that such an argument does not always exist because of the minimality condition on the support of an argument.

Proposition 3.2 (Sentinel [Arioua et al., 2017]). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, $A \in repairs(\mathcal{KB})$ and $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$. If the argument $a = (A, A) \in \mathscr{A}$ then $\{a\}$ is an admissible set.*

Example 3.5 (Example 3.2 cont'd). *The set $\{animal(tom), dog(tom)\}$ is a repair of $\mathcal{KB}$. Thus, it holds that $\{a_9\}$ is an admissible set.*

Proposition 3.3 shows that an argumentation graph generated from an inconsistent knowledge base has at least one non empty preferred extension.

Proposition 3.3 (Non-triviality [Arioua et al., 2017]). *Let $\mathcal{KB}$ be an inconsistent knowledge base. Then $\mathfrak{F}_{\mathcal{KB}}$ is non-trivial.*

Example 3.6 (Example 3.2 cont'd). *In our example, we have three preferred extensions. Namely, $Ext_p(\mathfrak{F}_{\mathcal{KB}}) = \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$ where:*

- $\varepsilon_1 = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$
- $\varepsilon_2 = \{a_1, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}\}$
- $\varepsilon_3 = \{a_2, a_8, a_{14}\}$

The next proposition shows that, in an argumentation graph generated from an inconsistent knowledge base, each argument belongs to at least one preferred (respectively stable) extension.

Proposition 3.4 (Rejected argument [Arioua et al., 2017]). *Let $\mathcal{KB}$ be an inconsistent knowledge base. Then $\mathfrak{F}_{\mathcal{KB}}$ has no rejected arguments under preferred and stable semantics.*

Example 3.7 (Example 3.6 cont'd). *Every argument in $\mathscr{A} = \{a_1, \ldots, a_{14}\}$ is in at least one preferred extension of $Ext_p(\mathfrak{F}_{\mathcal{KB}}) = \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$.*

In general, working with argumentation frameworks is hard as shown by Dunne and Wooldridge [2009] and Dimopoulos et al. [1999]. However, when some properties are satisfied, this task can become easier. For instance, when the stable and preferred coincide, the problem of the skeptical membership of an argument is much easier. Indeed, Dunne and Bench-Capon [2002] have shown that checking whether an argument is in every stable extensions is coNP-complete whereas checking whether an argument is in every preferred extension is in the second level of the polynomial hierarchy ($\Pi_2^p - complete$). Another example is the problem of finding whether there is a unique extension with respect to an argumentation semantics and a given argumentation frameworks. Wolfgang [2017] shows that the uniqueness problem is DP-complete for the stable semantics whereas it is coNP-complete for the preferred semantics. Thus, coincidence between multiple argumentation semantics can sometimes induce a large reduction in complexity. In view of this, we highlight, in the next propositions and corollaries, the coincidences and inclusions results between argumentation semantics in this specific logic-based argumentation framework.

Proposition 3.5 shows that the set of preferred and stable extensions coincide. Although Coste-Marquis et al. [2005] showed that symmetric frameworks are always coherent, it has to be pointed out that the framework considered here is not symmetric.

Proposition 3.5 (Coherence [Arioua et al., 2017]). *Let $\mathcal{KB}$ be an inconsistent knowledge base. Then $\mathfrak{F}_{\mathcal{KB}}$ is coherent.*

Example 3.8 (Example 3.6 cont'd). *In our example, it holds that the set of preferred extensions is equal to the set of stable extensions. Namely, $Ext_p(\mathfrak{F}_{\mathcal{KB}}) = Ext_s(\mathfrak{F}_{\mathcal{KB}}) = \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$.*

In the next corollary, we show that the set of preferred extensions is included in the set of cf2 extensions. Note that the inverse inclusion does not hold.

Corollary 3.1 (Preferred and cf2 inclusion). *Let $\mathcal{KB}$ be an inconsistent knowledge base. Then it holds that $Ext_p(AS_{\mathcal{KB}}) \subseteq Ext_{cf2}(\mathfrak{F}_{\mathcal{KB}})$.*

Proof. Since we know that the set of stable extensions is included in the set of cf2 extensions [Gaggl and Woltran, 2013] and that the argumentation graph is coherent then we can conclude that the set of preferred extensions is included in the set of cf2 extensions □

Example 3.9 (Example 3.6 cont'd). *In our example, we have $Ext_p(\mathfrak{F}_{\mathcal{KB}}) = Ext_s(\mathfrak{F}_{\mathcal{KB}}) = Ext_{cf2}(\mathfrak{F}_{\mathcal{KB}}) = \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$. Please note that the equality is not true and that it is possible to find a knowledge base such that there is a cf2 extension of the generated argumentation graph that is not a preferred extension (see Example 3.19 on page 63).*

In the next corollary, we show that the semi-stable semantics is equivalent to the preferred and stable semantics in this argumentation framework.

Corollary 3.2 (Semi-stable equivalence). *Let $\mathcal{KB}$ be an inconsistent knowledge base then it holds that $Ext_p(AS_{\mathcal{KB}}) = Ext_{ss}(\mathfrak{F}_{\mathcal{KB}}) = Ext_s(\mathfrak{F}_{\mathcal{KB}})$.*

Proof. By definition, we know that $Ext_s(\mathfrak{F}_{\mathcal{KB}}) \subseteq Ext_{ss}(\mathfrak{F}_{\mathcal{KB}})$. Moreover, Caminada [2006] proved that $Ext_{ss}(\mathfrak{F}_{\mathcal{KB}}) \subseteq Ext_p(AS_{\mathcal{KB}})$ holds in the general case. Thus, since it holds that in this argumentation framework, we have $Ext_p(\mathfrak{F}_{\mathcal{KB}}) \subseteq Ext_s(\mathfrak{F}_{\mathcal{KB}})$, we can conclude the proof. □

The next proposition shows that the grounded extension is equal to the intersection of all the preferred extensions.

Proposition 3.6 (Relative groundedness [Arioua et al., 2017]). *Let $\mathcal{KB}$ be an inconsistent knowledge base. Then $\mathfrak{F}_{\mathcal{KB}}$ is relatively grounded.*

Example 3.10 (Example 3.6 cont'd). *In our example, we have that $Ext_g(\mathfrak{F}_{\mathcal{KB}}) = \{\emptyset\} = \bigcap Ext_p(\mathfrak{F}_{\mathcal{KB}})$.*

The next proposition shows that we can never create an argumentation graph that is well-founded.

Proposition 3.7 (Well-foundedness [Arioua et al., 2017]). *There is no inconsistent knowledge base $\mathcal{KB}$ such that $\mathfrak{F}_{\mathcal{KB}}$ is well-founded.*

In the next proposition, we show that self-attacking arguments do not exist in this framework.

Proposition 3.8 (No self-attacking arguments). *Let $\mathcal{KB}$ be an inconsistent knowledge base and $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ the corresponding argumentation framework. There is no $a \in \mathscr{A}$ such that $(a, a) \in \mathscr{R}$.*

Proof sketch. We show this by contradiction. If there is a self-attacking argument, then the union of its conclusion with its support is $\mathcal{R}$-inconsistent. This is a contradiction with the consistency of the support of an argument. □

In the next proposition, we show that if an argument a is attacked by an argument b, then b is also attacked.

Proposition 3.9 (Defense existence). *Let $\mathcal{KB}$ be an inconsistent knowledge base and $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ be the corresponding argumentation framework. For any $a, b \in \mathscr{A}$ such that $(a, b) \in \mathscr{R}$, there exists $c \in \mathscr{A}$ such that $(c, a) \in \mathscr{R}$.*

In the next proposition, we highlight that all subsets of a minimal inconsistent set can be directly "translated" into arguments.

Proposition 3.10 (Conflict-based arguments). *Let $\mathcal{KB}$ be an inconsistent knowledge base and $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ be the corresponding argumentation framework. If $C \in MI(\mathcal{KB})$ and $E \subset C$ then $(E, E) \in \mathscr{A}$.*

Proof. By definition, E is $\mathcal{R}$-consistent. Let us prove Proposition 3.10 by contradiction. Suppose that $(E, E) \notin \mathscr{A}$, it means that there exists $H \subset E$ with $E \subseteq Sat_{\mathcal{R}}(H)$ (minimality). Thus, $(C \setminus E) \cup H$ is $\mathcal{R}$-inconsistent and $((C \setminus E) \cup H) \subset C$, contradiction. □

Example 3.11 (Example 3.2 cont'd). *In our example, we have that $MI(\mathcal{KB}) = \{C\}$ where $C = \{animal(tom), miaow(tom), dog(tom)\}$. We can thus directly create the arguments corresponding to subsets of C, namely a_1, a_2, a_3, a_8, a_9 and a_{14}.*

In the next corollary, we show that if there is at least one minimal conflict of size at least 2, then there is a cycle in the argumentation graph.

Corollary 3.3 (Cycle existence). *Let $\mathcal{KB}$ be an inconsistent knowledge base such that there exists $S \in MI(\mathcal{KB})$ with $|S| \geq 2$ and $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ be the corresponding argumentation framework. $\mathfrak{F}_{\mathcal{KB}}$ has at least one cycle.*

Proof. Let us consider $S \in MI(\mathcal{KB})$ such that $|S| \geq 2$, $S_1, S_2 \subseteq S$, $S_1 \cap S_2 = \emptyset$, $|S_1| = 1$ and $S_1 \cup S_2 = S$. Using Proposition 3.10, we know that there exists two arguments $a_1 = (S_1, S_1)$ and $a_2 = (S_2, S_2)$ such that $(a_1, a_2) \in \mathscr{R}$ and $(a_2, a_1) \in \mathscr{R}$. □

In the next corollary, we show that for each minimal inconsistent set, there is a specific subset of the argumentation graph that is complete.

Corollary 3.4 (Conflict-based complete graphs). *Let $\mathcal{KB}$ be an inconsistent knowledge base and $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ be the corresponding argumentation framework. If $C \in MI(\mathcal{KB})$ then there exists a subgraph of $\mathfrak{F}_{\mathcal{KB}}$ with $|C|$ arguments that is complete.*

Proof. Let us consider the set $\mathscr{A}_C = \{a \in \mathscr{A} \mid a = (E, E)$ with $E \subset C$ and $|E| = |C|-1\}$. It is easy to see that for every $a, b \in \mathscr{A}_C$, we have that $(a, b) \in \mathscr{R}$. Thus the restriction of $\mathfrak{F}_{\mathcal{KB}}$ to $\mathscr{A}_C$ is a complete directed graph. □

Example 3.12 (Example 3.2 cont'd). *It holds that the subgraph of $\mathfrak{F}_{\mathcal{KB}}$ composed of the arguments $\mathscr{A}_C = \{a_4, a_9, a_{14}\}$ is a complete graph.*

In this section, we studied the structural properties, inclusion in graph classes and equivalence between argumentation semantics of argumentation graphs generated from general inconsistent knowledge bases. In the next section, we restrict ourselves to knowledge bases without positive rules.

3.1.2 Argumentation graphs generated from knowledge bases without rules

The graph theoretical results of this section are solely looking at the case where the knowledge base is composed of a set of facts and a set of negative constraints defined on these facts. Therefore, at the basis of our results lies the notion of knowledge base minimal conflict (and thus repair). Please note that knowledge bases without positive rules are not uncommon and without interests. Indeed, in the Big data setting, the data sources are sometimes presented in their saturated form, i.e. after the ontological rules have been applied. In this context, our results are directly applicable. In this section, we exhibit three main results proven by Yun et al. [2018b]:

1. The first result deals with the conflict-induced structural properties. Namely, we characterise *dummy arguments*, arguments that are unattacked and that do not attack other arguments, and show the repetitious nature of the argumentation graph by introducing the notion of k-copy graph.

2. The second result deepens these results and looks into the symmetries of the argumentation graph based on graph automorphisms.

3. Last, we look into the connectivity of the graph and demonstrate strongly connected components related results.

Please note that these three points will enable us to completely characterise the structural properties of argumentation graphs generated from knowledge bases without positive rules.

We begin by proving that the number of dummy arguments is exponential with respect to the number of free facts. Proposition 3.11 is important as it shows that even when there is no rules in the knowledge base, the number of arguments can be exponentially increased when free facts are added.

Definition 3.4 (Dummy argument). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework. We say that $a \in \mathscr{A}$ is a dummy argument if and only if there is no $(x, y) \in \mathscr{R}$ such that $x = a$ or $y = a$.*

Proposition 3.11 (Characterisation of dummy arguments). *Let* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *be a knowledge base such that* $\mathcal{R} = \emptyset$ *and* $|\mathcal{F}| = n$*. There are exactly* $2^k - 1$ *dummy arguments in* $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$*, where* $k = |Free(\mathcal{KB})|$*.*

Proof sketch. Since we can build $(2^k - 1)$ $\mathcal{R}$-consistent subsets from the set of free facts, the number of dummy arguments is at least $2^k - 1$. Then, by means of contradiction, we show that there cannot be a dummy argument with a support that is not included in the set of free facts (cf. detailed proof in Section 7.2.1 on page iii). □

Example 3.13 (Characterisation of dummy arguments). *Let us consider the following knowledge base* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *inspired from the* impossible food triad problem *proposed by George W. Hart:*

- $\mathcal{F} = \{potatoes(m), mayonnaise(m), cabbage(m), dish(m), edible(m)\}$
- $\mathcal{R} = \emptyset$
- $\mathcal{N} = \{\forall X(potatoes(X) \wedge mayonnaise(X) \wedge cabbage(X) \rightarrow \bot)\}$.

The knowledge base $\mathcal{KB}$ *expresses the idea that an edible dish* m *cannot contain potatoes, mayonnaise and cabbage at the same time. We have that* $Free(\mathcal{KB}) = \{dish(m), edible(m)\}$*, we conclude that there is* $2^2 - 1 = 3$ *dummy argument. Those arguments correspond to:*

- $(\{dish(m)\}, \{dish(m)\})$
- $(\{edible(m)\}, \{edible\})$
- $(\{edible(m), dish(m)\}, \{edible(m), dish(m)\})$.

We now analyse the related behaviour of atoms in at least one conflict. To do so, we introduce the notion of k-copy graph. A k-copy graph of an argumentation graph is another graph that has k times more arguments and each copy a' of an argument a attacks the same arguments as a and is attacked by the same arguments. Formally:

Definition 3.5 (k-copy graph). *Let* $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ *and* $\mathfrak{F}' = (\mathscr{A}', \mathscr{R}')$ *be two argumentation frameworks. We say that* $\mathfrak{F}$ *is a* k*-copy graph of* $\mathfrak{F}'$ *if and only if:*

- $|\mathscr{A}| = k * |\mathscr{A}'|$ *and there is a surjective function* f *from* $\mathscr{A}$ *to* $\mathscr{A}'$ *such that for every argument* $a' \in \mathscr{A}'$*, we have* $|W_{a'}| = k$*, where* $W_{a'} = \{a \in \mathscr{A} \mid f(a) = a'\}$*.*
- *For all* $a, b \in \mathscr{A}$*,* $(a, b) \in \mathscr{R}$ *if and only if* $(f(a), f(b)) \in \mathscr{R}'$*.*

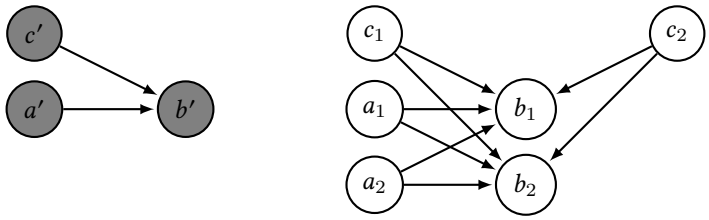

Figure 3.1: Representation of a 2-*copy* graph

Example 3.14 (k-copy graph). *In Figure 3.1, the graph G (on the right) is a 2-copy graph of the graph G′ (on the left). In our example, we have that* $W_{a'} = \{a_1, a_2\}, W_{b'} = \{b_1, b_2\}, W_{c'} = \{c_1, c_2\}$.

Please note that if two arguments are the copies of the same argument, then they attack the same arguments and are attacked by the same arguments.

Definition 3.6 (Subgraph). *Let* $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ *be an argumentation framework. We say that* $\mathfrak{F}' = (\mathscr{A}', \mathscr{R}')$ *is a subgraph of* $\mathfrak{F}$ *if and only if* $\mathscr{A}' \subseteq \mathscr{A}$, $\mathscr{R}' \subseteq \mathscr{R}$ *and for every* $(x, y) \in \mathscr{R}'$, $x, y \in \mathscr{A}'$.

The following proposition shows that if there is a knowledge base $\mathcal{KB}$ with no rule and k free facts, then there exists a subgraph of $\mathfrak{F}_{\mathcal{KB}}$ that is a 2^k-*copy* graph of $\mathfrak{F}_{\mathcal{KB}'}$ where $\mathcal{KB}'$ is the knowledge base with no rules, the same negative constraints as $\mathcal{KB}$ and that contains only the facts that are in at least one conflict. Proposition 3.12 is important as it shows the behaviour of the instantiation in the case of addition of free facts (facts not appearing in conflicts). It shows the structure of the graph and exhibits the exponential growth of the number of arguments with respect to these facts.

Proposition 3.12 (Number of arguments). *Let* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *be a knowledge base with* $\mathcal{R} = \emptyset$. *If* $|Free(\mathcal{KB})| = k$ *then there is a subgraph of* $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ *that is a* (2^k)*-copy graph of* $\mathfrak{F}_{\mathcal{KB}'} = (\mathscr{A}', \mathscr{R}')$ *where* $\mathcal{KB}' = (\mathcal{F} \setminus Free(\mathcal{KB}), \mathcal{R}, \mathcal{N})$ *and* $|\mathscr{A}| = (|\mathscr{A}'| + 1) * 2^k - 1$.

Proof sketch. If the set of free facts is empty, then it is obvious that $\mathfrak{F}_{\mathcal{KB}}$ is a 1-*copy* graph of itself. If the set of free facts is not empty, we consider $\mathfrak{F}_{\mathcal{KB}'} = (\mathscr{A}', \mathscr{R}')$ where $\mathcal{KB}'$ is the same knowledge base without the free facts. We show that the number of arguments in $\mathscr{A}'$ is the same as the number of $\mathcal{R}$-consistent subsets of $\mathcal{KB}'$. Last, we show that the subgraph $\mathfrak{F}''_{\mathcal{KB}} = (\mathscr{A}'', \mathscr{R}'')$ of $\mathfrak{F}_{\mathcal{KB}}$ where $\mathscr{A}'' = \{a \in \mathscr{A} \mid Supp(a) \not\subseteq Free(\mathcal{KB})\}$ and $\mathscr{R}'' = \mathscr{R}_{|\mathscr{A}''}$ is a $(2^{|Free(\mathcal{KB})|})$-*copy* graph of $\mathfrak{F}_{\mathcal{KB}'}$ (cf. detailed proof in Section 7.2.1 on page iii). □

We want to emphasise the result of Proposition 3.12 as it shows that the addition of "superfluous" facts will increase the size of the argumentation graph by an exponential factor.

Example 3.15 (Example 3.13 cont'd). *The argumentation framework* $\mathfrak{F}_{\mathcal{KB}}$ *has a subgraph that is a 4-copy graph of* $\mathfrak{F}_{\mathcal{KB}'}$*, where* $\mathcal{KB}' = (\{potatoes(m), mayonnaise(m), cabbage(m)\}, \emptyset, \mathcal{N})$*. We show below the list of arguments of the argumentation framework* $\mathfrak{F}_{\mathcal{KB}'}$*:*

- $a'_1 : (\{potatoes(m)\}, \{potatoes(m)\})$
- $a'_2 : (\{mayonnaise(m)\}, \{mayonnaise(m)\})$
- $a'_3 : (\{potatoes(m), mayonnaise(m)\}, \{potatoes(m), mayonnaise(m)\})$
- $a'_4 : (\{cabbage(m)\}, \{cabbage(m)\})$
- $a'_5 : (\{potatoes(m), cabbage(m)\}, \{potatoes(m), cabbage(m)\})$
- $a'_6 : (\{mayonnaise(m), cabbage(m)\}, \{mayonnaise(m), cabbage(m)\})$

The subgraph of $\mathfrak{F}_{\mathcal{KB}}$ *that is a 4-copy graph of* $\mathfrak{F}_{\mathcal{KB}'}$ *has four times more arguments than* $\mathfrak{F}_{\mathcal{KB}'}$ *because a copy of an argument is obtained by adding a set of free facts in its support and conclusion.*

We now focus on detecting symmetries in the graph. Please first note that we have the presence of symmetric arcs in the argumentation framework without rules. It obviously holds that if all negative constraints are binary, then the graph has only symmetric arcs (since the undermining will rely on binary sets). However, if the set of rules is not empty the symmetry of the attack relation no longer holds.

Next, we explore the link between the instantiation and symmetries in graphs. The following definitions introduce the notions needed to comprehend symmetries, namely, permutations of arguments, orbit of an argument and the cycle notation of a permutation.

Definition 3.7 (Permutation). *A permutation on a set of elements* X *is a bijection* σ *from* X *to* X*. Given a permutation* σ*, the orbit of element* $x \in X$ *is the set* $\mathcal{O}_x = \{x, \sigma(x), \sigma^2(x), \dots, \sigma^n(x)\}$*, with* $n \in \{0, 1, \dots\}$ *the minimal integer such that* $\sigma^{n+1}(x) = x$*.*

Example 3.16 (Permutation). *Let us consider the set* $X = \{1, 2, 3, 4, 5\}$ *and the permutation* σ *such that the images of* X *under* σ *are given in Table 3.2. The orbit of the element* 1 *is* $\mathcal{O}_1 = \{1, 2, 5\}$*.*

Definition 3.8 (Orbit cycle). *Given a permutation* σ *on* X*, an orbit* $\mathcal{O}$ *and an element* $x \in \mathcal{O}$*, an orbit cycle of* $\mathcal{O}$ *is a sequence* $(x, \sigma(x), \sigma^2(x) \dots, \sigma^n(x))$*, where* $n \in \{0, 1, \dots\}$ *is the minimal integer such that* $\sigma^{n+1}(x) = x$*.*

x	1	2	3	4	5
$\sigma(x)$	2	5	4	3	1

Table 3.2: Images of the permutation σ on X

A permutation can be compactly expressed as a product of cycles corresponding to the orbits of the permutation. In the rest of the thesis, and in order to simplify the notation, we omit cycles of singleton orbits.

Example 3.17 (Example 3.16 cont'd). *Let us consider the previous orbit* $\mathcal{O}_1$*. The sequence* $(1,2,5)$ *is an orbit cycle of* $\mathcal{O}_1$ *whereas the sequence* $(1,5,2)$ *is not. Thus, the permutation* σ *can be expressed by* $(1,2,5)(3,4)$*.*

Definition 3.9 (Automorphism). *Let* $G = (V,E)$ *be a graph. A permutation* σ *on set* V *is an automorphism of* G *if and only if for every two nodes* $v_1, v_2 \in V$*, we have that* $(v_1, v_2) \in E$ *if and only if* $(\sigma(v_1), \sigma(v_2)) \in E$*.*

The set of automorphisms of a graph, together with the function composition operator, form a group called the automorphism group. The automorphism groups of a graph characterise its symmetries, and are therefore very useful in determining certain of its properties. A subset of a group is called *a generating set of a group* if and only if every group's element can be expressed as the combination (under group operation) of finitely many elements of the subset and their inverses.

Proposition 3.13 (Automorphisms in k-copy graphs). *Let* $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ *be a* k*-copy graph of* $\mathfrak{F}' = (\mathscr{A}', \mathscr{R}')$*. For every* $a' \in \mathscr{A}'$*, for every* a_1, a_2 *in* $W_{a'}$*, the permutation* (a_1, a_2) *is an automorphism of* $\mathfrak{F}$*.*

The next proposition shows that if we add nodes (and no arc) to a graph with automorphisms, then the obtained graph also has automorphisms. It is used for showing, in Corollary 3.5, that a graph constructed on a knowledge base with no rules possesses non trivial automorphisms derived from its subgraph.

Proposition 3.14 (Automorphisms transfer). *Let* $G = (V,E)$ *be a graph such that* σ *is an automorphism of* G*. The graph* $G' = (V \cup X, E)$*, where* $X \cap V = \emptyset$*, has the automorphism* σ' *from* $V \cup X$ *to* $V \cup X$*:*

$$\text{for every } v \in V \cup X, \sigma'(v) = \begin{cases} \sigma(v) & \text{if } v \in V \\ v & \text{if } v \in X \end{cases}$$

Corollary 3.5 (Automorphisms inheritance). *Let* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *be a knowledge base with* $\mathcal{R} = \emptyset, |Free(\mathcal{KB})| = k, k > 0, \mathcal{KB}' = (\mathcal{F} \setminus Free(\mathcal{KB}), \mathcal{R}, \mathcal{N})$ *and* $\mathfrak{F}''$ *be a* (2^k)*-copy graph of* $\mathfrak{F}_{\mathcal{KB}'} = (\mathscr{A}', \mathscr{R}')$*. If* $\mathfrak{F}''$ *has* k' *automorphisms, then* $\mathfrak{F}_{\mathcal{KB}}$ *has at least* k' *automorphisms.*

Proof. From Proposition 3.12, we know that $\mathfrak{F}_{\mathcal{KB}}$ has a subgraph $\mathfrak{F}''_{\mathcal{KB}} = (\mathscr{A}'', \mathscr{R}'')$ that is a 2^k-*copy* graph of $\mathfrak{F}_{\mathcal{KB}'}$. We first show that every argument $a \in \mathscr{A} \setminus \mathscr{A}''$ is a dummy argument. Then we use Proposition 3.14.

1. We showed in the proof of Proposition 3.12 that $\mathscr{A}'' = \{a \in \mathscr{A} \mid Supp(a) \nsubseteq Free(\mathcal{KB})\}$. Thus, $\mathscr{A} \setminus \mathscr{A}'' = \{a \in \mathscr{A} \mid Supp(a) \subseteq Free(\mathcal{KB})\}$. Since we there are no rules, the arguments in $\mathscr{A} \setminus \mathscr{A}''$ cannot attack other arguments.

2. From Proposition 3.14, we conclude that there is an automorphism of $\mathfrak{F}_{\mathcal{KB}}$ for every automorphism of $\mathfrak{F}''_{\mathcal{KB}}$.

□

Corollary 3.5 is important as it shows that a graph inherits all of the automorphisms of its subgraph. This will be useful when designing new solvers relying on symmetries.

We now characterise the connectivity of the graph by showing the structure of the strongly connected components. We first define the impossible set associated to a minimal conflict C as the set containing all the possible subsets of $\mathcal{F}$ that are supersets of at least one subset of C of size $|C-1|$.

Definition 3.10 (Impossible set). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and C be a minimal conflict of $MI(\mathcal{KB})$. The impossible set of C denoted by $Imp(C)$ is $\{X \subseteq \mathcal{F} \mid X' \subseteq X$ and $X' \subset C$ with $|X'| = |C-1|\}$.*

In the following proposition, we characterise the structure of the strongly connected components of an argumentation framework obtained from a knowledge base without rules.

Proposition 3.15 (SCC characterisation). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that $\mathcal{R} = \emptyset$ and $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ be the corresponding argumentation framework. We have that:*

1. *$s_i \in SCC(\mathfrak{F}_{\mathcal{KB}})$ where $s_i = \{(X_i, X_i)\}$ with $X_i \in 2^{\mathcal{F}} \setminus \bigcup_{C \in MI(\mathcal{KB})} Imp(C)$*

2. *$(\mathscr{A} \setminus \bigcup_i s_i) \in SCC(\mathfrak{F}_{\mathcal{KB}})$*

Proof sketch. The proof is split in two parts. First, we show by contradiction that s_i is a strongly connected component by itself. Then, we show that all of the other arguments in $(\mathscr{A} \setminus \bigcup_i s_i)$ are strongly connected (cf. detailed proof in Section 7.2.1 on page iv).

□

Corollary 3.6 (Number of SCCs). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that $\mathcal{R} = \emptyset$. There are $|2^{\mathcal{F}} \setminus \bigcup_{C \in MI(\mathcal{KB})} Imp(C)| + 1$ strongly connected components in $\mathfrak{F}_{\mathcal{KB}}$.*

Example 3.18 (Example 3.13 cont'd). *The only minimal conflict is* $C_1 = \{potatoes(m), mayonnaise(m), cabbage(m)\}$. *Thus, we conclude that:*

$$2^{\mathcal{F}} \setminus \bigcup_{C \in MI(\mathcal{KB})} Imp(C) = \{\{potatoes(m)\}, \{mayonnaise(m)\}, \{cabbage(m)\}, \{dish(m)\}, \{edible(m)\}, \{dish(m), edible(m)\}, \{potatoes(m), dish(m)\}, \{potatoes(m), edible(m)\}, \{potatoes(m), dish(m), edible(m)\}, \{mayonnaise(m), dish(m)\}, \{mayonnaise(m), edible(m)\}, \{mayonnaise(m), dish(m), edible(m)\}, \{cabbage(m), dish(m)\}, \{cabbage(m), edible(m)\}, \{cabbage(m), dish(m), edible(m)\}\}$$

Therefore, there are $15 + 1 = 16$ *strongly connected components in* $\mathfrak{F}_{\mathcal{KB}}$.

We now summarise all the structural properties of the argumentation frameworks generated from simple knowledge bases using Figure 3.2 as an example:

- There is one k-copy graph (encircled in the dashed-line zone).
- The arguments that are not inside the k-copy graph are dummy arguments (arguments that are outside the dashed-line zone) and their number can be computed using Proposition 3.11.
- There is one dense strongly connected component composed of the majority of the arguments (encircled in the grey circle).
- The other strongly connected components are composed of only one argument each (arguments that are outside of the grey circle). The number of strongly connected components can be computed using Corollary 3.6.

Since we deal with strongly connected components, one of the research questions that naturally arises from this is whether or not the cf2 semantics [Baroni et al., 2011; Gaggl and Woltran, 2013] is equivalent to the preferred semantics in argumentation graphs generated from knowledge bases without positive rules.

On one hand, it appears that if the set of negative constraints is composed of only binary negative constraints, then the graph only has symmetric arcs. We conclude that since all SCCs are not linked to each other, the cf2 semantics coincides with the naive and preferred semantics.

Proposition 3.16 (Naive and preferred equivalence). *Let* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *be a knowledge base such that* $\mathcal{R} = \emptyset$. *The cf2 semantics coincides with the preferred (respectively stable) and the naive semantics in* $\mathfrak{F}_{\mathcal{KB}}$.

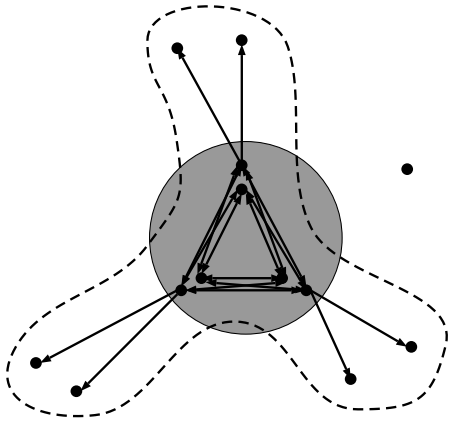

Figure 3.2: Structural properties of argumentation frameworks generated from simple knowledge bases

On the other hand, if we add ternary negative constraints, the cf2 semantics will no longer coincide with the preferred semantics as shown by a counter-example in Example 3.19.

Example 3.19. *Let* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *be a knowledge base such that* $\mathcal{F} = \{a(m), b(m), c(m), d(m), e(m)\}$, $\mathcal{R} = \emptyset$ *and* $\mathcal{N} = \{\forall X(a(X) \wedge b(X) \wedge c(X) \rightarrow \bot), \forall X(e(X) \wedge d(X) \rightarrow \bot)\}$. *The corresponding argumentation framework is composed of* 161 *attacks and* 20 *arguments. The set of preferred extensions is* $\{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5, \varepsilon_6\}$ *wheareas the set of cf2 extensions is the set* $\{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5, \varepsilon_6, \varepsilon_7, \varepsilon_8\}$. *The list of arguments and the composition of the extensions is given in Section 7.1 on page i.*

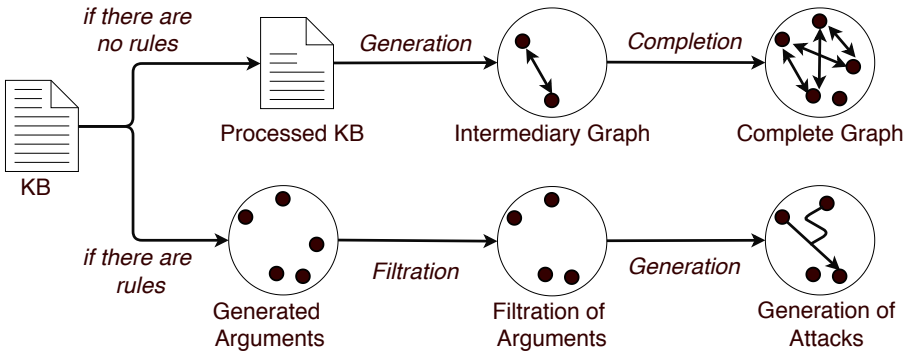

Figure 3.3: Approach workflow for optimising the argument generation phase.

3.2 Improving the argument generation

As mentioned in the previous section, the main drawback of using argumentation as a reasoning method over inconsistent knowledge bases relies in the large number of arguments generated. For instance, even for a modest knowledge base composed of seven facts, three rules and one binary negative constraint, one gets an argumentation graph with 383 arguments and 32768 attacks [Yun et al., 2017b]. In this section, we address this drawback and ask the following research question:

> "*How can one filter out the arguments generated over the knowledge base without compromising the semantical outcome of the corresponding argumentation graph?*"

We answer this question by providing a methodology adapted for knowledge bases with or without positive rules. In the first case of knowledge bases without rules, we use the observation that free facts induce an exponential growth on the argumentation graph without any impact on its underlying structure [Yun et al., 2017b] (see Definition 3.5 on page 57 about k-copy graph). Therefore, we will first generate the argumentation graph corresponding to the knowledge base without the free facts and then recreate the whole graph including the arguments of the free facts in an efficient manner. This method allows to generate the graph faster.

In the second case of the knowledge bases with rules, we introduce a new structure for the arguments and the attacks. In this new structure, we have less arguments (up to 73% filtered arguments in our experiments). We show that this new framework is semantically equivalent to the framework introduced by Croitoru and Vesic [2013]. The whole aforementioned methodology is represented in Figure 3.3.

Against this background, this section is organised as follows. In Section 3.2.1, we present the methodology above for argumentation graphs generated

from knowledge bases without positive rules. In Section 3.2.2, we present the other methodology above for argumentation graphs with positive rules. We then provide an empirical evaluation of our work in which we benchmark our approach on the knowledge bases introduced by Yun et al. [2017b] and show that in most of the cases, the number of arguments and attacks of the argumentation graphs corresponding to knowledge bases with rules is reduced (at least by 25 % for the arguments and at least 14 % for the attacks).

3.2.1 Optimisation for knowledge bases without rules

In this section, we propose an optimisation for the generation of the aforementioned argumentation framework in the case where knowledge bases contain no positive rules. The idea is to process the knowledge base before generating the argumentation graph and recreate the whole argumentation graph from this reduced graph.

In fact, as the number of free facts increases, the number of dummy arguments (non attacked arguments that do not attack other arguments) grows exponentially. However, a further result of Yun et al. [2017b] is that if one removes the free facts from the knowledge base before generating the argumentation graph, this argumentation graph possibly possesses "exponentially less arguments" with respect to the number of free facts compared to the original argumentation graph. Hence, we propose a four-step approach for generating the original argumentation graph faster:

1. We identify the set $Free(\mathcal{KB})$. This step can be done by finding the minimal inconsistent sets using existing algorithms [Grégoire et al., 2007; Rocher, 2013]

2. We create the graph $\mathfrak{F}_{\mathcal{KB}'}$ where $\mathcal{KB}' = (\mathcal{F} \setminus Free(\mathcal{KB}), \mathcal{R}, \mathcal{N})$ following Definition 3.3 on page 51. Please note that this step can be achieved using the argumentation graph generator proposed by Yun et al. [2017b].

3. Then, we grow the generated graph to its original size. This can be done by copying each arguments 2^k times (where $k = |Free(\mathcal{KB})|$) and adding attacks following the two principles: (1) if a attacks b then a attacks all the copies of b and (2) if b is a copy of a in then b has the same attackers and attacks the same arguments than a.

4. Last, we add $2^k - 1$ dummy arguments to the generated graph.

Example 3.20. *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that:*

- $\mathcal{F} = \{a(m), b(m), c(m)\}$
- $\mathcal{R} = \emptyset$

- $\mathcal{N} = \{\forall X(a(X) \wedge b(X) \rightarrow \bot)\}$.

In this example, $\mathit{Free}(\mathcal{KB}) = \{c(a)\}$. *Hence, we generate the argumentation graph* $\mathfrak{F}_{\mathcal{KB}'}$ *from the knowledge base* $\mathcal{KB}' = (\{\{a(m), b(m)\}, \emptyset, \mathcal{N})$ *(Step 1 in Figure 3.4). Then, from the graph of* $\mathfrak{F}_{\mathcal{KB}'}$*, one can construct the corresponding k-copy graph in Step 2. Finally, the dummy arguments are added (Step 3). At this point, we reconstructed* $\mathfrak{F}_{\mathcal{KB}}$.

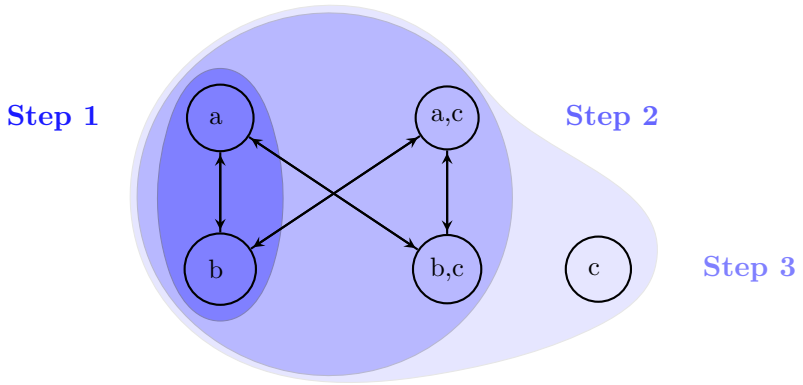

Figure 3.4: Three steps reconstruction using k-copy graphs

We would like to highlight that in the case where the set of positive rules is not empty, one cannot just remove the set of free facts as they can be used by rules and thus be used to build more complex arguments. We thus need a more complex approach to deal with general knowledge bases.

3.2.2 Optimisation for knowledge bases with rules

We now present a novel argumentation framework that aims at reducing the number of arguments and the number of attacks in the case where the set of rules is not empty. We show several desirable results such as the equivalence between the preferred and stable extensions of the aforementioned framework and the new one and some basic properties regarding attacks in the new framework. The idea behind this new framework is to remove, amongst the arguments with the same support, those that have conclusions that can be "decomposed". Let us illustrate the idea on Example 3.21.

Example 3.21. *Let us consider the following knowledge base* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *describing the situation: If* X *is a pitbull then* X *is a dog,* X *cannot be a dog and a cat at the same time.*

- $\mathcal{F} = \{pitbull(tom), cat(tom)\}$
- $\mathcal{R} = \{\forall X(pitbull(X) \rightarrow dog(X))\}$

- $\mathcal{N} = \{\forall X(dog(X) \wedge cat(X) \rightarrow \bot)\}$.

There are four arguments:

- $a_1 = (\{pitbull(tom)\}, \{pitbull(tom)\})$
- $a_2 = (\{pitbull(tom)\}, \{dog(tom)\})$
- $a_3 = (\{pitbull(tom)\}, \{dog(tom), pitbull(tom)\})$
- $a_4 = (\{cat(tom)\}, \{cat(tom)\})$

Our approach will delete the argument a_3 because we can "reconstruct" the argument a_3 from the two arguments a_1 and a_2.

Why do we only filter the arguments with the same support? Let us illustrate this with an example. Suppose that we also have the fact $adult(tom)$ in $\mathcal{KB}$. Amongst many others, we would have the following two arguments:

- $a_5 = (\{adult(Tom)\}, \{adult(Tom)\})$
- $a_6 = (\{pitbul(Tom), adult(Tom)\}, \{pitbul(Tom), adult(Tom)\})$

Could we remove the argument a_6 and reconstruct it from a_1 and a_5? We chose not to do this because it is not obvious that a_1 and a_5 are compatible with respect to the ontology and the negative constraints. That is the reason why we keep the argument a_6. On the contrary, note that a_1 and a_2 which have the same support must be compatible together. Let us now formalise this intuition.

Definition 3.11 (Filtrated set of arguments). *Let $\mathcal{KB}$ be a knowledge base and $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ be the argumentation framework constructed from $\mathcal{KB}$ using Definition 3.3 on page 51. Let $D(\mathfrak{F}_{\mathcal{KB}}) = \{a = (H, C) \in \mathscr{A} \mid$ there exists $X \subseteq \mathscr{A} \setminus \{a\}$ such that for every $b \in X, Supp(b) = H$ and $\bigcup_{b \in X} Conc(b) = C\}$. The filtrated set of arguments is $\mathscr{A}^* = \mathscr{A} \setminus D(\mathfrak{F}_{\mathcal{KB}})$.*

Example 3.22 (Example 3.21 cont'd). *In this example, the set $D(\mathfrak{F}_{\mathcal{KB}})$ is $\{a_3\}$ and the corresponding filtrated set of arguments is $\mathscr{A}^* = \{a_1, a_2, a_4\}$.*

Since we dropped some arguments, the attack relation have to be redesigned in order to keep all the conflicts. In particular, we need to allow for directed hyperedges (also called sets of attacking arguments) where arguments with the same support can jointly attack a single argument.

Definition 3.12 (Sets of attacking arguments). *An attack is a pair (X, a) where $X \subseteq \mathscr{A}^*$ and $a \in \mathscr{A}^*$ such that X is minimal for set inclusion set such that for every $x_1, x_2 \in X, Supp(x_1) = Supp(x_2)$ and there exists $\phi \in Supp(a)$ such that $(\bigcup_{x \in X} Conc(x)) \cup \{\phi\})$ is $\mathcal{R}$-inconsistent.*

The next example shows that this definition of attack is necessary in order to capture some attacks that would be lost otherwise.

Example 3.23 (Sets of attacking arguments). *Let us consider the following knowledge base* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *such that:*

- $\mathcal{F} = \{a(m), b(m), c(m)\}$
- $\mathcal{R} = \{\forall X(a(X) \wedge b(X) \rightarrow d(X) \wedge e(X))\}$
- $\mathcal{N} = \{\forall X(c(X) \wedge d(X) \wedge e(X) \rightarrow \bot)\}$

The list of arguments is represented in Table 3.3. We have the attack $(\{a'_3, a'_7\}, a'_{15})$ *where* $a'_3 = (\{a(m), b(m)\}, \{d(m)\}), a'_7 = (\{a(m), b(m)\}, \{e(m)\})$ *and* $a'_{15} = (\{c(m)\}, \{c(m)\})$*. Note that if the classical attack definition was used,* a'_{15} *would not be attacked anymore since we removed its attackers, for instance* $(\{a(m), b(m)\}, \{b(m), d(m), e(m)\})$*.*

Definition 3.13 (Filtrated argumentation framework). *Let* $\mathcal{KB}$ *be a knowledge base. The corresponding filtrated argumentation framework* $\mathfrak{F}^*_{\mathcal{KB}}$ *is the pair* $(\mathscr{A}^*, \mathscr{R}^*)$ *where* $\mathscr{A}^*$ *is as defined in Definition 3.11 and* $\mathscr{R}^*$ *is the set of all possible attacks that can be constructed using Definition 3.12.*

The proposed argumentation framework $\mathfrak{F}^*_{\mathcal{KB}}$ is an instantiation of the hypergraph argumentation framework proposed by Nielsen and Parsons [2006]. We briefly recall the necessary definitions of the extension-based semantics in this framework.

Definition 3.14 (Hypergraph argumentation semantics). *Let us consider the hypergraph argumentation framework* $\mathfrak{F}^*_{\mathcal{KB}} = (\mathscr{A}^*, \mathscr{R}^*)$*, we say that:*

- *A set of arguments* S *is* ***conflict-free*** *if and only if there is no argument* $a \in S$*, such that* $(S, a) \in \mathscr{R}^*$*.*
- *A set of arguments* S_1 ***attacks*** *a set of arguments* S_2 *if and only if there exists* $a \in S_2$ *such that* $(S_1, a) \in \mathscr{R}^*$*.*
- *An argument* a *is said to be* ***acceptable*** *with respect to a set of arguments* S*, if* S *defends* a *from all attacking sets of arguments in* a*.*
- *A set of arguments* S_1 ***defends*** *an argument* a *if and only if for every set of arguments* S_2 *such that* $(S_2, a) \in \mathscr{R}^*$*, we have that* S_1 *attacks* S_2*.*
- *A conflict-free set of arguments* S *is said to be* ***admissible*** *if each argument in* S *is acceptable with respect to* S*.*
- *An admissible set* S *is called a* ***preferred extension*** *if there is no admissible set* $S' \subseteq \mathscr{A}^*$*,* $S \subset S'$*.*

- *A conflict-free set S is a* ***stable extension*** *if S attacks all arguments in $\mathscr{A}^* \setminus S$.*

With a slight abuse of notation, we also use the notation $Ext_p(\mathfrak{F}^*_{\mathcal{KB}})$ (respectively $Ext_s(\mathfrak{F}^*_{\mathcal{KB}})$) to refer to the set of all preferred extensions (respectively stable extensions) of $\mathfrak{F}^*_{\mathcal{KB}}$. Let $X \subseteq \mathcal{F}$ and $\mathfrak{F}^*_{\mathcal{KB}} = (\mathscr{A}^*, \mathscr{R}^*)$, we denote by $Arg(X, \mathscr{A}^*)$, all the arguments in $\mathscr{A}^*$ such that their supports are included in X. Namely, $Arg(X, \mathscr{A}^*) = \{a^* \in \mathscr{A}^* \mid Supp(a^*) \subseteq X\}$.

In the next proposition, we show that we preserve the equivalence between the set of repairs and the set of preferred (respectively stable) extensions of $\mathfrak{F}^*_{\mathcal{KB}}$.

Proposition 3.17 (Repair equivalence). *Let us consider $\mathfrak{F}^*_{\mathcal{KB}} = (\mathscr{A}^*, \mathscr{R}^*)$. It holds that $Ext_x(\mathfrak{F}^*_{\mathcal{KB}}) = \{Arg(A', \mathscr{A}^*) \mid A' \in repairs(\mathcal{KB})\}$ for $i \in \{s, p\}$.*

Proof sketch. The proof is split in two parts. Let $\mathfrak{F}^*_{\mathcal{KB}} = (\mathscr{A}^*, \mathscr{R}^*)$ and $N = \{Arg(A', \mathscr{A}^*) \mid A' \in repairs(\mathcal{KB})\}$. First, we show that each set of arguments in N is a stable extension. Second, we show that each preferred extension is included in N. Last, since each stable extension is a preferred extension [Nielsen and Parsons, 2006], we can conclude the proof (see detailed proof in Section 7.2.1 on page v). □

Corollary 3.7 show that there is a one to one equivalence between the set of extensions of $\mathfrak{F}_{\mathcal{KB}}$ and $\mathfrak{F}^*_{\mathcal{KB}}$.

Corollary 3.7 (Extensions equivalence). *Let $\mathfrak{F}_{\mathcal{KB}}$ be an argumentation framework and $\mathfrak{F}^*_{\mathcal{KB}}$ be the corresponding filtrated argumentation framework. It holds that $Ext_x(\mathfrak{F}^*_{\mathcal{KB}}) = \{E \cap \mathscr{A}^* \mid E \in Ext_x(\mathfrak{F}_{\mathcal{KB}})\}$ with $x \in \{s, p\}$.*

Example 3.24 (Example 3.23 cont'd). *The argumentation framework $\mathfrak{F}_{\mathcal{KB}}$ is composed of 18 arguments and 51 attacks. The corresponding filtrated argumentation framework $\mathfrak{F}^*_{\mathcal{KB}}$ has 12 arguments and 66 attacks. The list of all the arguments and those filtrated (in grey) is represented in Table 3.3. There are three preferred and stable extensions in $\mathfrak{F}_{\mathcal{KB}}$:*

- $\varepsilon_1 = \{a'_0, a'_1, a'_2, a'_3, a'_4, a'_5, a'_6, a'_7, a'_8, a'_9, a'_{10}, a'_{11}, a'_{12}, a'_{13}, a'_{14}\}$
- $\varepsilon_2 = \{a'_0, a'_{15}, a'_{16}\}$
- $\varepsilon_3 = \{a'_1, a'_{15}, a'_{17}\}$.

Likewise, the preferred/stable extensions in $\mathfrak{F}^*_{\mathcal{KB}}$ are:

- $\varepsilon'_1 = \{a'_0, a'_1, a'_2, a'_3, a'_4, a'_5, a'_7, a'_8, a'_9\}$
- $\varepsilon'_2 = \{a'_0, a'_{15}, a'_{16}\}$
- $\varepsilon'_3 = \{a'_1, a'_{15}, a'_{17}\}$

Name	Description
a'_0	$(\{a(m)\}, \{a(m)\})$
a'_1	$(\{b(m)\}, \{b(m)\})$
a'_2	$(\{a(m), b(m)\}, \{a(m), b(m)\})$
a'_3	$(\{a(m), b(m)\}, \{d(m)\})$
a'_4	$(\{a(m), b(m)\}, \{a(m), d(m)\})$
a'_5	$(\{a(m), b(m)\}, \{b(m), d(m)\})$
a'_6	$(\{a(m), b(m)\}, \{a(m), b(m), d(m)\})$
a'_7	$(\{a(m), b(m)\}, \{e(m)\})$
a'_8	$(\{a(m), b(m)\}, \{a(m), e(m)\})$
a'_9	$(\{a(m), b(m)\}, \{b(m), e(m)\})$
a'_{10}	$(\{a(m), b(m)\}, \{a(m), b(m), e(m)\})$
a'_{11}	$(\{a(m), b(m)\}, \{d(m), e(m)\})$
a'_{12}	$(\{a(m), b(m)\}, \{a(m), d(m), e(m)\})$
a'_{13}	$(\{a(m), b(m)\}, \{b(m), d(m), e(m)\})$
a'_{14}	$(\{a(m), b(m)\}, \{a(m), b(m), d(m), e(m)\})$
a'_{15}	$(\{c(m)\}, \{c(m)\})$
a'_{16}	$(\{a(m), c(m)\}, \{a(m), c(m)\})$
a'_{17}	$(\{b(m), c(m)\}, \{b(m), c(m)\})$

Table 3.3: Arguments in $\mathfrak{F}_{\mathcal{KB}}$ obtained from the knowledge base of Example 3.23. Arguments in grey are filtrated and do not appear in $\mathfrak{F}^*_{\mathcal{KB}}$

Proposition 3.18 (Less arguments). *Let $\mathcal{KB}$ be a knowledge base, $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ be the corresponding argumentation framework and $\mathfrak{F}^*_{\mathcal{KB}} = (\mathscr{A}^*, \mathscr{R}^*)$ be the filtrated argumentation framework. Then it holds that $|\mathscr{A}^*| \leq |\mathscr{A}|$.*

However, it is not true that $|\mathscr{R}| \leq |\mathscr{R}^*|$.

Proposition 3.19 (Attack properties). *Let $\mathcal{KB}$ be a knowledge base, $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ be the corresponding argumentation framework and $\mathfrak{F}^*_{\mathcal{KB}} = (\mathscr{A}^*, \mathscr{R}^*)$ be the filtrated argumentation framework. It holds that:*

1. *$a \in \mathscr{A}^*$ is not attacked in $\mathfrak{F}_{\mathcal{KB}}$ if and only if a is not attacked in $\mathfrak{F}^*_{\mathcal{KB}}$*
2. *if $a \in \mathscr{A}^*$ is attacked in $\mathfrak{F}_{\mathcal{KB}}$ then $|Att^-_{\mathfrak{F}_{\mathcal{KB}}}(a)| \leq |Att^-_{\mathfrak{F}^*_{\mathcal{KB}}}(a)|$.*

Proof sketch. The proof for the first item is straightforward. In order to prove the second item, we create a particular function and show that it is injective. The reader is invited to read the detailed proof in Section 7.2.1 on page vii. □

The second item of Proposition 3.19 shows that the arguments that are not filtrated can only have more attackers. Indeed, for an arbitrary argument $a \in \mathscr{A}^\star$, if one of its attackers in $\mathfrak{F}_{\mathcal{KB}}$ is filtrated, there will be a set $S \subseteq \mathscr{A}^\star$ that will attack a in $\mathfrak{F}^\star$.

Please note that for an arbitrary filtrated argument $a = (Supp(a), Conc(a))$, it is not always possible to find a set of arguments $X = \{x_1, x_2, \ldots, x_n\}$ in $\mathfrak{F}^*_{\mathcal{KB}}$ such that for every $i \in \{1, \ldots, n\}, Supp(x_i) = Supp(a)$ and the conclusions of the arguments in X are *distinct* and the union of their conclusions is equal to the conclusion of a. The next example shows a counter-example.

Example 3.25 (Distinct conclusion). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that:*

- $\mathcal{F} = \{a(m), c(m)\}$
- $\mathcal{R} = \{\forall X(a(X) \rightarrow b(X))\}$
- $\mathcal{N} = \emptyset$

As one can see, the argument $d = (\{a(m), c(m)\}, \{a(m), c(m), b(m)\})$ is filtrated because of the two arguments $x_1 = (\{a(m), c(m)\}, \{a(m), c(m)\})$ and $x_2 = (\{a(m), c(m)\}, \{b(m), c(m)\})$. Note that here, it holds that $X = \{x_1, x_2\}$ satisfies $Supp(x_1) = Supp(x_2) = Supp(d)$ and $\bigcup_{x_i \in X} Conc(x_i) = Conc(d)$ but for all $x_1, x_2 \in X$, $Conc(x_1) \cap Conc(x_2) = \emptyset$ is not true.

Please note that in the case where the set of rules is empty, the set of filtrated arguments is empty and $\mathfrak{F}_{\mathcal{KB}}$ is equivalent to $\mathfrak{F}^*_{\mathcal{KB}}$.

Proposition 3.20 (No filtration). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that $\mathcal{R} = \emptyset$ and $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ be the corresponding argumentation framework. It holds that $D(\mathfrak{F}_{\mathcal{KB}}) = \emptyset$ and $\mathfrak{F}^*_{\mathcal{KB}} = (\mathscr{A}^*, \mathscr{R}^*)$ is such that $\mathscr{A}^* = \mathscr{A}$ and $(b, a) \in \mathscr{R}$ if and only if $(\{b\}, a) \in \mathscr{R}^*$.*

We end this section by discussing the efficiency of the new argumentation framework based on the filtration of arguments for reducing the number of arguments and attacks by comparing the number of arguments in $\mathfrak{F}_{\mathcal{KB}}$ and $\mathfrak{F}^*_{\mathcal{KB}}$. For our analysis, we chose to work with a particular subset of 108 knowledge bases (named b_1 to b_{108}) extracted from the study of Yun et al. [2017b]. Since this argumentation framework has been shown to grow exponentially with respect to the number of free facts, this particular set of knowledge bases was chosen for its small size. These knowledge bases were generated by fixing the size of the set of facts and successively adding negative constraints until saturation. This dataset is composed of knowledge bases with two to seven facts with different characteristics as shown in Table 3.5 on page 80.

We provide a generator based on the Graal Java Toolkit [Baget et al., 2015c] for directly generating $\mathfrak{F}^*_{\mathcal{KB}}$ from an inconsistent existential rules knowledge base expressed in DLGP format [Baget et al., 2015b]. This tool can be downloaded along with the dataset used in this section using the following link: `https://gite.lirmm.fr/yun/paper-comma-generator`.

In Table 3.4, we present the number of arguments and attacks in $\mathfrak{F}_{\mathcal{KB}}$ and $\mathfrak{F}^*_{\mathcal{KB}}$ along with the percentage of arguments filtered and the percentage of reduction of attacks. These two percentages are defined as:

$$\%Arg.\ Filtrated = \frac{|\mathscr{A}| - |\mathscr{A}^*|}{|\mathscr{A}|} \text{ and } \%Att.\ Reduction = \frac{|\mathscr{R}| - |\mathscr{R}^*|}{|\mathscr{R}|}$$

We can make the following observations:

1. This method does not provide any advantages in the case where the knowledge base is devoid of rules. Indeed, when there is no positive rules, Proposition 3.20 shows that there will be no filtrated arguments and the set of attacks will remain unchanged.

2. Although the instance with the highest percentage of reduction of attacks (here it is b_{12} with 88%) is also the instance with the highest percentage of arguments filtered (73%), this is not always the case. Indeed, the instances b_{10} and b_{13} both have a percentage of arguments filtered of 33% but they have a percentage of attacks filtered of 50% and 33% respectively.

3. Lastly, in all the instances with rules, there are less arguments and less attacks in $\mathfrak{F}^*_{\mathcal{KB}}$ compared to $\mathfrak{F}_{\mathcal{KB}}$. Please note that although it is

guaranteed that the number of arguments will be less or equal from Proposition 3.18, the number of attacks can increase in some cases (see Example 3.24).

Name of the KB	Median # arg. $\mathfrak{F}_{\mathcal{KB}}$	Median # att. $\mathfrak{F}_{\mathcal{KB}}$	Median # arg. $\mathfrak{F}^*_{\mathcal{KB}}$	Median # att. $\mathfrak{F}^*_{\mathcal{KB}}$	Median % arg. filtrated	Median % att. reduction
b_1 to b_6	17	80	17	80	0	0
b_{32}	4	6	4	6	0	0
b_{33} to b_{35}	8	24	8	24	0	0
b_{36} to b_{40}	17	96	17	96	0	0
b_{41} to b_{56}	36	380	36	380	0	0
b_7 to b_{12}	11	17.5	4.5	11	53.6	40.2
b_{13} to b_{18}	14	87.5	6	35	57.1	60.4
b_{19} to b_{28}	71	1280	41	704	38.7	37.5
b_{29} to b_{31}	16	29	8	21	50	26.7
b_{57} to b_{58}	8	13.5	6	11.5	25	14.8
b_{59} to b_{82}	28.5	303.5	15.5	173.5	46.2	45.9
b_{83} to b_{84}	12	34	9	24	25	30.1
b_{85} to b_{87}	24	129	12	57	50	55.8
b_{88} to b_{108}	85	1652	35	596	59.0	63.5

Table 3.4: Characteristics of the $\mathfrak{F}_{\mathcal{KB}}$ and $\mathfrak{F}^*_{\mathcal{KB}}$ generated from the knowledge bases.

3.3 The DAGGER tool

In this section, we introduce DAGGER: a generator for logic based argumentation frameworks instantiated from inconsistent knowledge bases expressed using *Datalog*± (or equivalently, existential rules language). The tool allows for the import of a knowledge base in DLGP (for Datalog Plus) format (see the work of Baget et al. [2015b] for a complete explanation of the DLGP syntax), the generation and visualisation of the corresponding argumentation graph. Furthermore, the argumentation framework can also be exported in the Aspartix format [Egly et al., 2008]. The DAGGER tool is currently available for download at `https://gite.lirmm.fr/yun/Dagger` and a demo video is available for viewing at `https://youtu.be/z96mDd9M6oE`.

While a lot of theoretical work in the past 23 years has focused, amongst others, on optimising the extension finding procedures [Gaggl, 2013; Lagniez et al., 2015], on the investigation of various extension notions [Baroni et al., 2011] or on the investigation of desirable properties of logic based instantiations [Amgoud, 2014; Modgil and Prakken, 2014], there are few tools that allow to generate an argumentation graph from a given knowledge base [Thimm, 2017]. Furthermore, the few available tools for reasoning using argumentation over inconsistent knowledge bases do not allow for further interoperability (allowing their output to be used by other tools).

The DAGGER tool allows any knowledge engineer to (1) input a knowledge base in a commonly used format, (2) generate, (3) visualise and (4) export the argumentation graph. This tool is very useful, especially for practical argumentation. Such scenario could be used when a non expert wants to reason, using argumentation, over a knowledge base in a particular domain ([Arioua et al., 2016; Tamani et al., 2014a,b], etc.). It could also be useful for investigating the theoretical properties of the generated argumentation framework. Given the fact that certain graph theoretical properties could radically improve the extension computation efficiency [Yun et al., 2017b] such visualisation could be a useful tool for argumentation specialists. Last, please note that, even when the knowledge base is modestly large, the corresponding argumentation graph can become truly immense [Yun et al., 2017b]. In this case, allowing tool interoperability that will directly and straightforwardly load a logically generated argumentation graph into efficient solvers [Thimm, 2017; Lagniez et al., 2015] can make the difference between time out errors and obtaining a result.

3.3.1 DAGGER's architecture

The layered architecture of the DAGGER tool is shown in Figure 3.5 and is detailed as follows:

- **High level:** This layer is mainly composed of the graphical user interface (GUI) that is used for the different interactions. It has a text

area that allows to enter a knowledge base expressed in the DGLP format (i.e. the format for expressing existential rules).

- **Mid level:** This layer is composed of the logical model: knowledge bases and argumentation frameworks.

- **Low level:** This layer is composed of the computational tools that allow the computation of the arguments, the attacks (via the Graal library) and the repairs (i.e. the extensions).

The information flow passes from the high level to the low level through the intermediate level using the different communication channels between modules.

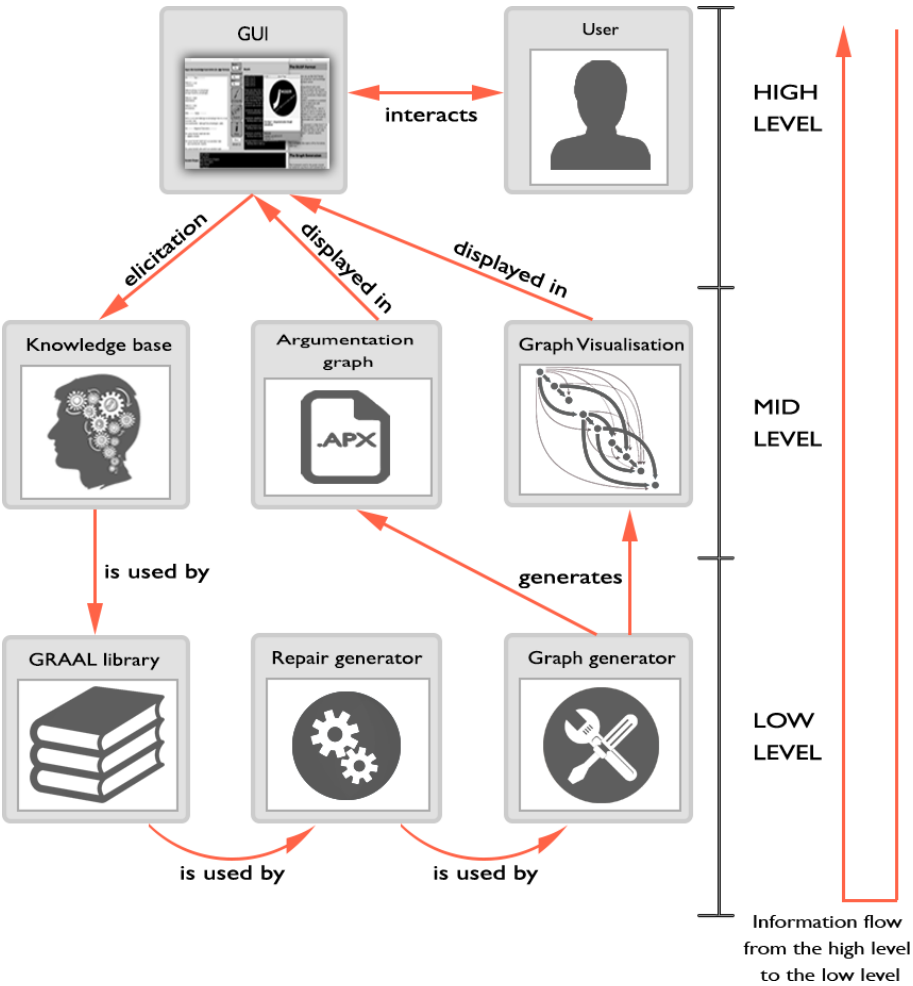

Figure 3.5: The 3-layer structure of DAGGER

3.3.2 Usability scenarios

We consider three usability scenarios of DAGGER. All of these scenarios are easily employed using DAGGER.

Scenario 1 First, we consider the task of a non computer science specialist inputting an inconsistent knowledge base of his expertise and wanting to find the maximally consistent point of views one can consider (see Figure 3.6). For instance, let us consider the knowledge base of Example 3.26. Please note that tools for assisting non domain experts in building such knowledge bases without computer expertise exists [Chein and Mugnier, 2009].

Figure 3.6: Screen capture of the main interface of the DAGGER tool and its repair computation module

Example 3.26. *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that:*

- $\mathcal{F} = \{packaging(a), has(a, plasticFilm), protectEnv(a)\}$
- $\mathcal{R} = \{\forall X(packaging(X) \wedge has(X, plasticFilm) \rightarrow pollute(X))\}$
- $\mathcal{N} = \{\forall X(pollute(X) \wedge protectEnv(X) \rightarrow \bot)\}$

In this knowledge base, a packaging a with a plastic film is said to protect the environment. However, since the possession of a plastic film leads

to pollution, this knowledge base is thus inconsistent. Finding maximally consistent point of views (or equivalently, repairs) consists in computing all maximal subsets of $\mathcal{F}$ that do not trigger a negative constraint of $\mathcal{F}$. Here, we have three repairs:

- $R_1 = \{packaging(a), has(a, plasticFilm)\}$,
- $R_2 = \{packaging(a), protectEnv(a)\}$
- $R_3 = \{has(a, plasticFilm), protectEnv(a)\}$

Scenario 2 Second, we consider an argumentation specialist looking for graph-based structural properties of argumentation graphs instantiated with particular knowledge bases (see Figure 3.7 and 3.8). For instance, let $\mathcal{KB}$ be a knowledge base with three facts $a(m), b(m), c(m)$, no rules and only containing one binary negative constraint $\forall X(b(X) \wedge c(X) \rightarrow \bot)$. By generating the graph representation one might observe that the graph is symmetrical thus satisfying certain restrictions over its extensions [Amgoud, 2014]. However, if one considers a ternary negative constraint that is added to the knowledge base (i.e. $\forall X(a(X) \wedge b(X) \wedge c(X) \rightarrow \bot)$), one can observe that the structure of the graph changes and it is no longer symmetric (and thus the properties of Amgoud [2014] do not hold anymore).

Scenario 3 Third, we consider a knowledge base composed of seven facts, two rules and one negative constraint. Generating the graph over such a knowledge base yields a graph of 383 arguments and 32768 attacks. Non optimised tools are not able to handle these large graphs for a computationally expensive operation such as finding all its extensions for a given argumentation semantics. However, ASPARTIX argumentation solvers based on SAT [Lagniez et al., 2015] will generate all extensions in less than one second. This is why one can use the export feature for such computations.

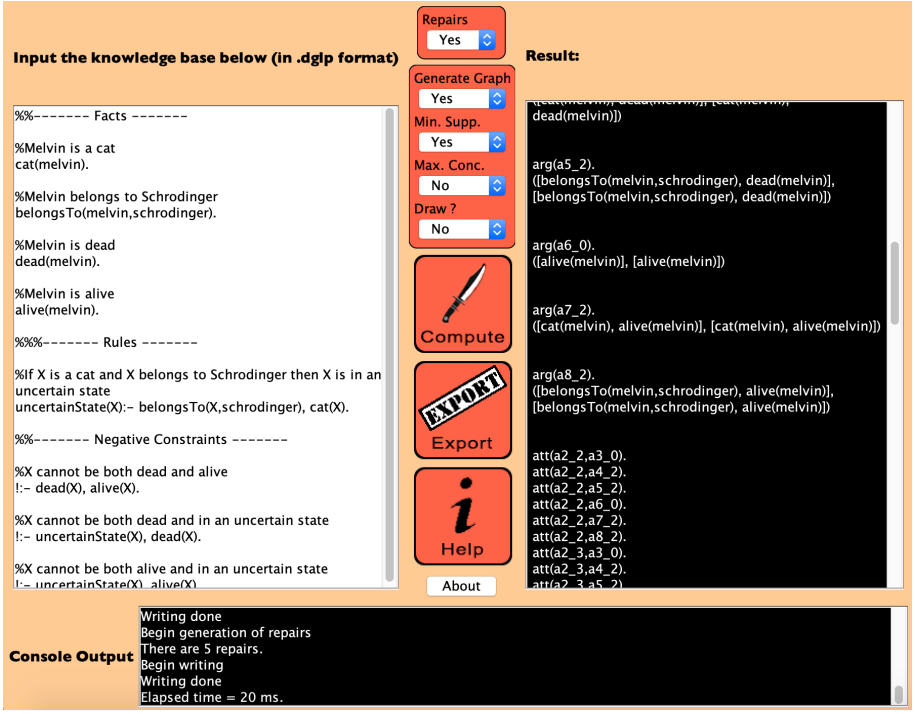

Figure 3.7: Screen capture of the main interface of the DAGGER tool and its argumentation module

3.4 Benchmarks on logic-based argumentation frameworks

Despite argumentation being a mature field, *practically inspired benchmarks are currently missing*. As a rare example of a practical argumentation benchmark consider NoDE[1], which contains graphs that model debates from Debatepedia[2], the drama "Twelve Angry Men" by Reginald Rose and Wikipedia revision history. However, the graphs from this benchmark are small (many of them have less than 10 arguments) and their structure is simplistic. The lack of benchmarks was acknowledged by the community long time ago, but became obvious with the appearance of the first International Competition on Computational Models of Argumentation (ICCMA)[3] in 2015. This is why new algorithms are always tested on randomly generated graphs, e.g. in the works of Nofal et al. [2014] and Cerutti et al. [2013].

The goal of this section is to address this drawback by *generating argu-*

[1] `http://www-sop.inria.fr/NoDE/`
[2] `http://debatepedia.org/`
[3] `http://argumentationcompetition.org/`

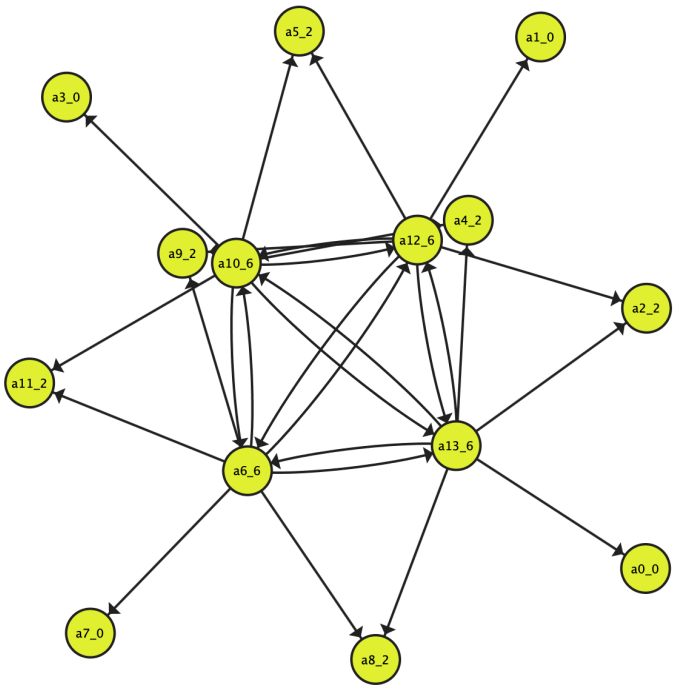

Figure 3.8: Screen capture of the argumentation graph interface of the DAGGER tool

mentation graphs from inconsistent knowledge bases and studying their properties empirically (by benchmarking argumentation solvers). We chose to instantiate the logic-based argumentation framework of Arioua et al. [2017] and Croitoru et al. [2015] as described in Definition 3.3 on page 51. As we explained in Section 3.1, this argumentation framework possesses many desirable properties and can be reused for query answering or exported in order to be reused by argumentation solvers.

We provide the first benchmark in the literature that uses graphs generated from knowledge bases expressed with existential rules instead of random graphs. Using a suite of parametrised existential rule knowledge bases, *we produced the first large scale practically-oriented benchmark in the literature.* Furthermore, we run the top six solvers from ICCMA 2015 *on the generated benchmark and show that the ranking is considerably different* from the one obtained during the competition on randomly generated graphs.

As seen in Section 3.1, the existential rules framework, as a logical language, provides many features (n-ary negative constraints, existential variables in the rule conclusion, etc.) that make the instantiated argumentation graph far from simplistic. Furthermore, the instantiated graph is reflecting the structure of the inconsistent knowledge bases and it is thus justifying its interest as practical benchmark. Generating such graphs is thus significant

for a broader community interested in reasoning in presence of inconsistency on the Semantic Web.

All experiments presented in the rest of this section were performed on a VirtualBox Linux machine running with a clean Ubuntu installation with one allocated processor (100%) of an Intel core i7-6600U 2.60GHz and 8GB of RAM. The argumentation graphs used (in the Aspartix format) are available online at `https://gite.lirmm.fr/yun/iccma-2019`.

The section is structured as follows. In Section 3.4.1, we discuss how we obtained the set of knowledge bases and the approach for generating the argumentation graphs. In Section 3.4.2, we study the results from benchmarking the top solvers from ICCMA on our generated argumentation graphs.

3.4.1 Benchmark generation

Knowledge Base Generation We generated a total of 134 knowledge bases: 108 different knowledge bases for the set of small graphs (denoted b_1 to b_{108}) and 26 for the set of big graphs accessible online at `https://github.com/anonymousIDA/Knowledge_bases`. This has been done in order to produce graphs of similar sizes to those of the *2015 International Competition on Computational Models of Argumentation* (ICCMA 2015). The ICCMA benchmark contains two sets of graphs: a set composed of small graphs (less than 383 arguments) and a set of big graphs (3783 to 111775 arguments). We define, for a fixed size of generated fact base (that varied from 2 to 5), some knowledge bases with binary (respectively ternary when applicable) constraints in order to obtain an incremental coverage of the facts. We then add rules in a similarly incremental manner. Table 3.5 shows the characteristics of the knowledge bases we selected. For example, if considering 3 facts $a(m), b(m), c(m)$, we chose a representative of binary constraints as $\forall X(a(X) \wedge b(X) \rightarrow \bot)$ or $\forall X(a(X) \wedge c(X) \rightarrow \bot)$ or $\forall X(b(X) \wedge c(X) \rightarrow \bot)$. We then chose $\forall X(a(X) \wedge b(X) \wedge c(X) \rightarrow \bot)$.

From Knowledge Bases to Argumentation Graphs In the argumentation graph generation process, we only kept knowledge bases whose argumentation frameworks were not automorphic to a previously generated graph. The knowledge base format is *DLGP* [Baget et al., 2015b], allowing translations to and from various Semantic Web languages such as RDF/S, OWL, RuleML or SWRL [Baget et al., 2015a]. For the graph generation, we made use of *Graal* [Baget et al., 2015c], a Java toolkit for reasoning within the framework of existential rules. Graal was used for storing the existential rule knowledge bases and for computing conflicts. On top of Graal we provided a graph generation program that works in two steps:

1. All possible arguments are generated: $\mathcal{R}$-consistent subsets of $\mathcal{F}$ are used as supports and conclusions are deduced from them. Then, non minimal arguments are removed (see Definition 3.1 on page 50).

Name of the KB	Number of facts	Number of rules	Number of NC	Type of NC	Number of Args	Number of Attacks
b_1 to b_6	2 to 7	$\emptyset$	1	Binary	2 to 95	2 to 2048
b_{32}	3	$\emptyset$	2	Binary	4	6
b_{33} to b_{35}	4	$\emptyset$	2 to 3	Binary	7 to 9	24 to 32
b_{36} to b_{40}	5	$\emptyset$	2 to 3	Binary	14 to 19	56 to 128
b_7 to b_{12}	2	1 to 6	1	Binary	4 to 30	5 to 240
b_{13} to b_{18}	2	2,4 or 6	1	Binary	6 to 30	15 to 450
b_{19} to b_{28}	2 to 7	1 or 3	1	Binary	11 to 383	32 to 32768
b_{29} to b_{31}	3	2	1	Binary	16	27 to 30
b_{57} to b_{58}	3	1	2	Binary	8	13 to 14
b_{59} to b_{82}	4	3	2 to 4	Binary	22 to 71	123 to 896
b_{41} to b_{56}	3 to 6	$\emptyset$	1 to 3	Ternary	6 to 55	9 to 752
b_{83} to b_{84}	3	1	1	Ternary	12	29 to 39
b_{85} to b_{87}	3	2	1	Ternary	24	93 to 147
b_{88} to b_{108}	4	3	1 to 2	Ternary	78 to 103	990 to 2496

Table 3.5: Characteristics of the small knowledge bases

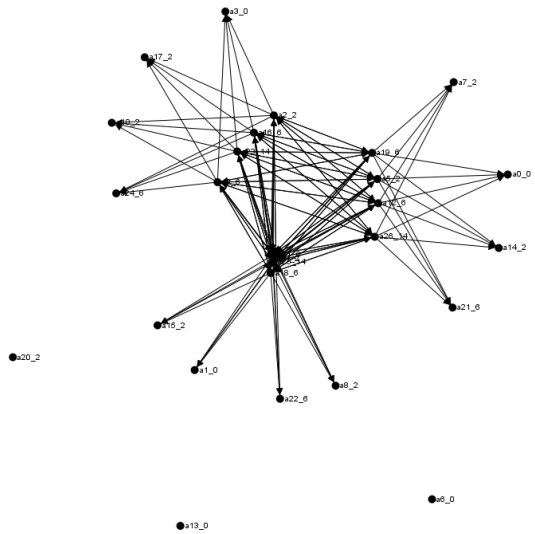

Figure 3.9: Representation of the argumentation graph corresponding to the knowledge base b_{44}

2. Attacks are computed following Definition 3.2 on page 51.

The obtained graphs were translated in the *Aspartix (apx)* format (the same format used in ICCMA 2015).

Example 3.27. *Let us consider the knowledge base* $b_{44} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *such that:*

- $\mathcal{F} = \{a(m), b(m), c(m), d(m), e(m)\}$
- $\mathcal{R} = \emptyset$
- $\mathcal{N} = \{\forall X(a(X) \wedge b(X) \wedge c(X) \rightarrow \bot)\}$

The corresponding argumentation graph $\mathfrak{G}_{b_{44}}$ *is composed of* 26 *arguments and* 144 *attacks and is represented in Figure 3.9. We show by this example that some of our generated graphs also possess a sense of "symmetry".*

In the next section, we report the results obtained from the run of the top six overall argumentation solvers on the proposed benchmark.

3.4.2 Results of literature solvers over the benchmark

We recall that the graphs used in the ICCMA 2015 benchmark were separated in three sets: a first set of large graphs (1152 to 9473 arguments) with large grounded extensions and an average density[4] of 1.00%, a second

[4]Graph density for a directed $G = (V, E)$ is equal to $\frac{|E|}{|V|(|V|-1)}$ where V is the set of nodes and E the set of arcs.

set of smaller graphs (141 to 400 arguments) with numerous complete/preferred/stable extensions and an average density of 3.68% and a third set of medium graphs (185 to 996 arguments) with rich structure of strongly connected components and an average density of 7.75%. Our benchmark graphs are denser, having an average density of 31.27% for small graphs and 29.69% for large graphs.

To see if the proposed benchmark graphs behave in a similar manner as the randomly generated graphs of ICCMA 2015, we ran the top six solvers of the competition: *CoQuiAAS*, *ArgSemSAT (ArgS.SAT)*, *LabSATSolver (LabSATS.)*, *ASGL*, *ASPARTIX-D* and *ArgTools (ArgT.)*. We used the solvers to complete two computational tasks: SE (given an abstract argumentation framework, determine some extensions) and EE (given an abstract argumentation framework, determine all extensions). These two computational tasks were to be solved with respect to the following standard semantics: *complete semantics* (CO), *preferred semantics* (PR), *grounded semantics* (GR) and *stable semantics* (ST).

In order to have similar assessment conditions, we used exactly the same ranking method as ICCMA 2015. The solvers were ranked with respect to the number of timeouts on these instances and ties were broken by the actual runtime on the instances. Table 3.6 shows the average time needed for each solver to complete each task for each semantics in the case of small graphs. There were no errors or time-outs thus the average time reflects the actual ranking (see Table 3.7).

For large instances, many solvers did not support large inputs resulting in several crashes or errors. Ties were broken by the average time of successfully solved instances (Table 3.9). Please note that for large graphs, for some tasks, some solvers timed out for all instances resulting in equal rankings (*EE-CO: ASGL and ArgTools* for instance).

Task	ArgS.SAT	ASGL	ArgT.	Aspartix-d	CoQuiAAS	LabSATS.
SE-CO	0,0138	0,1719	0,0059	0,0249	0,0031	0,3644
SE-PR	0,0165	0,2137	0,0059	0,4445	0,0007	0,2906
SE-GR	0,0339	0,2101	0,0057	0,3217	0,0010	0,1944
SE-ST	0,0148	0,2194	0,0060	0,0279	0,0018	0,2520
EE-CO	0,0694	0,2282	0,0096	0,0247	0,0024	0,2908
EE-PR	0,0517	0,1660	0,0085	0,5763	0,0029	0,3765
EE-GR	0,0325	0,1861	0,0052	0,3239	0,0016	0,2262
EE-ST	0,0486	0,1661	0,0065	0,0231	0,0027	0,3151

Table 3.6: Average computational time for small instances (in seconds)

It is noticeable that *CoQuiAAS* comes first in the two batches of gen-

Task	ArgS.SAT	ASGL	ArgT.	Aspartix-d	CoQuiAAS	LabSATS.
SE-CO	3	5	2	4	1	6
SE-PR	3	4	2	6	1	5
SE-GR	3	5	2	6	1	4
SE-ST	3	5	2	4	1	6
EE-CO	4	5	2	3	1	6
EE-PR	3	4	2	6	1	5
EE-GR	3	4	2	6	1	5
EE-ST	4	5	2	3	1	6

Table 3.7: Ranking of solvers for the generated small graphs

Task	ArgS.SAT	ASGL	ArgT.	Aspartix-d	CoQuiAAS	LabSATS.
SE-CO	15	26	16	1	0	11
SE-PR	18	1	17	26	0	11
SE-GR	17	0	18	26	0	11
SE-ST	15	0	18	2	0	11
EE-CO	22	26	26	9	2	21
EE-PR	21	26	26	26	15	21
EE-GR	16	26	17	26	0	11
EE-ST	15	23	17	1	1	11

Table 3.8: Number of timeouts for the generated large graphs

Task	ArgS.SAT	ASGL	ArgT.	Aspartix-d	CoQuiAAS	LabSATS.
SE-CO	4	6	5	2	1	3
SE-PR	5	2	4	6	1	3
SE-GR	4	2	5	6	1	3
SE-ST	5	2	6	3	1	4
EE-CO	4	5	5	2	1	3
EE-PR	2	4	4	4	1	3
EE-GR	3	5	4	5	1	2
EE-ST	4	6	5	2	1	3

Table 3.9: Ranking of solvers for the generated large graphs

erated graphs. This may come from the fact that *CoQuiAAS* is based on

MiniSAT solver, which is known to work well in the presence of structured information (i.e. symmetries). It might be the case that the generated graphs keep some of their structure even after being translated into a SAT instance which could explain the obtained result.

In order to see how different the solver ranking on the random benchmark used by ICMMA 2015[5] is from the solver ranking on the knowledge base benchmark, we used the normalised Kendall's tau distance.[6] The distance outputs 0 if two rankings are identical and 1 if one ranking is the reverse of the other. Table 3.11 shows the normalised Kendall's tau distance between the rankings of the generated graphs and the competition ranking. What comes out is that:

- Although the ranking of the ICCMA 2015 benchmark and the one for large graphs for the task EE-GR is slightly different (we can not break the tie between ASPARTIX-d and ASGL), they are identical with respect to the Kendall's tau distance.
- We have the same normalised Kendall's tau distance for the small graphs and the large graphs for the tasks SE-CO, SE-PR and SE-GR.
- The small graphs have a higher normalised Kendall's tau distance than the large graphs for the tasks SE-ST, EE-CO, EE-PR and EE-GR.
- The small graphs have a lower normalised Kendall's tau distance than the large graphs for the task EE-SET.
- In average, the results are more similar for the large graphs than for the small graphs.

This benchmark is interesting because it shows that for the instantiated graphs we generated, it is strongly advised to use CoQuiAAS as the solver. For relatively small graphs, the choice of the solver can be bypassed as the differences are negligible. However, for larger graphs, we noticed several issues:

- It seems that ASGL uses a different algorithm for SE-GR and EE-GR (this is very noticeable by the difference in the number of timeouts).
- ASGL is not suitable for finding complete extensions.
- Aspartix-D is not suitable for finding preferred and grounded extensions.
- There are 15 instances that were too big to perform the task EE-PR for all solvers.

[5] `http://argumentationcompetition.org/2015/results.html`

[6] This distance is equal to the number of pairwise disagreements between two ranking lists and is normalised by dividing by $\frac{n(n-1)}{2}$, where n is the number of solvers.

Task	ArgS.SAT	ASGL	ArgT.	Aspartix-d	CoQuiAAS	LabSATS.
SE-CO	4	2	5	3	1	6
SE-PR	1	4	6	5	3	2
SE-GR	3	5	4	6	1	2
SE-ST	2	5	6	1	4	3
EE-CO	2	5	6	1	3	4
EE-PR	1	4	6	5	2	3
EE-GR	3	5	4	6	1	2
EE-ST	2	4	5	1	3	6

Table 3.10: Rankings extracted from the ICCMA 2015 website

Task	Small graphs	Large graphs
SE-CO	0.400	0.400
SE-PR	0.467	0.467
SE-GR	0.200	0.200
SE-ST	0.600	0.467
EE-CO	0.467	0.200
EE-PR	0.400	0.067
EE-GR	0.267	0.000
EE-ST	0.333	0.400
Average	0.392	0.275

Table 3.11: Normalised Kendall's tau distance between the rankings of the generated graphs and the competition ranking

3.5 Summary

In this chapter, we presented a deductive argumentation framework for inconsistent knowledge bases expressed in the existential rules language. We showed that the instantiation of this argumentation framework possesses desirable properties in two cases: the general case (when the argumentation graph is generated from general inconsistent knowledge bases) but also in the case where the knowledge base has no positive rules.

In the general case, we showed that the generated graphs are included/not included in several graph classes (included in the class of coherent and relatively grounded), possess a specific structure (cycle, defense, no self-attacking arguments) and that some subgraphs can be characterised with respect to the conflicts of the knowledge base. In the case where the graph is generated from a knowledge base with no positive rules, we showed some conflict induced structural properties (dummy arguments characterisation, k-copy graphs), symmetry results and highlighted the general shape of strongly connected components. However, we also exhibited that this argumentation framework has an exponential number of arguments even in the case where the underlying knowledge has no positive rules.

Against this background, we presented two avenues for improving the generation of arguments and attacks: In the case where the graph is generated from a knowledge base without positive rules, one can filtrate the free facts from the knowledge base before the graph generation and recreate the full graph without loss of generality. However, the aforementioned strategy does not work in the case of an inconsistent knowledge base with positive rules since free facts can be used in rule applications. That is the reason why we propose a new method that consists in filtrating arguments that can be "reconstructed" by other arguments having the same support. However, since this method removes arguments and attacks, one needs to change the binary attack relation to sets of attacking arguments in order to keep the basic rationality desiderata. Furthermore, we showed that, in this new framework, most of the basic properties are still satisfied (repairs and extensions equivalence).

On the practical side, we also developed the DAGGER tool which is the first all-in-one implementation that can generate, allow for visualisation and export argumentation graphs from inconsistent knowledge bases expressed in the existential rule language.

Lastly, we used the DAGGER tool and designed an experiment using the top six solvers from the 2015 international competition on computational models of argumentation (see `http://argumentationcompetition.org/2015/`) in order to prove that the peculiar structure of the generated graphs have an impact on the solvers. The results shown that a solver outperformed all of the other solvers on our instances and that the general ranking was relatively different.

Chapter 3 in a Nutshell

- We showed several properties for the framework in the case where the argumentation graph is generated from a knowledge without positive rules (symmetry, dummy arguments, k-copy graphs, etc.) and in the general case (class of graph, special arguments, cycles, etc.)
- We showed that we can speed up the argument generation process by processing of the knowledge in the case where there are no positive rules. In the case where the knowledge has rules, we can filtrate the arguments and add hyperedges in order to reduce the number of arguments and attacks.
- DAGGER is an implementation of the deductive argumentation framework for the existential rules language and is the first tool that allows the generation, the visualisation and the exportation of such argumentation graphs.
- We empirically showed that the intrinsic structure of the generated argumentation graphs have an effect on the performance of the current top solvers for argumentation semantics.

4

Argumentation Hypergraphs

4.1 Argumentation hypergraphs with the existential rules language **91**

4.1.1 Hypergraph argumentation framework $\mathfrak{F}^\star$ 92

4.1.2 Argumentation framework properties 94

4.1.3 Rationality postulates 99

4.1.4 Empirical analysis 100

4.2 The NAKED tool **106**

4.2.1 The argument and attack generation 107

4.2.2 The structure of NAKED 108

4.2.3 Usability scenarios 110

4.3 Summary **112**

In the previous chapter, we showed that logic-based argumentation is an approach that can accommodate with reasoning in the context of inconsistent logic knowledge bases. Indeed, specifically crafted instantiations for *Datalog*± (such as the instantiation of Croitoru and Vesic [2013], Yun et al. [2017b] and Arioua et al. [2017]), have been proven to respect rationality desiderata [Amgoud, 2014; Caminada and Amgoud, 2007] and to output a set of extensions equivalent to the set of repairs [Lembo et al., 2010; Bienvenu, 2012] of the knowledge base (i.e. the maximum consistent sets of facts with respect to inclusion). Unfortunately, it was shown that these instantiations suffer from a major drawback: an exponential number of arguments and attacks [Yun et al., 2018d,b]. This problem occurs even in the case where there are no rules in the knowledge base (count, for instance, a graph with 13 arguments and 30 attacks for a meagre knowledge base with solely four facts, no positive rules and a single negative rule). As a consequence, the argumentation graph for a "normally-sized" knowledge base cannot be held in main memory, requires dedicated large-graph visualisation tools, and, despite their polynomial complexity regarding the number of arguments, still poses combinatorial challenges for the computation of ranking techniques. This is a big problem as it shows that the reasoning efficiency may be lacking particularly when compared to other inconsistent tolerant reasoning methods such as ASP [Ostrowski and Schaub, 2012] or dedicated tools [Bourgaux, 2016].

Please note that this does not mean that logic-based argumentation does not have an added value. For instance, its explanatory power might benefit to increase the scrutinability of the system by human users [Arioua et al., 2016; Besnard et al., 2014] and the use of ranking semantics can induce a stratification of the inconsistent knowledge base [Amgoud and Ben-Naim, 2015] that might be of use for query answering techniques [Yun et al., 2018g].

The question that naturally arose is whether or not we can find "better" or more efficient argumentation frameworks for *Datalog*±. In the previous chapter, we proposed methods for improving the argument generation by either processing the knowledge base or filtering the arguments and switching to hypergraphs. Following the latter intuition that hypergraphs might be a more efficient and compact form of argumentation frameworks, in this chapter, we first propose a new hypergraph argumentation framework and show how to instantiate it directly and straightforwardly from an inconsistent knowledge base expressed in *Datalog*±. Second, we show that this new hypergraph argumentation framework benefits from the same desirable properties (rationality postulates, structural properties, etc.) as its binary argumentation framework counterpart. Lastly, we show an implementation of the NAKED tool for generating, visualising and exporting argumentation hypergraphs from knowledge bases expressed using *Datalog*±. Then, using the aforementioned tool, we compare this new hypergraph argumentation framework with the previous binary framework and show how we improve upon the state-of-the-art.

The chapter is organized as follows: In Section 4.1, we introduce the hypergraph argumentation framework, show the various properties that it satisfies and compare it to the previous argumentation framework for *Datalog*±. In Section 4.2, we show the first implementation of a generator for our hypergraph argumentation framework.

Research Questions in this Chapter

- *How can we instantiate a hypergraph from an inconsistent knowledge base in Datalog*±*? Is this representation efficient?*
- *Can we provide a tool that generates such hypergraphs?*

4.1 Argumentation hypergraphs with the existential rules language

As seen in the previous chapter, the existing argumentation frameworks for the existential rules suffer from the exponential increase of arguments when free facts are added in the fact base. More generally, this problem is stemming from the observation that for a consistent subset of the fact base, nearly all of the subsets of its saturation with rule application will constitute a new argument.

Thus, the intuitions behind this new framework are twofold: (1) The arguments will be built upon other arguments (*à la* ASPIC+). In the essence, this framework is close to rule-based formalisms where arguments are constructed from the ontology. (2) Our framework is different from existing frameworks because of a special feature: *sets of attacking arguments*.

In this section, we thrive to show that this framework retains all of the desirable properties (repair equivalence, rationality postulate, etc.) with less arguments and attacks than the existing frameworks.

There are three main contributions in this section. First, we introduce a logic-based argumentation framework with directed hyperedges for an inconsistent knowledge base expressed using $Datalog\pm$. Second, we provide a theoretical analysis of the new argumentation framework with respect to its syntactic (i.e. graph theoretical) and semantic properties. Namely we show that:

- The rationality postulates (indirect consistency, direct consistency and closure) defined by Caminada and Amgoud [2007] are satisfied by the proposed framework.
- There is a one-to-one correspondence between the repairs of the knowledge base and the preferred (resp. stable) extensions of the new argumentation framework.
- The grounded extension is equal to both the intersection of preferred (resp. stable) extensions and to the set of arguments generated from the intersection of the repairs of the knowledge base.
- The premises of the non attacked arguments are subsumed by the set of free-facts (the intersection of repairs of the knowledge base).
- There are no self-attacking arguments, all the attacked arguments are defended and there is at least one cycle in the argumentation framework if the set of arguments is finite.
- If there are no rules in the knowledge base, the number of arguments is less or equal to the number of facts and the upper-bound to the number of attacks is $\sum_{i=1}^{n-1} \binom{n}{i}(n-i)$ where n is the number of arguments.

Third, we provide a tool for generating this argumentation framework from a *Datalog*± knowledge base expressed in DLGP format and study its performance in terms of argumentation graph compression rate and generation time.

4.1.1 Hypergraph argumentation framework $\mathfrak{F}^\star$

Please note that although the framework described in this section has some similarities with the ASPIC+ framework, we would like to highlight that ASPIC+ cannot be directly instantiated with *Datalog*± because the language does not have the negation and the contrariness function is not general enough for this language. Moreover, when instantiating ASPIC+ with a lot of logical languages (e.g. classical propositional logic), one has to add all the tautologies of the language in the set of strict rules in order to guarantee that the result will be consistent, i.e. to satisfy the rationality postulates defined by Caminada and Amgoud [2007]. In order to avoid adding this enormous number of rules and also with the goal of decreasing the number of arguments, we propose not to add them. However, the cost of forgetting to add those rules (and the arguments generated using them) would result in violation of rationality postulates. We propose to solve this problem in a more elegant way. Namely, we allow for the use of sets of attacking arguments (i.e. directed hyperedges).

In the next definition, we extend Dung's abstract framework with sets of attacking arguments, i.e. sets of arguments can now jointly attack an argument.

Definition 4.1 (Argumentation framework $\mathfrak{F}^\star$). *An hypergraph argumentation framework is a pair $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ with $\mathscr{A}^\star$ a set of arguments and $\mathscr{R}^\star \subseteq (2^{\mathscr{A}^\star} \setminus \emptyset) \times \mathscr{A}^\star$ a set of attacking arguments.*

Notation 4.1. *We denote by $\mathfrak{K}_{\mathfrak{F}^\star}$ the set of all possible hypergraph argumentation frameworks. Let $\mathfrak{F}^\star, \mathfrak{F}^{\star\prime} \in \mathfrak{K}_{\mathfrak{F}^\star}$, we define $\mathfrak{F}^\star \oplus \mathfrak{F}^{\star\prime}$ as the argumentation framework $(\mathscr{A}^\star \cup \mathscr{A}^{\star\prime}, \mathscr{R}^\star \cup \mathscr{R}^{\star\prime})$.*

In the next definition, we show how we build arguments and how to instantiate $\mathfrak{F}^\star$ from an inconsistent knowledge base $\mathcal{KB}$.

Definition 4.2 (Argumentation framework $\mathfrak{F}^\star{}_{\mathcal{KB}}$). *Let us consider the knowledge base $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$. The corresponding argumentation framework denoted by $\mathfrak{F}^\star{}_{\mathcal{KB}} = (\mathscr{A}^\star, \mathscr{R}^\star)$ with $\mathscr{R}^\star \subseteq 2^{\mathscr{A}^\star} \times \mathscr{A}^\star$ is such that:*

- *An argument $a \in \mathscr{A}^\star$ is either:*
  - *f, where $f \in \mathcal{F}$. $Conc(a) = f$ and $Prem(a) = \{f\}$*
  - *$a_1, \ldots, a_n \rightarrow f'$ if $a_1, \ldots, a_n$ are arguments such that there exists a tuple (r, π) where $r \in \mathcal{R}, \pi$ is a homomorphism from the body*

of r to $\{Conc(a_1), \ldots, Conc(a_n)\}$ and f' is the resulting atom from the rule application. $Conc(a) = f'$ and $Prem(a) = Prem(a_1) \cup \cdots \cup Prem(a_n)$

where $Prem(a)$ is $\mathcal{R}$-consistent.

- *An attack is a pair $(X, a) \in \mathscr{R}^\star$ where $X \subseteq \mathscr{A}^\star$ and $a \in \mathscr{A}^\star$ such that X is minimal for set inclusion such that $\bigcup_{x \in X} Prem(x)$ is $\mathcal{R}$-consistent and there exists $\varphi \in Prem(a)$ such that $(\bigcup_{x \in X} Conc(x)) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent.*

Notation 4.2. *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, $X \subseteq \mathcal{F}$ be a set of facts and $X' \subseteq \mathscr{A}^\star$ a set of arguments. We define the set of arguments generated by X as $Arg^\star(X) = \{a \in \mathscr{A}^\star \mid Prem(a) \subseteq X\}$ and the base of a set of arguments X' as $Base^\star(X') = \bigcup_{x' \in X'} Prem(x')$.*

Example 4.1. *Suppose that one is indecisive about what to eat for an appetiser. He decides that the dish should contain salted cucumbers, sugar, yogurt, not be a soup and be edible. However, he finds out that combining together salted cucumbers, sugar and yogurt may not be a good idea. Furthermore, combining salted cucumbers with yogurt is a dish called "tzaziki" which is a famous greek soup. We model the situation with the following knowledge base $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$:*

- $\mathcal{F} = \{contains(m, saltC), contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\}$
- $\mathcal{R} = \{\forall X(contains(X, saltC) \wedge contains(X, yogurt) \rightarrow tzaziki(X))\}$
- $\mathcal{N} = \{\forall X(contains(X, saltC) \wedge contains(X, sugar) \wedge contains(X, yogurt) \rightarrow \bot), \forall X(tzaziki(X) \wedge notSoup(X) \rightarrow \bot)\}$

The resulting argumentation graph $\mathfrak{F}^\star{}_{\mathcal{KB}}$ is composed of eleven attacks and the six following arguments:

- $a_1 = contains(m, sugar)$
- $a_2 = contains(m, saltC)$
- $a_3 = contains(m, yogurt)$
- $a_4 = notSoup(m)$
- $a_5 = edible(m)$
- $a_6 = a_2, a_3 \rightarrow tzaziki(m)$

An example attack of $\mathscr{R}^\star$ is $(\{a_1, a_2\}, a_3)$.

In case of binary attacks (i.e. classical Dung's framework [Dung, 1995]), a set of arguments X is said to attack an argument a if and only if there exists $b \in X$ such that b attacks a. We need a similar notion here except that we already have a notion of attack from a set towards an argument. In order not to mix up the two notions, we introduce the notation $\mathscr{R}^\circ$, which stands for the saturated set of attacks. For example, if $(\{a, b\}, c) \in \mathscr{R}^\star$ then each set X' containing a and b (i.e. such that $\{a, b\} \subseteq X'$) attacks c too (i.e. $(X', c) \in \mathscr{R}^\circ$).

Definition 4.3 (Saturated set of attacks). *Let $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ be an argumentation framework. We define $\mathscr{R}^\circ \subseteq 2^{\mathscr{A}^\star} \times \mathscr{A}^\star$ the saturated set of attacks as $\mathscr{R}^\circ = \{(X, a)$ such that there exists $(X', a) \in \mathscr{R}^\star$ with $X' \subseteq X\}$.*

4.1.2 Argumentation framework properties

The propose argumentation framework $\mathfrak{F}^\star$ is an instantiation of the abstract hypergraph framework proposed by Nielsen and Parsons [2006, 2007]. For the purpose of this thesis being self-contained, we briefly recall the necessary definitions.

Definition 4.4 (Argumentation semantics). *Let $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ be an argumentation framework and $\mathscr{R}^\circ$ be the corresponding saturated set of attacks:*

- *A set of arguments S is **conflict-free** if and only if there is no argument $a \in S$ such that $(S, a) \in \mathscr{R}^\circ$.*
- *A set of arguments S_1 **attacks** a set of arguments S_2 if and only if there exists $a \in S_2$ such that $(S_1, a) \in \mathscr{R}^\circ$. By abuse of notation, we will use the notation $(S_1, S_2) \in \mathscr{R}^\circ$ for the case when S_1 attacks a set of arguments S_2.*
- *A set of arguments S_1 **defends** an argument a if and only if for every set of arguments S_2 such that $(S_2, a) \in \mathscr{R}^\star$, we have that $(S_1, S_2) \in \mathscr{R}^\circ$.*

- *A conflict-free set of arguments S is said to be **admissible** if each argument in S is defended by S.*
- *An admissible set S is called a **preferred extension** if there is no admissible set $S' \subseteq \mathscr{A}^\star$, $S \subset S'$.*
- *A conflict-free set S is a **stable extension** if S attacks all arguments in $\mathscr{A}^\star \setminus S$.*
- *An admissible set S is called a **grounded extension** if S is minimum for set inclusion such that it contains every argument defended by S.*

The set of all preferred (resp. stable and grounded) extensions of an argumentation framework $\mathfrak{F}^\star$ is denoted by $Ext_p(\mathfrak{F}^\star)$ (resp. $Ext_s(\mathfrak{F}^\star)$ and $Ext_g(\mathfrak{F}^\star)$).

Example 4.2 (Example 4.1 cont'd). *The preferred (resp. stable) extensions of $Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}})$ (resp. $Ext_s(\mathfrak{F}^\star{}_{\mathcal{KB}})$) are:*

- $E_1 = \{a_2, a_3, a_5, a_6\}$
- $E_2 = \{a_1, a_2, a_4, a_5\}$
- $E_3 = \{a_1, a_3, a_4, a_5\}$

The grounded extension is $E_{GE} = \{a_5\}$

In the next proposition, we show that there is one-to-one correspondence between the set of preferred (resp. stable) extensions and the set of repairs.

Proposition 4.1 (Preferred and stable characterisation). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, $\mathfrak{F}^\star{}_{\mathcal{KB}}$ be the corresponding argumentation framework and $x \in \{s, p\}$. Then:*

$$Ext_x(\mathfrak{F}^\star{}_{\mathcal{KB}}) = \{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\}$$

Proof sketch. The sketch of the proof is as follows:

1. We prove that $\{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\} \subseteq Ext_s(\mathfrak{F}^\star{}_{\mathcal{KB}})$.
2. We prove that $Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}}) \subseteq \{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\}$.
3. Since every stable extension is a preferred one [Nielsen and Parsons, 2007], we can proceed as follows. From the first item, we have that $\{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\} \subseteq Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}})$, thus the theorem holds for preferred semantics. From the second item, we have $Ext_s(\mathfrak{F}^\star{}_{\mathcal{KB}}) \subseteq \{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\}$, thus the theorem holds for stable semantics.

For a detailed proof, please refer to Section 7.2.2 on page viii. □

Example 4.3 (Example 4.2 cont'd). *As explained in Proposition 4.1, we have a one to one correspondence between repairs and preferred (resp. stable) extensions. Hence, we have:*

- $E_1 = Arg^\star(\{contains(m, saltC), contains(m, yogurt), edible(m)\})$
- $E_2 = Arg^\star(\{contains(m, sugar), contains(m, saltC), notSoup(m), edible(m)\})$
- $E_3 = Arg^\star(\{contains(m, sugar), contains(m, yogurt), notSoup(m), edible(m)\})$

We now show that there is an equivalence between the non-attacked arguments and the arguments generated from free-facts.

Corollary 4.1 (Non-attacked characterisation). *Let $\mathcal{KB}$ be a knowledge base, $\mathfrak{F}^{\star}{}_{\mathcal{KB}} = (\mathscr{A}^{\star}, \mathscr{R}^{\star})$ be the corresponding argumentation framework and $a \in \mathscr{A}^{\star}$. There is no $S \subseteq \mathscr{A}^{\star}$ such that $(S, a) \in \mathscr{R}^{\star}$ if and only if $Prem(a) \subseteq \bigcap repairs(\mathcal{KB})$.*

Proof. We split this proof in two parts:

- ($\Rightarrow$) By definition, if $a \in \mathscr{A}^{\star}$ is such that there is no $(S, a) \in \mathscr{R}^{\star}$ then it means that a belongs to every preferred (resp. stable) extensions. Using the result of Proposition 4.1, we deduce that $Prem(a) \subseteq \bigcap repairs(\mathcal{KB})$.

- ($\Leftarrow$) Suppose that $Prem(a) \subseteq \bigcap repairs(\mathcal{KB})$. It means that for every $\mathcal{R}$-consistent subset $X \subseteq \mathcal{F}$, we have that $Prem(a) \cup X$ is $\mathcal{R}$-consistent. Suppose that there exists $S \subseteq \mathscr{A}^{\star}$ such that $(S, a) \in \mathscr{R}^{\star}$. It means that there exists $\varphi \in Prem(a)$ such that $(\bigcup_{s \in S} Conc(s)) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent. Thus, $(\bigcup_{s \in S} Prem(s)) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent and $(\bigcup_{s \in S} Prem(s)) \cup Prem(a)$ is $\mathcal{R}$-inconsistent, contradiction.

□

Please note that although it is tempting to say that the non-attacked arguments do not contribute to attacks because the are based on free-facts, we show in the next example that this is not true in the general case.

Example 4.4. *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be the knowledge base such that:*

- $\mathcal{F} = \{d(m), a(m), c(m)\}$
- $\mathcal{R} = \{\forall X(d(X) \rightarrow e(X)), \forall X(a(X) \rightarrow b(X))\}$
- $\mathcal{N} = \{\forall X(a(X) \wedge c(X) \rightarrow \bot), \forall X(e(X), b(X) \wedge c(X) \rightarrow \bot)\}$

The knowledge base $\mathcal{KB}$ has two repairs, i.e. $repairs(\mathcal{KB}) = \{\{a(m), d(m)\}, \{d(m), c(m)\}\}$. The fact $d(m)$ is a free-fact. There are five arguments in the argumentation framework $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$:

- $a_0 = d(m)$
- $a_1 = a_0 \rightarrow e(m)$
- $a_2 = c(m)$
- $a_3 = a(m)$
- $a_4 = a_3 \rightarrow b(m)$

We can see that the non-attacked argument a_1 is contributing to an attack to a_2 with a_4, i.e. $(\{a_1, a_4\}, a_2) \in \mathscr{R}^\star$.

In the next proposition, we show that, in our argumentation framework, the grounded extension is equal to the intersection of the preferred extensions. Note that although it is always true that the set grounded extension is included the intersection of the preferred extensions, the equality is not true in the general case.

Proposition 4.2 (Grounded and preferred). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, $\mathfrak{F}^\star{}_{\mathcal{KB}}$ be the corresponding argumentation framework and E_{GE} be the grounded extension of $\mathfrak{F}^\star{}_{\mathcal{KB}}$. Then:*

$$E_{GE} = \bigcap Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}})$$

Proof sketch. Nielsen and Parsons [2007] showed that $E_{GE} \subseteq \bigcap Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}})$. We now show that $\bigcap Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}}) \subseteq E_{GE}$ Let use consider $a \in \bigcap Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}})$. Using Proposition 4.1, we have that $a \in \bigcap_{A' \in repairs(\mathcal{KB})} Arg^\star(A')$. Thus, $a \in Arg^\star(\bigcap repairs(\mathcal{KB}))$. We conclude that $Prem(a) \subseteq \bigcap repairs(\mathcal{KB})$, then with Corollary 4.1, we conclude that a is not attacked and $a \in E_{GE}$. □

Example 4.5 (Example 4.2 cont'd). *We have that the grounded extension $E_{GE} = E_1 \cap E_2 \cap E_3 = \{a_5\}$.*

We now show that the grounded extension is equal to the set of arguments generated by the intersection of all the repairs.

Proposition 4.3 (Grounded characterisation). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, $\mathfrak{F}^\star{}_{\mathcal{KB}}$ be the corresponding argumentation framework and E_{GE} be the grounded extension of $\mathfrak{F}^\star{}_{\mathcal{KB}}$. Then:*

$$E_{GE} = Arg^\star(\bigcap repairs(\mathcal{KB}))$$

Proof sketch. The proof relies on three results: (1) The equivalence between the preferred extensions and the repairs as proven in Proposition 4.1; (2) The fact that the grounded extension is equal to the intersection of all preferred extensions (see Proposition 4.2) and (3) the fact that for every collection of set of formulae $S_1, \ldots, S_n$, $Arg^\star(S_1 \cap \cdots \cap S_n) = Arg^\star(S_1) \cap \cdots \cap Arg^\star(S_n)$. □

Example 4.6 (Example 4.2 cont'd). *We have that the grounded extension is $E_{GE} = \{a_5\} = Arg^\star(\{edible(m)\})$.*

We now show that the argumentation framework $\mathfrak{F}^\star{}_{\mathcal{KB}}$ does not have self-attacking arguments.

Proposition 4.4 (Self-attacking arguments). *Let $\mathcal{KB}$ be a knowledge base and $\mathfrak{F}^\star{}_{\mathcal{KB}} = (\mathscr{A}^\star, \mathscr{R}^\star)$ be the corresponding argumentation framework. There is no $(S, t) \in \mathscr{R}^\star$ such that $t \in S$.*

Proof. Suppose that there is $(S, t) \in \mathscr{R}^\star$ such that $t \in S$. It means that there exists $\varphi \in Prem(t)$ such that $(\bigcup_{s \in S} Conc(s)) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent. It means that $(\bigcup_{s \in S} Prem(s)) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent and thus $\bigcup_{s \in S} Prem(s)$ is $\mathcal{R}$-inconsistent since $t \in S$, contradiction with the definition of an attack. □

In Proposition 4.5 below, we show that an attacked argument is always defended by a set of arguments.

Proposition 4.5 (Defense). *Let $\mathcal{KB}$ be a knowledge base and $\mathfrak{F}^\star{}_{\mathcal{KB}} = (\mathscr{A}^\star, \mathscr{R}^\star)$ be the corresponding argumentation framework. If there is $(S, t) \in \mathscr{R}^\star$ then there exists $(S', s) \in \mathscr{R}^\star$ such that $s \in S$.*

Proof. Let us prove it by contradiction. Suppose that $(S, t) \in \mathscr{R}^\star$ and that there is no $S' \in \mathscr{A}^\star$ such that $(S', s) \in \mathscr{R}^\star$, where $s \in S$. Namely, it means that there exists $\varphi \in Prem(t)$ such that $(\bigcup_{s \in S} Conc(s)) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent. Thus, $(\bigcup_{s \in S} Prem(s)) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent. Let $X = \{x_1, \ldots, x_n\}$ be a subset of $(\bigcup_{s \in S} Prem(s))$ such that $X \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent and for all $X' \subset X$, $X' \cup \{\varphi\}$ is $\mathcal{R}$-consistent. Then, let us denote by s_1, the argument in S such that $x_1 \in Prem(s_1)$, we have that $(\{a_2, \ldots, a_n, a_\varphi\}, s_1) \in \mathscr{R}^\star$ where $a_i = x_i$ for $i \in \{2, \ldots, n\}$ and $a_\varphi = \varphi$, contradiction. □

We now introduce the definition of cycle for an argumentation framework with sets of attacking arguments.

Definition 4.5 (Cycle). *A cycle in $\mathfrak{F}^\star = (\mathfrak{F}^\star, \mathscr{R}^\star)$ is a sequence of attacks in $\mathscr{R}^\star$ of the form $((S_1, t_1), \ldots, (S_n, t_n))$ such that for every $i \in \{1, \ldots, n-1\}, t_i \in S_{i+1}$ and $t_n \in S_1$.*

The following Corollary 4.2 follows directly from Proposition 4.5. If the number of arguments is finite then there exists at least one cycle.

Corollary 4.2 (Cycle existence). *Let $\mathcal{KB}$ be a knowledge base and $\mathfrak{F}^\star{}_{\mathcal{KB}} = (\mathscr{A}^\star, \mathscr{R}^\star)$ be the corresponding argumentation framework. If $\mathscr{A}^\star$ has a finite number of arguments and $\mathscr{R}^\star \neq \emptyset$ then there exists a cycle in $\mathfrak{F}^\star{}_{\mathcal{KB}}$.*

Example 4.7 (Example 4.1 cont'd). *The following sequence of attacks $((\{a_2, a_3\}, a_1), (\{a_1, a_3\}, a_2))$ is a cycle in $\mathfrak{F}^\star{}_{\mathcal{KB}}$.*

Contrary to the argumentation framework $\mathfrak{F}$ where the number of arguments can be exponential even in the case where the set of rules is empty, we show that for the framework $\mathfrak{F}^\star$ described in Definition 4.2, the set of arguments is at most equal to the number of facts.

Proposition 4.6 (Argument upper-bound). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that $\mathcal{R} = \emptyset$ and $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ be the corresponding argumentation framework. It holds that $|\mathscr{A}^\star| \leq |\mathcal{F}|$.*

Proof. Let $a \in \mathscr{A}^\star$ be an argument of $\mathfrak{F}^\star$. Since there are no rules then $a = f$, where $f \in \mathcal{F}$. Thus we have that $|\mathscr{A}^\star| \leq |\mathcal{F}|$. Note that the equality holds if and only if there are no facts f such that $\{f\}$ is inconsistent. □

In the next proposition, we show an upper bound to the number of attacks with respect to the number of arguments.

Proposition 4.7 (Attack upper-bound). *Let $\mathcal{KB}$ be a knowledge base and $\mathfrak{F}^\star{}_{\mathcal{KB}} = (\mathscr{A}^\star, \mathscr{R}^\star)$ be the corresponding argumentation framework. If $|\mathscr{A}^\star| = n$ then $|\mathscr{R}^\star| \leq \sum_{i=1}^{n-1} \binom{n}{i}(n-i)$.*

Proof sketch. We prove the proposition by induction (for a detailed proof, see Section 7.2.2 on page ix). □

Please note that this attack upper-bound is generally never reached in large instances because of the minimality condition on the set of attacking arguments.

4.1.3 Rationality postulates

In this section, we prove that the framework we propose in this thesis satisfies the rationality postulates for instantiated argumentation frameworks. We first prove the indirect consistency postulate.

Notation 4.3. *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and $\mathfrak{F}^\star{}_{\mathcal{KB}} = (\mathscr{A}^\star, \mathscr{R}^\star)$ be the corresponding argumentation framework. If $X \subseteq \mathscr{A}^\star$ is a set of arguments, then we define $Concs(X) = \bigcup_{x \in X} Conc(x)$ and $Output_x(\mathfrak{F}^\star{}_{\mathcal{KB}}) = \bigcap_{E \in Ext_x(\mathfrak{F}^\star{}_{\mathcal{KB}})} Concs(E)$ where $x \in \{s, p, g\}$.*

Proposition 4.8 ($\mathfrak{F}^\star$ Indirect consistency). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, $\mathfrak{F}^\star{}_{\mathcal{KB}}$ be the corresponding argumentation framework and $x \in \{s, p, g\}$. Then:*

- *for every $E \in Ext_x(\mathfrak{F}^\star{}_{\mathcal{KB}}), Concs(E)$ is a $\mathcal{R}$-consistent.*
- *$Output_x(\mathfrak{F}^\star{}_{\mathcal{KB}})$ is $\mathcal{R}$-consistent.*

For the complete proof of Proposition 4.8, see Section 7.2.2 on page x.

Since our instantiation satisfies indirect consistency then it also satisfies direct consistency. Indeed, if a set is $\mathcal{R}$-consistent, then it is consistent. Thus, we obtain the following corollary.

Corollary 4.3 ($\mathfrak{F}^\star$ Direct consistency). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, ${\mathfrak{F}^\star}_{\mathcal{KB}}$ be the corresponding argumentation framework and $x \in \{s, p, g\}$. Then:*

- *for every $E \in Ext_x({\mathfrak{F}^\star}_{\mathcal{KB}}), Concs(E) \not\models \bot$.*
- $Output_x({\mathfrak{F}^\star}_{\mathcal{KB}}) \not\models \bot$.

We now show that the argumentation framework satisfies the closure postulate.

Proposition 4.9 ($\mathfrak{F}^\star$ Closure). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, ${\mathfrak{F}^\star}_{\mathcal{KB}}$ be the corresponding argumentation framework and $x \in \{s, p, g\}$. Then:*

- *for every $E \in Ext_x({\mathfrak{F}^\star}_{\mathcal{KB}}), Concs(E) = Sat_{\mathcal{R}}(Concs(E))$*
- $Output_x({\mathfrak{F}^\star}_{\mathcal{KB}}) = Sat_{\mathcal{R}}(Output_x({\mathfrak{F}^\star}_{\mathcal{KB}}))$

For the complete proof, see Section 7.2.2 on page xi.

4.1.4 Empirical analysis

In this section, we compare our approach with the existing argumentation frameworks for *Datalog*± with respect to the number of arguments and the number of attacks.

We chose to work with the set of knowledge bases extracted from the study of Yun et al. [2017b, 2018f] (see Table 3.5 on page 80). We recall that these inconsistent knowledge bases are composed of two main sets:

- A set A composed of 108 knowledge bases. This dataset is further split into three smaller set of knowledge bases:
  - A set of A_1 of 31 knowledge bases without rules, two to seven facts, and one to three negative constraints.
  - A set A_2 of 51 knowledge bases generated by fixing the size of the set of facts and successively adding negative constraints until saturation.
  - A set A_3 of 26 knowledge bases with only ternary negative constraints, three to four facts and one to three rules.
- A set B composed of 26 knowledge bases with eight facts, six rules and one or two negative constraints. This set contains more free-facts than the knowledge bases in set A.

Deductive argumentation framework $\mathfrak{F}_{\mathcal{KB}}$			
$\mathcal{KB}$	# Arg.	# Att.	Gen. Time (ms)
A_1	22	128	160
A_2	25	283	133
A_3	85	1472	399,5
B	5967	11542272	533089

Hypergraph argumentation framework $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$						
$\mathcal{KB}$	#Arg.	%Arg. ↓	#Att.	%Att. ↓	Gen. Time (ms)	%Time ↓
A_1	5	77,27	6	93,75	276,00	-81,48
A_2	7	72,00	8	92,93	342,00	-183,57
A_3	7	91,76	9	99,26	369,50	1,66
B	14	99.77	20.5	99.99	7814.5	98.08

Table 4.1: Comparison of the median number of arguments, attacks and generation time needed between $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ and $\mathfrak{F}_{\mathcal{KB}}$ on the sets of knowledge bases A_1, A_2, A_3 and B

For each knowledge base $\mathcal{KB}$ in these two sets, we compare the number of generated arguments and attacks of the new framework $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ with the framework $\mathfrak{F}_{\mathcal{KB}}$.

We provide a the tool based on the Graph of Atom Dependency defined by Hecham et al. [2017a] and the Graal Java Toolkit [Baget et al., 2015c] for generating the new argumentation framework from an inconsistent knowledge base expressed in the DLGP format. The tool is available for download at the following address: `https://www.dropbox.com/sh/dlpmr07gqvpuc61/AABDgwfHJRNVYcsqpDg7kMfEa?dl=0`

4.1.4.1 Experimental results

In Table 4.1, we show the number of arguments and attacks of the two frameworks for the two sets A and B. We make the following observations:

- The generation of $\mathfrak{F}^{\star}$ is slower than the one for $\mathfrak{F}$ when the number of arguments and attacks is relatively low (see A_1 and A_2) but when the number of arguments and attacks increases, we can see that the generation of $\mathfrak{F}^{\star}$ is much faster (see Figure 4.1 and 4.2). The reason for this is simple. Contrary to the framework $\mathfrak{F}$, there is no exponential increase of the number of arguments with the number of free-facts in $\mathfrak{F}^{\star}$ as seen with the knowledge bases in set B. Moreover, for all the knowledge bases considered in sets A and B, the number of arguments

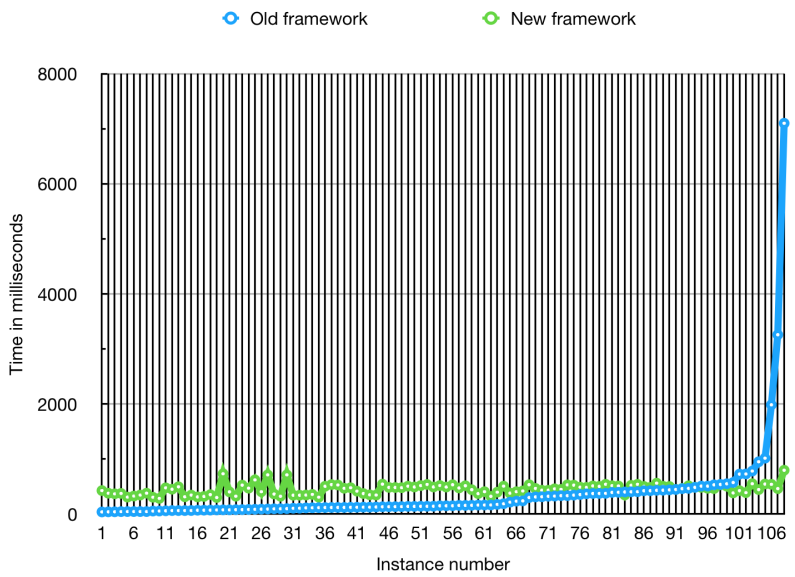

Figure 4.1: Generation time for both the old framework $\mathfrak{F}_{\mathcal{KB}}$ and the new framework $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ with the set of knowledge bases A. Instances are sorted with respect to the generation time of $\mathfrak{F}_{\mathcal{KB}}$

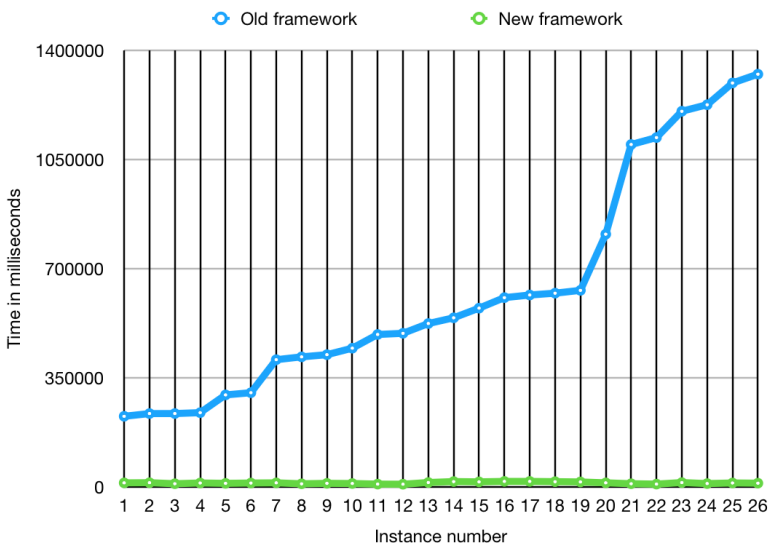

Figure 4.2: Generation time for both the old framework $\mathfrak{F}_{\mathcal{KB}}$ and the new framework $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ with the set of knowledge bases B. Instances are sorted with respect to the generation time of $\mathfrak{F}_{\mathcal{KB}}$

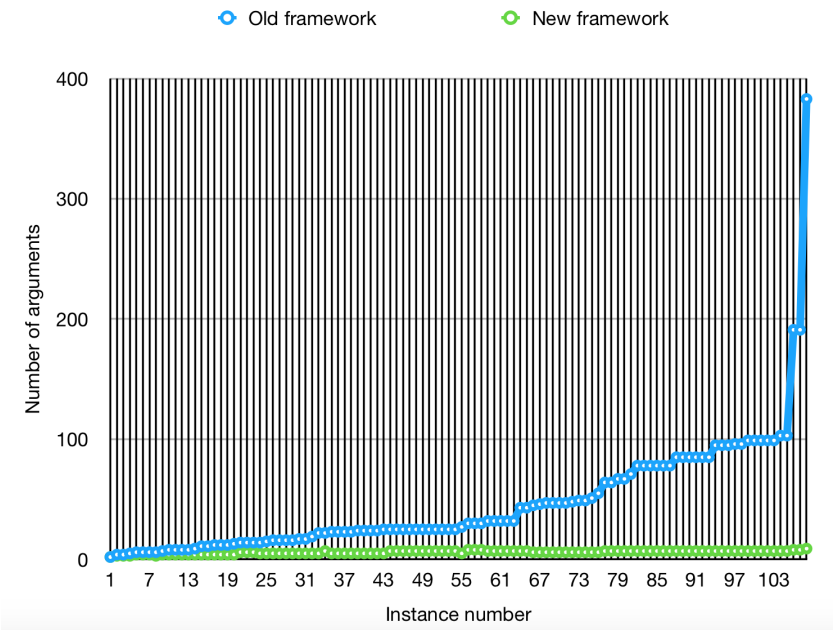

Figure 4.3: Number of arguments for both the old framework $\mathfrak{F}_{\mathcal{KB}}$ and the new framework $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ with the set of knowledge bases A. Instances are sorted with respect to the number of arguments of $\mathfrak{F}_{\mathcal{KB}}$

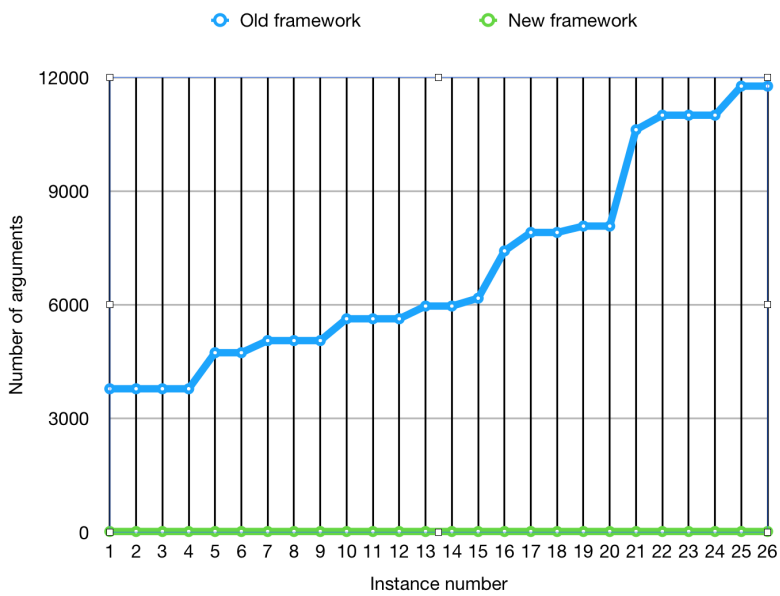

Figure 4.4: Number of arguments for both the old framework $\mathfrak{F}_{\mathcal{KB}}$ and the new framework $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ with the set of knowledge bases B. Instances are sorted with respect to the number of arguments of $\mathfrak{F}_{\mathcal{KB}}$

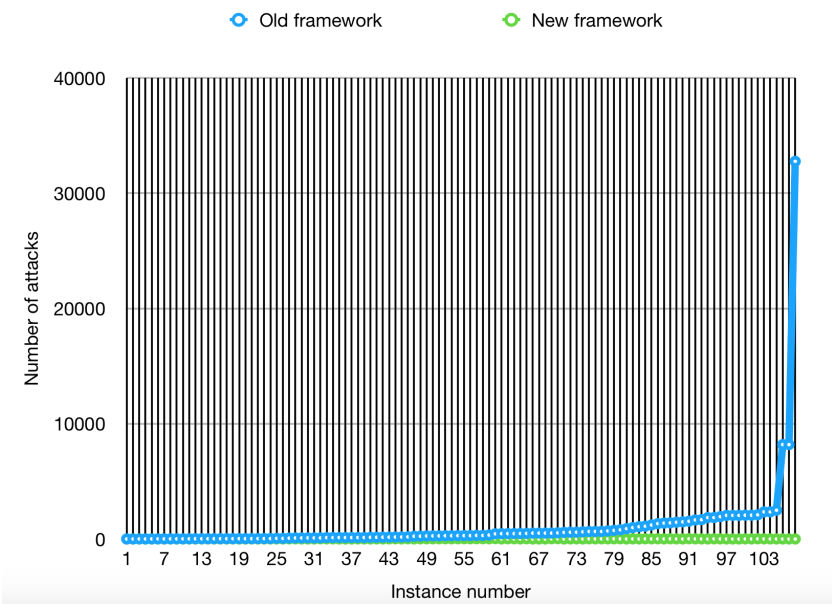

Figure 4.5: Number of attacks for both the old framework $\mathfrak{F}_{\mathcal{KB}}$ and the new framework $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ with the set of knowledge bases A. Instances are sorted with respect to the number of attacks of $\mathfrak{F}_{\mathcal{KB}}$

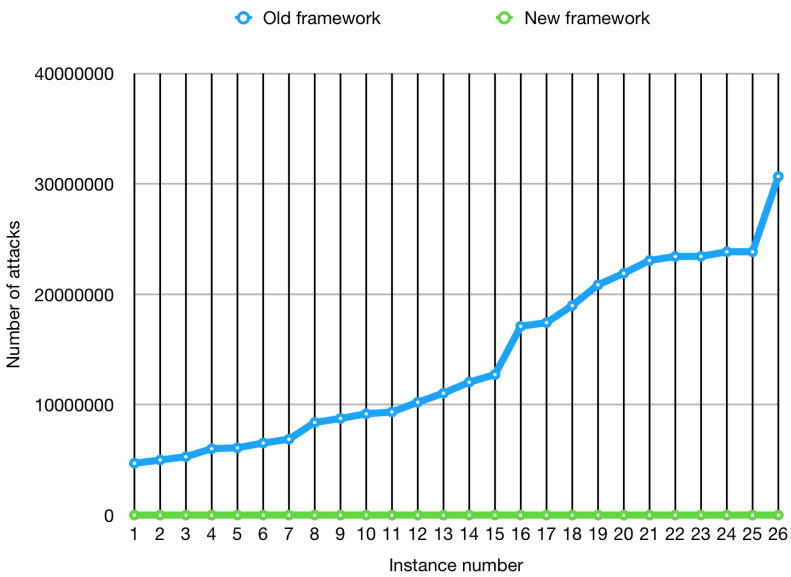

Figure 4.6: Number of attacks for both the old framework $\mathfrak{F}_{\mathcal{KB}}$ and the new framework $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ with the set of knowledge bases B. Instances are sorted with respect to the number of attacks of $\mathfrak{F}_{\mathcal{KB}}$

and attacks in $\mathfrak{F}^\star$ is less or equal to the number of arguments and attacks in $\mathfrak{F}$ (see also Figure 4.3, 4.4 and 4.5 and 4.6).

- When only the set of negative constraints is varying, the number of arguments of $\mathfrak{F}^\star$ seems to be unchanged whereas in $\mathfrak{F}$, it is varying. Furthermore, the framework $\mathfrak{F}$ seems much denser than $\mathfrak{F}^\star$. Indeed, the median density[1] of $\mathfrak{F}$ is 26.34% and 31,03% whereas the median density of $\mathfrak{F}^\star$ is 4,69% and 0.02% on the sets A and B respectively.

The entire experiment was conducted on a Debian computer with an Intel Xeon E5-1620 (3.60GHz) processor and 64GBs of RAM.

[1]The density is equal to the number of attacks divided by the maximum number of possible attacks. In the case of a directed graph, the maximum number of attacks is given by $n(n-1)$ where n is the number of nodes. In the case of $\mathfrak{F}^\star$, we use the formula in Proposition 4.7 to obtain the maximum number of attacks.

4.2 The NAKED tool

In this section, we present NAKED, a tool for generating and visualising hypergraph argumentation framework that uses knowledge bases expressed in *Datalog*± [Calì et al., 2009]. In the NAKED tool, we instantiate the framework of Nielsen and Parsons [2007] which allows us to avoid the explosion of the number of arguments while keeping all the desirable properties of the previous argumentation framework (see the previous Section 4.1).

Classically, reasoning with argumentation graphs consists of either finding extensions or ranking arguments from the most to the least acceptable. As a result, most of the past work has been focused, amongst others, on optimising the efficiency of the extension finding procedures [Gaggl, 2013; Lagniez et al., 2015], on the investigation of various extension-based and ranking-based notions [Baroni et al., 2011; Bonzon et al., 2016; Amgoud and Ben-Naim, 2015] and on the investigation of desirable properties of logic based instantiations [Amgoud, 2014; Modgil and Prakken, 2014].

There are few practical tools that allow to generate an argumentation framework from a given knowledge base [Thimm, 2017; Yun et al., 2018c]. Furthermore, the few available tools for reasoning using argumentation over inconsistent logical knowledge bases often suffer from on of the following problems: (1) they do not allow further tool interoperability (allowing their output argumentation graph to be loaded in other tools), (2) they do not scale up for a practical use or (3) they only consider binary argumentation frameworks. Our workflow enables any data engineer to:

1. input a knowledge base in the well-known DLGP format for *Datalog*± and generate an argumentation hypergraph that instantiates the framework of Nielsen and Parsons [2007] and outputs a visualisation (see Figure 4.7 for an example of hypergraph visualisation).

2. interact with the graph representation by allowing manual arguments re-positioning. Moreover, the end-user has the possibility to observe a specific argument by highlighting the corresponding argument and its attackers in different colours.

3. exporting the generated argumentation hypergraph in the DOT format for a better tool interoperability.

All of these functions are useful for a non computer science expert who wants to reason over an inconsistent knowledge base in a particular domain using argumentation [Arioua et al., 2016; Tamani et al., 2014a,b]. NAKED could also be useful for investigating the theoretical properties of the graph-based representation of the generated argumentation framework in the same fashion as Yun et al. [2017b] and Arioua et al. [2017]. Given the fact that certain graph theoretical properties could radically improve the extension

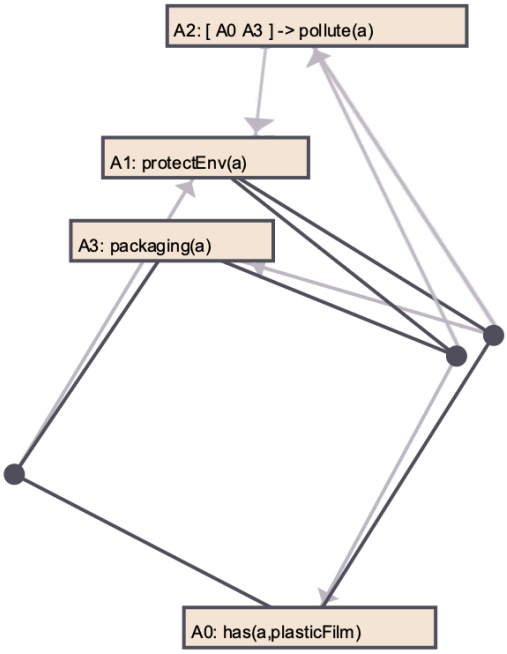

Figure 4.7: An argumentation hypergraph about packagings.

computation efficiency [Yun et al., 2017b] such visualisation is a useful tool for argumentation specialists. A presentation video explaining all of the features of NAKED is available online at: `https://youtu.be/q54iNWBZ9dY`. NAKED is meant to assist domain experts and argumentation developers in the specification, visualisation and/or export of logic based argumentation frameworks built over the *Datalog*± language.

This section is structured as follows. In Section 4.2.1, we describe the different steps and mechanisms of the arguments and attacks generation in NAKED. In Section 4.2.2, we describe the main visual areas of the NAKED tool and how argumentation hypergraph are represented. In Section 4.2.3, we show two real-life scenarios of the usage of NAKED.

4.2.1 The argument and attack generation

In this section, we show how NAKED allows for the generation of arguments and attacks. As shown in Section 4.1, the arguments are built upon each other based on how rules can be applied. In a sense, the arguments can be seen as pieces of rule derivations. Thus, in order to generate all of the arguments that conclude a given fact, we need to extract all the possible derivations for this specific fact. Without a sound a complete derivation extraction mechanism, we will not be able to generate all the arguments. The most common chases such as the frontier/skolem chase or the oblivious chase all possess a derivation reducer in order for the chase to stop. However,

those derivation reducers might induce a loss of rule applications depending on the order in which the rules are applied. Although this loss of derivations has no effect for the classical entailment, it has critical consequences for the generation of arguments. In order to fix this derivation loss problem, Hecham et al. [2017b] provides the Graph of Atom Dependency (GAD) which is constructed alongside a chase. This GAD is constructed differently depending on the choice of the chase [Hecham et al., 2017a]. Moreover, the GAD allows for the extraction of all the derivations for any given fact. Against this background, the generation of the GAD is necessary for the generation of arguments. Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be an arbitrary inconsistent knowledge base, we show the four step process that leads to the generation of the argumentation framework $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$:

1. First, the graph of atom dependency (GAD) of the knowledge base $\mathcal{KB}$ is constructed alongside the chase. Here, the GAD is a specific edge-labeled directed hypergraph where nodes are atoms of $Sat_{\mathcal{R}\cup\mathcal{N}}(\mathcal{F})$ and the hyperedges represent the different rule applications (see [Hecham et al., 2017a] for more details).

2. Second, we recursively build the arguments in $\mathscr{A}^{\star}$ by parsing the derivations of the GAD of $\mathcal{KB}$ and filter arguments with inconsistent premises (see Definition 4.2 on page 93).

3. Third, the set of repairs of $\mathcal{KB}$ are computed. Although this step is arduous, one can find more efficient approaches than the naive method. We compute the minimal inconsistent sets of $\mathcal{KB}$ using the GAD and use efficient algorithms from graph theory to calculate the repairs from the minimal inconsistent sets.

4. Fourth, for each repair R, we compute the set of arguments $Arg^{\star}(R)$. As we know that there are no attacks from a subset of $Arg^{\star}(R)$ to an argument of $Arg^{\star}(R)$, we only compute attacks from a subset of $Arg^{\star}(R)$ to an argument of $\mathscr{A}^{\star} \setminus Arg^{\star}(R)$.

4.2.2 The structure of NAKED

In this section, we introduce the three main areas of the NAKED interface (see Figure 4.9):

- **Area 1 (the blue area):** In this area, the user can input an inconsistent knowledge base $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ in the DLGP format [Baget et al., 2015b]. This format can accommodate existential rules (and other constructs that can be seen as special kinds of existential rules: facts, negative constraints and conjunctive queries). Once the knowledge base is specified, the user can launch the graph generation by pressing the "Compute" button.

- **Area 2 (the red area):** In this area, the argumentation hypergraph $\mathfrak{F}^{\star}{}_{\mathcal{KB}} = (\mathscr{A}^{\star}, \mathscr{R}^{\star})$ is automatically displayed and laid out. The square nodes represent arguments in $\mathscr{A}^{\star}$ whereas round black nodes represent the hyperedges in $\mathscr{R}^{\star}$. In our visualisation, we decided to group attacks that originate from the same set of attacking arguments, i.e. each round black node represents a maximal group of hyperedges in $\mathscr{R}^{\star}$ that have the same set of attacking arguments S. Thus, for each round black node, there is a unique set of attacking arguments S and each argument in S is linked to the round black node with an undirected black edge. Each of the grey directed edge from a round black node to an argument node a represents the attack $(S, a) \in \mathscr{R}^{\star}$. This is explained in Example 4.8.

  Example 4.8 (Hyperedge representation). *In Figure 4.8, the set of attacking arguments associated with the round black node is* $S = \{A_1, A_0\}$. *There are two outgoing grey edges form the round black node to* A_2 *and* A_3*, thus the two corresponding attacks in* $\mathscr{R}^{\star}$ *are* (S, A_2) *and* (S, A_0).

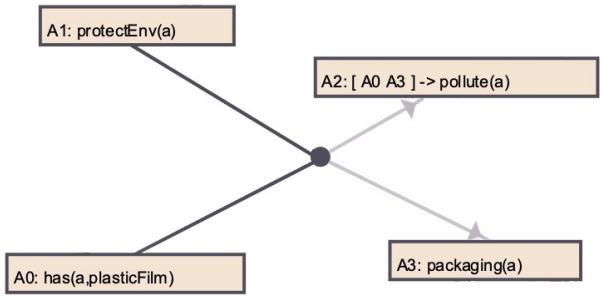

Figure 4.8: Representation of hyperedges in NAKED

  Moreover, each node can be individually moved and repositioned by doing a *drag and drop* gesture.

- **Area 3 (the green area):** This area is used for textually observing the graph and exporting it to other formats. There are five tabs:

  – In the "Arguments & Attacks" tab, the user can retrieve the list of arguments and attacks as well as their meaning.

  – In the "Observer" tab, the user can highlight a particular argument and its attackers as well as retrieve the list of the argument's attackers (see Figure 4.10)

  – In the "Repairs" tab, the user can find the list of maximal consistent set of facts (repairs) of the knowledge base.

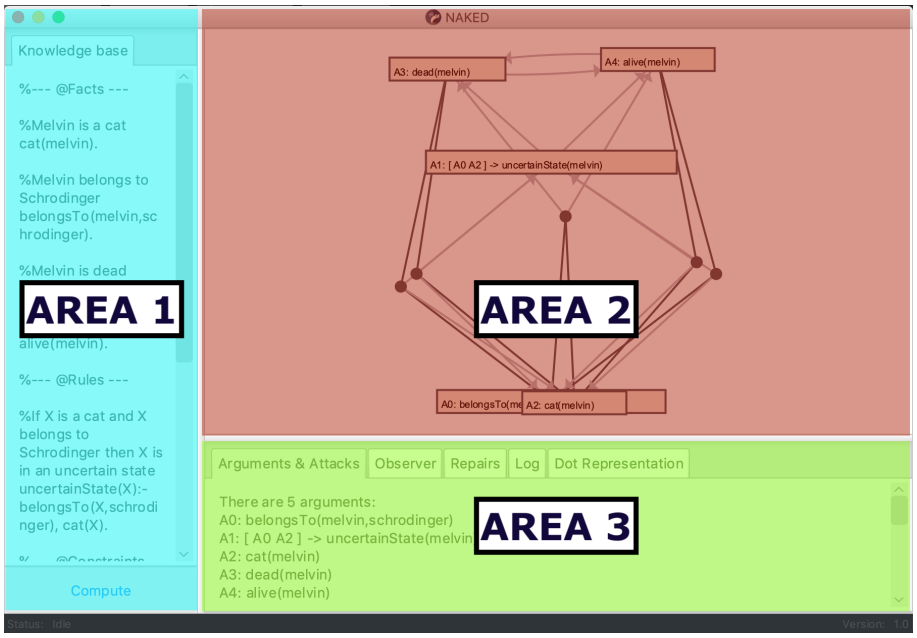

Figure 4.9: Representation of the areas of interest in NAKED

- In the "Log" tab, the user can find the history of the interactions with the system, errors, feedbacks, etc.
- In the "Dot Representation" tab, the user find the graph in the DOT format.

4.2.3 Usability scenarios

We consider two usability scenarios of NAKED.

Scenario 1 First, we consider the task of a non computer science specialist inputting an inconsistent knowledge base of his or her expertise and wanting to find the maximally consistent point of views one can consider with respect to a given knowledge base. Please note that tools for assisting non domain experts in building such knowledge bases without computer expertise exists [Chein and Mugnier, 2009]. Finding maximally consistent point of views (or repairs) consists in computing all maximal subsets of facts that do not trigger any negative constraints of the knowledge base. NAKED will not only be able to provide the list of repairs but also highlight why a given fact does not belong to a specific repair using the *observer tab*.

Scenario 2 Second, we consider the task of an argumentation expert that wants to generate argumentation hypergraphs for benchmarking purposing.

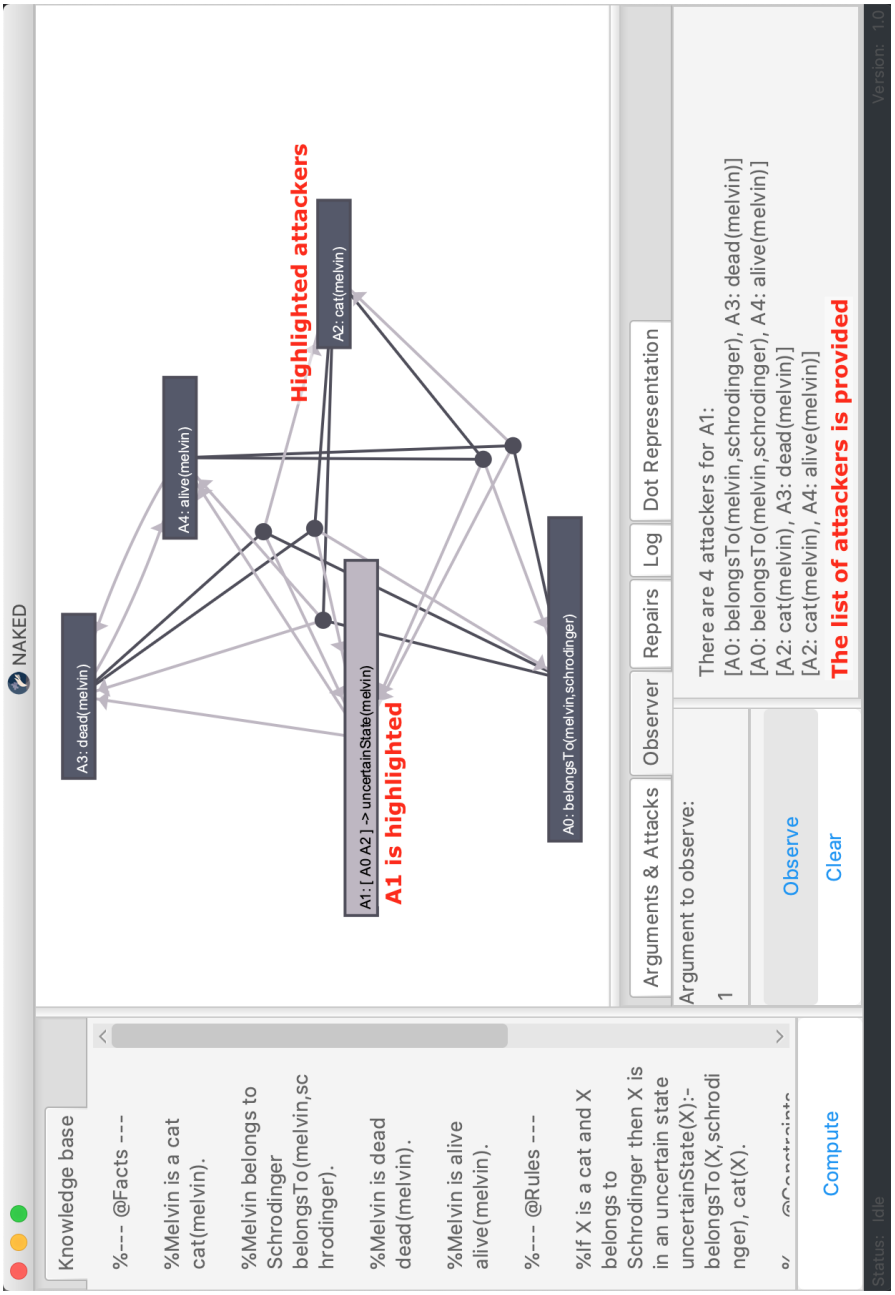

Figure 4.10: Argument highlight feature in NAKED

Although efficient algorithms that compute extensions exists for argumentation hypergraphs [Nielsen and Parsons, 2006], there is a lack of such graphs. Our tool provides a DOT format output which enables interoperability with many graph tools.

4.3 Summary

In this chapter, we presented a new hypergraph argumentation framework for the inconsistent knowledge bases expressed in the existential rules language. We showed that the instantiation of this argumentation framework possesses good properties and is semantically equivalent to the previous deductive argumentation framework while having less arguments and attacks. This new framework is a solution that effectively tackles the problem of the exponential growth of the argumentation framework that the previous deductive argumentation framework possessed.

Moreover, we provided the NAKED tool for generating, visualising and exporting such hypergraph argumentation frameworks in an intuitive way. This tool is useful for both argumentation experts that want to generate argumentation graphs for benchmarking purposes and for non computer science specialists that want to model a specific inconsistent knowledge base. For the later, the NAKED tool is specially useful as it can highlight the conflicts in the knowledge base by putting an argument and its attackers in the foreground.

In the next chapter, we focus on the benefits that ranking-based reasoning can yield not only in the context of argumentation but also in the more general setting of knowledge representation and reasoning for query answering by introducing novel frameworks.

Chapter 4 in a Nutshell

- We proposed a hypergraph argumentation framework for the existential rules language and showed that it possesses many desirable properties (rationality postulates and structural results). We conducted an experiment and showed that this new framework is more efficient than the old deductive framework.
- NAKED is an implementation of the hypergraph argumentation framework for *Datalog*± and is the first tool that allows the generation, the visualisation and the exportation of such argumentation graphs.

5

Ranking-Based Reasoning

5.1 Ranking with existential rules deductive argumentation framework . . . 116

5.1.1 Core equivalence . . . 117

5.1.2 Characterising ranking changes . . . 126

5.2 Ranking-based semantics with argumentation hypergraphs . . . 131

5.2.1 Properties for ranking-based semantics on hypergraphs . . . 132

5.2.2 The nh-categoriser . . . 136

5.3 Ranking facts in inconsistent knowledge bases . . . 142

5.3.1 The ranking-based inference framework . . . 143

5.3.2 $\mathcal{RIF}$ results . . . 148

5.4 Summary . . . 156

In the argumentation theory, the standard approach consists in using extension-based semantics [Baroni et al., 2011; Dung, 1995] aimed at evaluating which arguments can be accepted together. These semantics return subsets of arguments (called extensions) that can be used to evaluate the acceptability of arguments in a ternary way by checking whether an argument belongs to one, all or no extensions with respect to a particular argumentation semantics. However, this restricted gradation has been criticised as lacking for some applications such as online debates [Leite and Martins, 2011] or decision-making tools (an example can be found in the introduction of Chapter 2 in the thesis of Delobelle [2017]). That is why another approach, called ranking-based semantics, based on ranking arguments according to their acceptability was introduced.

Ranking-based semantics have been extensively studied by a large amount of researchers [Pereira et al., 2011; Amgoud and Ben-Naim, 2015; Gabbay and Rodrigues, 2015; Baroni et al., 2015; Amgoud et al., 2016; Amgoud and Ben-Naim, 2013; Bonzon et al., 2016; Rago et al., 2016; Yun et al., 2017a]. New semantics are being introduced, as well as the principles they should satisfy. One of the main reasons of their popularity is that they offer a finer evaluation than extension-based semantics [Dung, 1995; Caminada et al., 2012; Baroni et al., 2011; Caminada, 2007]. To put it in a nutshell,

classical argumentation semantics consider sets of arguments (extensions) whereas ranking-based semantics work at the level of individual arguments. Although these two approaches may seem distinct, they both give different details and insights on the underlying argumentation framework that they are applied on.

Among the many uses of ranking-based semantics in logic-based argumentation, Amgoud and Ben-Naim [2015] introduced the Argumentation-based Ranking Logics (ARL) as a framework for ranking conclusions of arguments using ranking-based semantics. However, the original work by Amgoud and Ben-Naim [2015] is only defined for argumentation frameworks without redundancies as these redundancies may "mislead the argumentation graph by creating more attacks and arguments" [Amgoud et al., 2014]. We argue that redundancies are an important part of real-life knowledge bases and interest ourselves into studying how much these redundancies can impact the ranking on conclusions of arguments returned. Moreover, our study will cover the behaviour of ranking-based semantics in the presence of redundant knowledge and how their removal (with the use of cores) impacts the ranking on facts returned.

Next, in the same fashion as works in the literature have extended ranking-based semantics to other particular binary argumentation frameworks such as weighted argumentation frameworks [Amgoud et al., 2017], support argumentation frameworks [Amgoud and Ben-Naim, 2016] or bipolar argumentation frameworks [Amgoud et al., 2008], we extended ranking-based semantics to the special case of hypergraph argumentation frameworks. We show that ranking-based semantics can also be applied to this type of argumentation frameworks by providing both desirable properties that these semantics should satisfy and the first ranking-based semantics called *nh-categoriser* for argumentation hypergraphs.

Lastly, inspiring ourselves from the frameworks defined by Bonzon et al. [2018], Konieczny et al. [2015] and Yun et al. [2018h], that use a ranking method and an aggregation function to obtain a ranking on sets of arguments, we propose a new framework for ranking repairs in the OBDA setting. This new framework shows that the ranking on the facts of an inconsistent knowledge base, obtained by computing the contribution of each fact in the intrinsic conflicts of the knowledge base, can be used to rank and improve the set of repairs of the knowledge base. We show in a real-life use-case how this framework can be useful in the context of the food industry and help professionals in ranking packaging alternatives.

The chapter is organized as follows: In Section 5.1, we focus on ranking-based reasoning with argumentation frameworks in the context of logic-based argumentation and we study the effect on redundancies on the rankings of argument conclusions. In Section 5.2, we extend the notion of ranking-based semantics for hypergraph argumentation frameworks by introducing both desirable properties that such semantics should satisfy and

the first ranking-based semantics for argumentation hypergraphs. In Section 5.3, we show how to improve the set of repairs with inconsistency values for query answering in the presence of inconsistencies.

Research Questions in this Chapter

- *What is the impact of redundancies on the rankings returned by ranking-based semantics?*
- *What are the properties that a ranking-based semantics for hypergraph argumentation frameworks should satisfy? Is there a semantics satisfying those properties?*
- *How can we rank repairs of a knowledge base without external information?*

5.1 Ranking with existential rules deductive argumentation framework

In the rest of this section, we do not follow the extension-based argumentation semantics but we focus on the effect of ranking-based semantics, in the setting of logic based argumentation. In the context of logic-based argumentation, arguments are sometimes based upon equivalent and redundant data. Cores are notions introduced by Amgoud et al. [2014] that delete such arguments. We investigate two different notions of core in such a logically instantiated argumentation framework that will remove redundant arguments and attacks in a different manner. We then ask the following research question:

"*Will the manner of defining the core of a logically instantiated argumentation framework affect the ranking output of ranking-based semantics?*"

Our initial intuition was that the answer was negative, since the core of an argumentation framework is supposed to return an equivalent, but smaller, argumentation framework. Surprisingly, the answer is positive. We give an example of such a change using one particular ranking-based semantics and show why this change happens. Our contribution is thus not only to uncover this unexpected behaviour but also to investigate some of its reasons. The salient points of this section are :

- The first investigation of ranking-based semantics in the first-order logic fragment of existential rules.
- The study of several notions of core in logical argumentation framework and the proof of their equivalences and properties.
- The first characterisation of core-induced ranking modifications of semantics satisfying postulates from Amgoud and Ben-Naim [2013] and Bonzon et al. [2016].

The *core* is a notion introduced by Amgoud et al. [2014] that enables to simplify logically instantiated argumentation frameworks without losing data. In this section we will use two notions of core initially defined by Amgoud et al. [2014] and we will adapt them to the logical instantiation using existential rules of this thesis. We give an example of how the two core notions yield argumentation frameworks with significantly less arguments for the same logical output and prove two new key results that extend the state-of-the-art. First, we give the relation between the base of the two cores for existential rules instantiated argumentation frameworks. Second, we show how the two cores can be obtained from each other.

5.1.1 Core equivalence

The notion of core relies on the notion of equivalence of formulae, arguments and, subsequently, of induced argumentation frameworks. To define the notion of core we first need to define the notion of equivalence of formulae. Adapting the work of Amgoud et al. [2014] for existential rules, two facts are equivalent if the sets given by the closure[1] of each fact are equal. Similarly, we say that two sets of facts are equal if, for each fact in every set, we can find an equivalent fact in the other set.

Definition 5.1 (Equivalent facts or sets of facts). *Let f_1, f_2 be two facts and $\mathcal{F}_1$, $\mathcal{F}_2$ be two sets of facts. We say that:*

- *f_1 and f_2 are equivalent ($f_1 \asymp f_2$) if and only if $Sat_{\mathcal{R}}(\{f_1\}) \equiv Sat_{\mathcal{R}}(\{f_2\})$.*
- *$\mathcal{F}_1$ and $\mathcal{F}_2$ are equivalent ($\mathcal{F}_1 \cong \mathcal{F}_2$) if and only if for every $f \in \mathcal{F}_1$, there exists $f' \in \mathcal{F}_2$ such that $f \asymp f'$ and for every $f \in \mathcal{F}_2$, there exists $f' \in \mathcal{F}_1$ such that $f \asymp f'$. We have that $\mathcal{F}_1 \ncong \mathcal{F}_2$ otherwise.*

Please note that in the first item of Definition 5.1, the $\equiv$ notation refers to the logical equivalence defined in Definition 2.4 on page 16.

Example 5.1 (Equivalent facts). *Let us consider the inconsistent knowledge base $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ expressing the fact that we cannot have a cat, a mouse and a turtle at the same time, cats are pussycats (and conversely, pussycats are cats), cats and mouses are mammals, Tom is a cat, Tom is a pussycat, Jerry is a mouse and John is a turtle.*

- $\mathcal{F} = \{pussycat(tom), cat(tom), mouse(jerry), turtle(john)\}$.
- $\mathcal{R} = \{\forall X(cat(X) \rightarrow pussycat(X)),$
  $\forall X(pussycat(X) \rightarrow cat(X)),$
  $\forall X(cat(X) \rightarrow mammal(X)),$
  $\forall X(mouse(X) \rightarrow mammal(X))\}$.
- $\mathcal{N} = \{\forall X, Y, Z(cat(X) \wedge mouse(Y) \wedge turtle(Z) \rightarrow \bot)\}$.

We have that $pussycat(tom) \asymp cat(tom)$ since $Sat_{\mathcal{R}}(\{pussycat(tom)\}) = Sat_{\mathcal{R}}(\{cat(tom)\}) = \{cat(tom), pussycat(tom), mammal(tom)\}$.

Using the equivalence of formulae from Definition 5.1 and inspiring ourselves from the work of Amgoud et al. [2014], we can now define the notion of equivalence between arguments. We will consider *two equivalence relations*. The first one ($\bowtie_1$) considers two arguments as being equivalent if

[1] In the following we consider that the rule application is using the restricted chase which does not consider redundant new facts generated by each step of the rule application (see more details in the work of Baget et al. [2014b]).

they have *equal supports and equivalent conclusions.* The second one ($\bowtie_2$) considers two arguments as being equivalent if they have *equivalent supports and equivalent conclusions.* Note that if there are two arguments a and a' such that $a \bowtie_1 a'$ then obviously $a \bowtie_2 a'$.

Definition 5.2 (Arguments equivalence [Amgoud et al., 2014]). *Let a and a' be two arguments. We have:*

- $a \bowtie_1 a'$ *if and only if* $Supp(a) = Supp(a')$ *and* $Conc(a) \cong Conc(a')$.
- $a \bowtie_2 a'$ *if and only if* $Supp(a) \cong Supp(a')$ *and* $Conc(a) \cong Conc(a')$.

Example 5.2 (Example 5.1 cont'd). *Let us consider the argumentation framework $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ such that the arguments of $\mathscr{A}$ are represented in Table 5.1. We have that $a_{13} \not\bowtie_1 a_1$ whereas $a_{13} \bowtie_2 a_1$ where $a_{13} = (\{cat(tom)\}, \{cat(tom)\})$ and $a_1 = (\{pussycat(tom)\}, \{pussycat(tom)\})$.*

We can note that if there is an attack between two arguments, they are not equivalent.

Proposition 5.1 (Attacked non equivalence). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework and $a, a' \in \mathscr{A}$. If $(a, a') \in \mathscr{R}$ then $a \not\bowtie_1 a'$ and $a \not\bowtie_2 a'$.*

Before we can define the notion of core, we first need to give the notions of equivalence relation, equivalence class and the set of all possible equivalence classes.

Definition 5.3 (Equivalence classes). *If X is a set of elements, $\bowtie$ an equivalence relation on X and $x \in X$, then $\bar{x}_{\bowtie} = \{x' \in X | x' \bowtie x\}$. We say that $\bar{x}_{\bowtie}$ is the equivalence class of an element x for equivalence relation $\bowtie$. The set of all possible equivalence classes will be denoted by $X/\bowtie = \{\bar{x}_{\bowtie} | x \in X\}$. Note that for readability purposes, we will sometimes denote $\bar{x}_{\bowtie}$ by $\bar{x}$ whenever the equivalence relation is obvious.*

We are now ready to define the notion of core of a logical argumentation framework. A core of an argumentation system $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ is an argumentation system that can be seen as a particular subgraph $\mathfrak{F}' = (\mathscr{A}', \mathscr{R}')$ of $\mathfrak{F}$. There are three restrictions. First, $\mathscr{A}'$ must obviously be a subset of the set of arguments $\mathscr{A}$. Second, for a given equivalence relation $\bowtie$ on the arguments, there must be a *unique* argument in $\mathscr{A}'$ for each equivalence class. Third, $\mathscr{R}'$ must be a restriction of $\mathscr{R}$ to $\mathscr{A}'$.

Definition 5.4 (Core [Amgoud et al., 2014]). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ and $\mathfrak{F}' = (\mathscr{A}', \mathscr{R}')$ be two argumentation frameworks and $\bowtie$ be an equivalence relation on arguments. $\mathfrak{F}'$ is a core of $\mathfrak{F}$ if and only if the three following items are satisfied:*

- $\mathscr{A}' \subseteq \mathscr{A}$
- *for every* $G \in \mathscr{A} / \bowtie$, *there exists a unique* $a \in \mathscr{A}$ *such that* $a \in G \cap \mathscr{A}'$ *for the given equivalence relation* $\bowtie$.
- $\mathscr{R}' = \mathscr{R} \cap (\mathscr{A}' \times \mathscr{A}')$.

We denote by $Core_{\bowtie}(\mathfrak{F})$ *the set of all cores of an argumentation framework* $\mathfrak{F}$ *for equivalence relation* $\bowtie$.

Note that since we consider two equivalence relations for arguments we can naturally construct two sets of cores from an argumentation framework $\mathfrak{F}$: $Core_{\bowtie_1}(\mathfrak{F})$ that follows from the first equivalence relation and $Core_{\bowtie_2}(\mathfrak{F})$ that follows from the second.

Example 5.3 (Example 5.2 cont'd). *We are interested in which arguments are contained in two different cores. Table 5.1 has five columns. The first three columns represents an example of 20 arguments (out of 75) that can be constructed over the knowledge base of Example 5.1 along with their respective supports and conclusions. The last two columns show whether the 20 arguments belong or not to two examples of cores* c_1 *and* c_2. *The two examples of cores have been constructed using respectively the first and the second equivalence relations: core* $c_1 \in Core_{\bowtie_1}(\mathfrak{F}_{\mathcal{KB}})$ *and core* $c_2 \in Core_{\bowtie_2}(\mathfrak{F}_{\mathcal{KB}})$ *(such that it is included in* $Core_{\bowtie_2}(c_1)$*, as it can be verified in Table 5.1).*

The next section shows properties of the two types of cores obtained from the two equivalence relations $\bowtie_1$ and $\bowtie_2$.

5.1.1.1 Core equivalence properties

Let us first summarise the theoretical results of this section. In Proposition 5.2, we show that the attack relation satisfies the properties defined by Amgoud et al. [2014] which implies equivalences between the argumentation framework and any of its cores. In Proposition 5.3, we show that it is useless to employ a more restrictive equivalence relation once a core has already been obtained using a less restrictive equivalence relation. In Proposition 5.4, we show that all cores constructed using $\bowtie_2$ can be constructed using specific cores of $\bowtie_1$ on which we compute a core using $\bowtie_2$. This basically means that we can bypass the core constructed with $\bowtie_1$ when we are interested by a less restrictive relation such as $\bowtie_2$. Proposition 5.3 and Proposition 5.4 combined provide an important result as it will allow us not to be concerned about the order of applying cores on the argumentation framework.

According to the work of Amgoud et al. [2014], there are two properties ($C1b$ and $C2b$) that are desirable for an attack relation. The first property states that two arguments with equivalent conclusions should attack the same arguments. The second property states that two arguments with

Name	Support	Conclusion	c_1	c_2
a_1	$\{P\}$	$\{P\}$	✓	✓
a_2	$\{P\}$	$\{M_T\}$	✓	✓
a_3	$\{M\}$	$\{M\}$	✓	✓
a_4	$\{M\}$	$\{M_J\}$	✓	✓
a_5	$\{T\}$	$\{T\}$	✓	✓
a_6	$\{P, M\}$	$\{P, M\}$	✓	✓
a_7	$\{P, M\}$	$\{M_T, M\}$	✓	✓
a_8	$\{P, M\}$	$\{P, M_J\}$	✓	✓
a_9	$\{P, M\}$	$\{M_T, M_J\}$	✓	✓
a_{10}	$\{P, T\}$	$\{P, T\}$	✓	✓
a_{11}	$\{P, T\}$	$\{M_T, T\}$	✓	✓
a_{12}	$\{T, M\}$	$\{T, M\}$	✓	✓
a_{13}	$\{C\}$	$\{C\}$	✓	
a_{14}	$\{C\}$	$\{M_T\}$	✓	
a_{15}	$\{C, M\}$	$\{C, M\}$	✓	
a_{16}	$\{C, M\}$	$\{M_T, M\}$	✓	
a_{17}	$\{C, M\}$	$\{C, M_J\}$	✓	
a_{18}	$\{C, M\}$	$\{M_T, M_J\}$	✓	
a_{19}	$\{C, T\}$	$\{C, T\}$	✓	
a_{20}	$\{C, T\}$	$\{M_T, T\}$	✓	

Acronym	Meaning
C	$cat(tom)$
M_J	$mammal(jerry)$
M_T	$mammal(tom)$
M	$mouse(jerry)$
P	$pussycat(tom)$
T	$turtle(john)$

Table 5.1: Some arguments constructed from the knowledge base of Example 5.12 and two particular cores obtained using $\bowtie_1$ and $\bowtie_2$.

equivalent supports should be attacked by the same arguments. In the next proposition, we show that the attack relation considered in this section respects both properties.

Proposition 5.2 (Attackers equivalence). *Let $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ be a logical argumentation framework such that $\mathscr{A}$ is the set of arguments defined in Definition 3.1 and $\mathscr{R}$ is the set of attacks defined according to Definition 3.2. $\mathscr{R}$ enjoys both of the following properties:*

1. *C1b : for every $a, b, c \in \mathscr{A}$, if $Conc(a) \cong Conc(b)$ then $((a, c) \in \mathscr{R}$ if and only if $(b, c) \in \mathscr{R})$.*

2. *C2b : for every $a, b, c \in \mathscr{A}$, if $Supp(a) \cong Supp(b)$ then $((c, a) \in \mathscr{R}$ if and only if $(c, b) \in \mathscr{R})$.*

Please see Section 7.2.3 on page xii for the proof of Proposition 5.2.

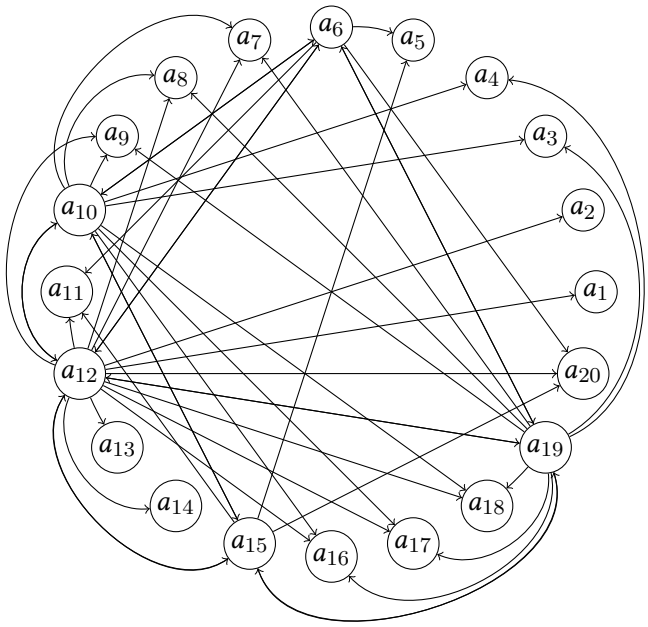

Figure 5.1: Representation of the core c_1 of $\mathfrak{F}_{\mathcal{KB}}$ using $\bowtie_1$ and displayed in Table 5.1

A natural question one can ask at this point is whether the order of applying the cores matters. That is to say: "Is it possible to first compute a core for a particular equivalence relation and then use it to compute another core?" To answer this question, we provide two main results. The first proposition shows that applying a more restrictive equivalence relation than the one used to compute a core does not change this core. We begin by defining the notion of less restrictive equivalence relation and follow with the proposition.

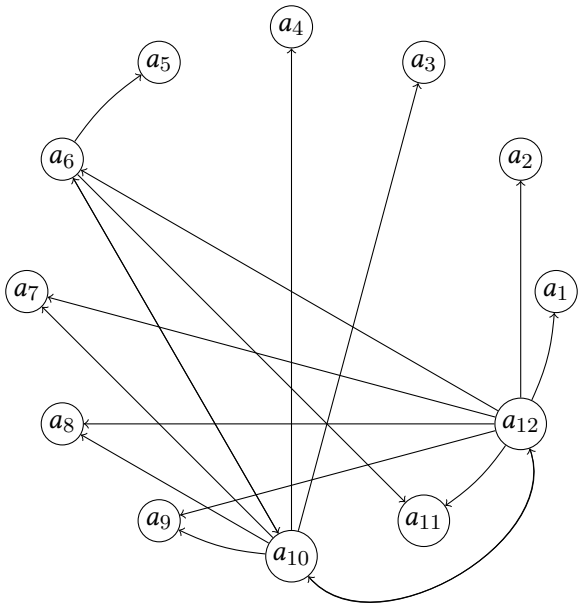

Figure 5.2: Representation of the core c_2 of $\mathfrak{F}_{\mathcal{KB}}$ using $\bowtie_2$ and displayed in Table 5.1

Definition 5.5 (More restrictive equivalence relation). *Let $\bowtie$ and $\bowtie'$ be two equivalence relation on a set of elements X, we say that $\bowtie$ is* more restrictive *than $\bowtie'$ (and thus, $\bowtie'$ is* less restrictive *than $\bowtie$) if and only if for every $x, x' \in X$ such that $x \bowtie x'$, it holds that $x \bowtie' x'$.*

A trivial result is that a more restrictive equivalence relation means that less arguments are equivalent. Likewise, a less restrictive equivalence relation will have more equivalent arguments and thus filter more arguments.

Proposition 5.3 (No filtration result). *Let $\mathfrak{F}$ be an argumentation framework and $\bowtie, \bowtie'$ be two equivalence relation such that $\bowtie$ is more restrictive than $\bowtie'$. It holds that:*

$$\textit{for every } c' \in Core_{\bowtie'}(\mathfrak{F}), Core_{\bowtie}(c') = \{c'\}$$

Proof. Suppose that we have for every $a, a' \in \mathscr{A}$, it holds that if $a \bowtie a'$ then $a \bowtie' a'$. Now, let us consider $c' = (\mathscr{A}_{c'}, \mathscr{R}_{c'}) \in Core_{\bowtie'}(\mathfrak{F})$ and $c = (\mathscr{A}_c, \mathscr{R}_c) \in Core_{\bowtie}(c')$. We denote by X the set such that $\mathscr{A}_{c'} = X \cup \mathscr{A}_c$ and $X \cap \mathscr{A}_c = \emptyset$. If $X \neq \emptyset$, it means that there exists $b \in X$ and $b' \in \mathscr{A}_c$ such that $b \not\bowtie' b'$ and $b \bowtie b'$, contradiction. It follows that $X = \emptyset$ and that c' is the only core of $Core_{\bowtie}(c')$. □

We now prove that the set of cores of an argumentation framework $\mathfrak{F}$ obtained using the equivalence relation $\bowtie_2$ is equal to the set of cores that are built on cores of $\mathfrak{F}$ using $\bowtie_1$.

Proposition 5.4 (Core construction equivalence). *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework and $\bowtie_1, \bowtie_2$ be the equivalence relations defined in Definition 5.2. It holds that:*

$$Core_{\bowtie_2}(\mathfrak{F}) = \bigcup_{c_1 \in Core_{\bowtie_1}(\mathfrak{F})} Core_{\bowtie_2}(c_1)$$

Please refer to Section 7.2.3 on page xiii for the complete proof of Proposition 5.4. From Proposition 5.3 and Proposition 5.4, the next proposition holds.

Proposition 5.5 (Symmetric construction equivalence). *Let $\mathfrak{F}$ be an arbitrary argumentation framework and $\bowtie_1, \bowtie_2$ be the equivalence relations defined in Definition 5.2. It holds that:*

$$\bigcup_{c_2 \in Core_{\bowtie_2}(\mathfrak{F})} Core_{\bowtie_1}(c_2) = \bigcup_{c_1 \in Core_{\bowtie_1}(\mathfrak{F})} Core_{\bowtie_2}(c_1)$$

Proposition 5.5 is important for the next section that characterises ranking changes induced by cores as it tells us that if we are only concerned by $\bowtie_2$-induced ranking changes, we can bypass the core obtained via $\bowtie_1$.

5.1.1.2 Rankings on different cores

Now that we have investigated the notions of core for an argumentation framework, we can study how ranking-based semantics behave on them. In the work of Amgoud and Ben-Naim [2015], the authors define the notion of Argumentation-based Ranking Logic (ARL) that takes a knowledge base as input and, using ranking-based semantics on the instantiated argumentation graph, provides a ranking of the formulae of the knowledge base. In the following we adapt their results and provide the corresponding existential rules argumentation-based ranking logic. The extension we provide is two fold. First we consider the existential rules framework for instantiation. Second, we take into account the notion of core in the reasoning mechanism. The new process is composed of four steps:

1. First, an argumentation framework is instantiated from a knowledge base $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ (see Example 5.12 for the knowledge base considered as example throughout this section).

2. Second, a core c constructed using an equivalence relation is considered (see Table 5.1 for two examples of cores on the knowledge base from Example 5.12 considering the two equivalence relations defined in Definition 5.2, and Figures 5.1 and 5.2 for the visual depiction of the cores as graphs).

3. Third, the arguments of c are ranked using a ranking-based semantics σ[2] (see Table 5.2 and Table 5.3 for the ranking of the arguments of the two cores from Figures 5.1 and 5.2 and Table 5.1 outputted by the burden-based and discussion-based ranking-based semantics [Amgoud and Ben-Naim, 2013]).

4. Finally, their conclusions are ranked following a simple principle: a conclusion is ranked higher than another conclusion if it is supported by an argument which is ranked higher than any argument supporting the second conclusion (see Table 5.4 and Table 5.5).

a_1, a_2, a_{13}, a_{14}
a_3, a_4, a_5
$a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{15}, a_{16}, a_{17}, a_{18}, a_{19}, a_{20}$
a_{12}

Table 5.2: Ranking on arguments of c_1 using the burden-based (and discussion-based) ranking-based semantics

a_1, a_2, a_3, a_4, a_5
$a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}$

Table 5.3: Ranking on arguments of c_2 using the burden-based (and discussion-based) ranking-based semantics

Before commenting on the results of ranking on the knowledge base, let us first define the ARL for existential rules. The definition that we provide is inspired by the work of Amgoud and Ben-Naim [2015] adapted for existential rules and the notion of core. We recall that the ranking on arguments resulting from a ranking-based semantics σ on an argumentation framework $\mathfrak{F}$ will be denoted by $\preceq^{\sigma}_{\mathfrak{F}}$ or simply by $\preceq$ if there is no ambiguity. For two arguments $a, b \in \mathscr{A}$, the notation $a \preceq b$ means that b is at least as acceptable as a.

Definition 5.6 (Existential rules ARL). *An existential rules ARL is a tuple:*

$$\mathcal{ARL} = (\mathcal{KB}, \mathfrak{F}, c, \sigma, K, \mathfrak{C}')$$

[2]We recall that ranking-based semantics are defined in Definition 2.51 on page 40.

$\{P\}, \{M_T\}, \{C\}$
$\{M\}, \{M_J\}, \{T\}$
$\{P, M\}, \{M_T, M\}, \{P, M_J\}, \{M_T, M_J\}, \{P, T\}, \{M_T, T\}, \{C, M\}, \{C, M_J\}, \{C, T\}$
$\{T, M\}$

Table 5.4: Ranking K using the ranking on arguments of Table 5.2

$\{P\}, \{M_T\}, \{C\}, \{M\}, \{M_J\}, \{T\}$
$\{P, M\}, \{M_T, M\}, \{P, M_J\}, \{M_T, M_J\}, \{P, T\}, \{M_T, T\}, \{C, M\}, \{C, M_J\}, \{C, T\}, \{T, M\}$

Table 5.5: Ranking K using the ranking on arguments of Table 5.3

where $\mathcal{KB}$ is an inconsistent knowledge base, $\mathfrak{F}$ is the instantiated argumentation framework (this may be omitted if the instantiation used is clear), $c = (\mathscr{A}', \mathscr{R}')$ is the chosen core of $\mathfrak{F}$ for a given equivalence relation $\bowtie$, σ is a ranking-based semantics and $K, \mathfrak{C}'$ are defined as follows:

- *for every $X \subseteq \mathcal{F}, \mathfrak{C}'(X) = \{Y|$ there exists $a \in \mathscr{A}' \cap Arg(X)$ such that $Conc(a) \cong Y\}$, i.e. $\mathfrak{C}'(X)$ is the set of equivalent sets of facts that can be concluded by arguments of c constructed on subsets of X.*
- *for every $X \subseteq \mathcal{F}$ and for every $Y, Z \in \mathfrak{C}'(X), (Y, Z) \in K(X)$ if and only if there exists $a, b \in \mathscr{A}' \cap Arg(X)$ such that $Conc(a) \cong Y, Conc(b) \cong Z$ and $a \preceq_c^\sigma b$. $K(X)$ corresponds to a ranking on elements of $\mathfrak{C}'(X)$ obtained via the ranking of arguments σ on c.*

Note that the equivalence relation is absent from $\mathcal{ARL}$ because the core is already given. Let us now show by the means of an example that the ranking-based semantics considered (namely burden-based) is sensitive to the notion of core and thus outputs different rankings for logically equivalent argumentation graphs.

Example 5.4 (Example 5.1 cont'd). *Let c_1 (respectively c_2) be the core of $\mathfrak{F}_{\mathcal{KB}}$ using equivalence relation $\bowtie_1$ (respectively $\bowtie_2$). The argumentation graph of c_1 (respectively c_2) is represented in Figure 5.1 (respectively Figure 5.2). The ranking on arguments of c_1 (respectively c_2) computed with the burden-based ranking-based semantics is given in Table 5.2 (respectively Table 5.3). Finally, the ranking of conclusions is computed and displayed in Table 5.4 (respectively Table 5.5). Note that in this example, the discussion-based ranking-based semantics gives the same ranking.*

This example shows that, surprisingly, a core does not always have the same ranking as the original argumentation framework (since c_1 and c_2 have different rankings). For instance, a_1 is ranked higher than a_3 for c_1 (Table 5.2) but a_1 is ranked equal to a_3 for c_2 (Table 5.3). This change in the

ranking of arguments is significant as it impacts the ranking of their conclusions. Hence, the atom $cat(tom)$ *which was ranked higher than* $mouse(jerry)$ *and* $turtle(john)$ *(Table 5.4) is now ranked equal to them (Table 5.5). This is caused by the existence of equivalences (here,* $cat(tom)$ *is equivalent to* $pussycat(tom)$*) in the knowledge base. In fact, these equivalences generate redundant attacks between arguments that decrease the ranking of other arguments. That is why, by deleting redundancy in cores, we can observe that the ranking of some arguments is modified.*

Hence, the chosen equivalence relation also plays a role in the ranking (as we have different rankings for the two cores). The next subsection investigates the reasons for such a behaviour.

5.1.2 Characterising ranking changes

In the rest of this section, we consider an argumentation framework and one of its cores constructed either using $\bowtie_1$ or $\bowtie_2$. We give a necessary and sufficient condition for obtaining an equal (with respect to the set of arguments) $\bowtie_1$ and $\bowtie_2$ induced core from its original argumentation framework. Then, for those argumentation frameworks where the **induced core is different**, we provide sufficient conditions for characterising the difference between the ranking of the core and the one of its original argumentation framework. More precisely:

1. We provide a sufficient condition **for an argument's rank to increase** in the induced core. The new ranking of these arguments is further characterised by a sufficient condition on their respective positions. This is done via the *CP* postulate characterisation.

2. We provide a sufficient condition for **an argument's rank to remain unchanged** in the induced core. This is done via the *NaE* postulate characterisation.

3. Last, we provide a sufficient condition for **an argument's rank to decrease** in the induced core. This is done via the *CP* and *SCT* postulates characterisation.

Identity of induced core. We begin by introducing the notation needed for the rest of this section.

Definition 5.7 (Different core). *Let us consider an argumentation framework* $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ *and one of its cores* $c = (\mathscr{A}', \mathscr{R}')$ *with respect to an equivalence relation, we denote by* X_c *(or* X *if the core is obvious) the set of arguments that have been deleted, namely* $\mathscr{A} = \mathscr{A}' \cup X_c$ *and* $X_c \cap \mathscr{A}' = \emptyset$*. If* $X_c \neq \emptyset$ *then the core is said to be* different *from* $\mathfrak{F}$*, otherwise it is* not different *from* $\mathfrak{F}$*.*

The next proposition gives a necessary and sufficient condition for all cores using $\bowtie_1$ of an argumentation framework $\mathfrak{F}$ to be *not different* from $\mathfrak{F}$.

Proposition 5.6 (Not different core characterisation $\bowtie_1$). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and $\mathfrak{F}_{\mathcal{KB}}$ be the corresponding argumentation framework. We have that $Core_{\bowtie_1}(\mathfrak{F}_{\mathcal{KB}}) = \{\mathfrak{F}_{\mathcal{KB}}\}$ if and only if for every $\mathcal{R}$-consistent subset Y of $\mathcal{F}$, there are no y_1, y_2 such that $Sat_{\mathcal{R}}(Y) \models y_1, Sat_{\mathcal{R}}(Y) \models y_2$, $y_1 \neq y_2$ and $y_1 \asymp y_2$.*

Please refer to Section 7.2.3 on page xiv for the proof of Proposition 5.6.

Similarly, we show a necessary and sufficient condition for all core using $\bowtie_2$ of an argumentation framework $\mathfrak{F}$ to be *no different* than $\mathfrak{F}$.

Proposition 5.7 (Not different cores characterisation $\bowtie_2$). *Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and $\mathfrak{F}_{\mathcal{KB}}$ be the corresponding argumentation framework. We have $Core_{\bowtie_2}(\mathfrak{F}_{\mathcal{KB}}) = \{\mathfrak{F}_{\mathcal{KB}}\}$ if and only if both the two following items are satisfied:*

- *there are no $f_1, f_2 \in \mathcal{F}$ such that $f_1 \asymp f_2$ and $f_1 \neq f_2$*
- *for every $\mathcal{R}$-consistent subset $Y \subseteq \mathcal{F}$, there are no y_1, y_2 such that $Sat_{\mathcal{R}}(Y) \models y_1, Sat_{\mathcal{R}}(Y) \models y_2, y_1 \neq y_2$ and $y_1 \asymp y_2$.*

Please refer to Section 7.2.3 on page xiv for the proof of Proposition 5.7.

Rank increase. From now on, we consider an argumentation framework $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ and $c' = (\mathscr{A}', \mathscr{R}')$, one of its cores for equivalence relation $\bowtie_1$ or $\bowtie_2$. An interesting property is that for each attack from an argument removed by the core towards an argument of the core, we can find an attack that comes from an argument of the core towards that same argument.

Proposition 5.8 (Attack equivalence). *Let us consider $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$, $c' = (\mathscr{A}', \mathscr{R}')$ one of its cores for equivalence relation $\bowtie_1$ or $\bowtie_2$, $E = \{(a, b) \in \mathscr{R} \mid a \in X_{c'}$ and $b \in \mathscr{A}'\}$ and $E' = \{W \subseteq \mathscr{R} \mid W$ is maximal for set inclusion such that for every $(w_i, w_j), (w_k, w_l) \in W, wi \bowtie w_k, w_j = w_l, \{wi, w_k\} \subseteq X_{c'}$ and $w_j, w_l \notin X_{c'}\}$.*

The function $f : \mathscr{R}' \to E'$, that associates to each attack $(a', b') \in \mathscr{R}'$ a set of attacks $W \in E'$ such that for every $(w_i, w_j) \in W, w_i \bowtie a'$ and $w_j = b'$, is surjective.

Proof. Let us consider $W \in E'$ and an element $(w_i, w_j) \in W$. Then since c' is a core of $\mathfrak{F}$ for $\bowtie_1$ (respectively $\bowtie_2$), we have that there exists a unique $z \in \bar{w}_{i\bowtie_2} \cap \mathscr{A}'$ (respectively $\bar{w}_{i\bowtie_1} \cap \mathscr{A}'$). Then, using Proposition 5.2, we get that $(z, w_j) \in \mathscr{R}'$. □

This proposition means that the modification of the ranking is induced mainly by a quantitative loss. We now introduce the notion of graph isomorphism which will be used to clone our argumentation frameworks.

Definition 5.8 (Isomorphism). *Let G_1, G_2 be two directed graphs. Let $V(G_i)$ denote the set of vertices of G_i and $E(G_i)$ the set of its arcs. We say that $\gamma : V(G_1) \to V(G_2)$ is an isomorphism from G_1 to G_2 if and only if for every $(x, y) \in E(G_1)$, $(\gamma(x), \gamma(y)) \in E(G_2)$. For simplicity purposes, we will also write $G2 = \gamma(G_1)$.*

Using the previous Proposition 5.8, we can have a better understanding as to why some arguments have better ranking in a core than in $\mathfrak{F}$ with some ranking-based semantics. The reason is because some arguments of the core c' that have equivalent arguments in $X_{c'}$ (with respect to $\bowtie_2$ or $\bowtie_1$) have their attacks amplified by those arguments. Of course, depending on the ranking-based semantics used, having more attackers does not always mean that the ranking of the argument is worst. This concept corresponds to the Cardinality Postulate (*CP*) postulate [Bonzon et al., 2016; Amgoud and Ben-Naim, 2013].

Definition 5.9 (CP [Amgoud and Ben-Naim, 2013]). *Let σ be a ranking-based semantics and $\preceq^{\sigma}_{\mathfrak{F}}$ be the ranking obtained after applying σ on $\mathfrak{F}$. We say that σ satisfies CP if and only if for every $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ and for every $a, b \in \mathscr{A}$ such that $|Att^-(a)| < |Att^-(b)|$, it holds that $b \preceq^{\sigma}_{\mathfrak{F}} a$ and $a \not\preceq^{\sigma}_{\mathfrak{F}} b$.*

Note that the *burden-based* and the *discussion-based* ranking-based semantics both satisfy the *CP* postulate [Amgoud and Ben-Naim, 2013].

We are now interested in the impact of arguments removed by a core on arguments that belong to the core.

Definition 5.10 (Set of arguments attacked by filtrated arguments). *Let $\mathfrak{F}$ be an argumentation framework and c' be one of its cores. We denote by $J_{c'}$ (or J if the core is obvious) the set of arguments of c' that have at least one attacker that is equivalent to an argument in $X_{c'}$. More precisely, $J = \{a \in \mathscr{A}' |$ there exists $(e, a) \in \mathscr{R}'$ and $f((e, a)) \neq \emptyset\}$, where f is the function defined in Proposition 5.8.*

Example 5.5. *Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework and $c' = (\mathscr{A}', \mathscr{R}')$ be a core of $\mathfrak{F}$ for an equivalence relation. In this example depicted in Figure 5.3, we have $\mathscr{A} = \{a, b, c, d, e, i, g, h\}$ and $\mathscr{R} = \{(i, a), (g, a), (c, b), (d, b), (e, b), (h, b)\}$. Suppose that $\bar{i} = \{i, g\}$ and $\bar{c} = \{c, d, e\}$. The core c' is such that $\mathscr{A}' = \{a, i, c, b, h\}$ and $J_{c'} = \{a, b\}$.*

The next proposition states that every argument of the core that is attacked by an argument equivalent to a deleted argument is ranked better in the core.

Proposition 5.9 (Argument rank increase). *Let $\mathfrak{F}, \mathfrak{F}'$ be two argumentation frameworks, c' be a core of $\mathfrak{F}$ with respect to an equivalence relation, γ be an isomorphism such that $\mathfrak{F}' = \mathfrak{F} \oplus \gamma(c')$ and σ be a ranking-based*

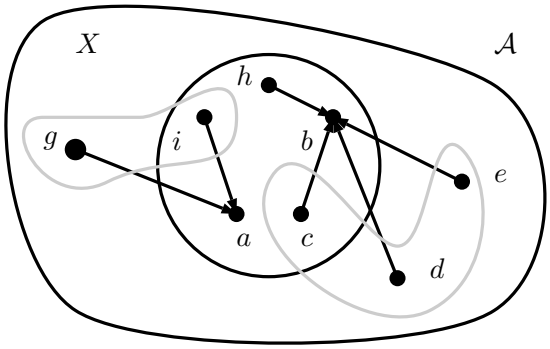

Figure 5.3: Representation of an argumentation framework $\mathfrak{F}$ and one of its cores c'

semantics that satisfies CP. It holds that for every $b \in J_{c'}, b \preceq^{\sigma}_{\mathfrak{F}'} \gamma(b)$ *and* $\gamma(b) \not\preceq^{\sigma}_{\mathfrak{F}'} b$.

Proof. Let (a, b) be an attack of c' such that $f((a, b)) \neq \emptyset$. It means that there exists an argument $a' \in X_{c'}$ such that $(a', b) \in \mathscr{R}$. We thus have $|Att^{-}(\gamma(b))| < |Att^{-}(b)|$ and since σ satisfies CP, $b \preceq^{\sigma}_{\mathfrak{F}'} \gamma(b)$ and $\gamma(b) \not\preceq^{\sigma}_{\mathfrak{F}'} b$. □

In Proposition 5.9, we showed that some arguments of the core may be ranked higher. We now proceed further in this direction by introducing a sufficient condition for characterising the ranking of such arguments.

Proposition 5.10 (Rank characterisation). *Let* $a, b \in J$. *If* σ *satisfies CP and* $\left[|Att^{-}(a)| - \sum_{e \in Att^{-}(a) \cap \mathscr{A}'} |f((e, a))|\right] < \left[|Att^{-}(b)| - \sum_{e \in Att^{-}(b) \cap \mathscr{A}'} |f((e, b))|\right]$ *then* $b \preceq^{\sigma}_{c'} a$ *and* $a \not\preceq^{\sigma}_{c'} b$.

Proof. We have for all arguments a in $\mathscr{A}'$, $|Att^{-}(a)| - \sum_{e \in Att^{-}(a) \cap \mathscr{A}'} |f((e, a))| = |Att^{-}(a) \cap \mathscr{A}'|$. Thus, we can say that $|Att^{-}(a) \cap \mathscr{A}'| < |Att^{-}(b) \cap \mathscr{A}'|$. Since σ is a semantics that satisfy CP, $b \preceq^{\sigma}_{c'} a$ and $a \not\preceq^{\sigma}_{c'} b$. □

Example 5.6 (Example 5.5 cont'd). *We have that* $f((i, a)) = \{(g, a)\}$, $f((c, b)) = \{(d, b), (e, b)\}$ *and* $f((h, b)) = \emptyset$. *Thus, we can compute that* $|Att^{-}(a)| - \sum_{e \in Att^{-}(a) \cap \mathscr{A}'} |f((e, a))| = 1$ *and* $|Att^{-}(b)| - \sum_{e \in Att^{-}(b) \cap \mathscr{A}'} |f((e, b))| = 4 - 2 = 2$. *We conclude that under a ranking-based semantics* σ *satisfying CP,* $b \preceq^{\sigma}_{c'} a$ *and* $a \preceq^{\sigma}_{c'} b$.

Unchanged rank. We now give a sufficient condition for an argument to keep the same rank. The basic notion behind this is that arguments that are not attacked by others do not undergo a change in their rank. This is true if the Non Attacked Equivalence (NaE) postulate is satisfied, namely if all the non-attacked argument have the same rank.

Definition 5.11 (NaE [Amgoud and Ben-Naim, 2013]). *We say that a ranking-based semantics σ satisfies the NaE if and only if for every argumentation framework $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ and for every $a, b \in \mathscr{A}$ such that $Att^-(a) = Att^-(b) = \emptyset$, it holds that $a \preceq^{\sigma}_{\mathfrak{F}} b$ and $b \preceq^{\sigma}_{\mathfrak{F}} a$.*

Note that the *burden-based*, *discussion-based*, *the h-categoriser* [Besnard and Hunter, 2001] and the *Tuples* [Cayrol and Lagasquie-Schiex, 2005] ranking-based semantics satisfy the *NaE* postulate.

Proposition 5.11 (Unchanged non attacked arguments). *Let $\mathfrak{F}$ and $\mathfrak{F}'$ be two argumentation frameworks, $c' = (\mathscr{A}', \mathscr{R}')$ be a core of $\mathfrak{F}$, $a \in \mathscr{A}'$, $Att^-(a) = \emptyset$ and γ be an isomorphism such that $\mathfrak{F}' = \mathfrak{F} \oplus \gamma(c')$. If σ is a ranking-based semantics that satisfies NaE then $a \preceq^{\sigma}_{\mathfrak{F}'} \gamma(a)$ and $\gamma(a) \preceq^{\sigma}_{\mathfrak{F}'} a$.*

Proof. We know that the core c' has fewer arguments and attacks than $\mathfrak{F}$. Thus, the argument a is not attacked in either c' or $\gamma(c')$. Furthermore, since σ satisfies *NaE*, $\gamma(a)$ and a are equivalent. □

Rank decrease. In the next proposition, we introduce a sufficient condition for an argument of the core to have its rank decreased. This condition holds if the semantics used for the ranking satisfies the *CP* and Strict Counter Transitivity (*SCT*) postulates. The *SCT* postulate specifies that if the attackers of an argument b are at least as numerous and acceptable as those of an argument a and either the attackers of b are strictly more numerous or acceptable than those of a, then a is strictly more acceptable than b.

Definition 5.12 (SCT [Amgoud and Ben-Naim, 2013]). *We say that a ranking-based semantics σ satisfies SCT if and only if for every argumentation framework $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ and for every $a, b \in \mathscr{A}$ such that there is an injective mapping $g : Att^-(a) \to Att^-(b)$ with for every $a' \in Att^-(a), a' \preceq^{\sigma}_{\mathfrak{F}} g(a')$ and ($|Att^-(a)| < |Att^-(b)|$ or there exists $a' \in Att^-(a)$, $a' \preceq^{\sigma}_{\mathfrak{F}} g(a'), g(a') \not\preceq^{\sigma}_{\mathfrak{F}} a'$) then $b \preceq^{\sigma}_{\mathfrak{F}} a$ and $a \not\preceq^{\sigma}_{\mathfrak{F}} b$.*

Note that the *burden-based*, *discussion-based* and *the h-categoriser* ranking-based semantics satisfy the *SCT* postulate.

The idea behind the next proposition is that if an argument's attackers have their ranks increased, then its rank is reduced.

Proposition 5.12 (Argument rank decrease). *Let $\mathfrak{F}$ and $\mathfrak{F}'$ be two argumentation frameworks, $c' = (\mathscr{A}', \mathscr{R}')$ be a core of $\mathfrak{F}$ for an arbitrary equivalence relation, a be an argument of $\mathscr{A}'$ with $a \notin J_{c'}$ and γ be an isomorphism such that $\mathfrak{F}' = \mathfrak{F} \oplus \gamma(c')$. If σ is a semantics that satisfies CP and SCT and $Att^-(a) \subseteq J_{c'}$ then $\gamma(a) \preceq^{\sigma}_{\mathfrak{F}'} a$ and $a \not\preceq^{\sigma}_{\mathfrak{F}'} \gamma(a)$.*

Proof. Since $a \notin J_{c'}$, we have that $Att^-(\gamma(a)) = \{\gamma(a') | a' \in Att^-(a)\}$ and thus $|Att^-(a)| = |Att^-(\gamma(a))|$. Now, since $Att^-(a) \subseteq J_{c'}$, we have that for every $b \in Att^-(b), b \preceq^{\sigma}_{\mathfrak{F}'} \gamma(b)$ and $\gamma(b) \not\preceq^{\sigma}_{\mathfrak{F}'} b$ (using Proposition 5.9). Finally, using the *SCT* postulate, we conclude that $\gamma(a) \preceq^{\sigma}_{\mathfrak{F}'} a$ and $a \not\preceq^{\sigma}_{\mathfrak{F}'} \gamma(a)$. □

5.2 Ranking-based semantics with argumentation hypergraphs

As shown in the previous chapter, hypergraph argumentation frameworks can be very useful in the context of logic-based argumentation. However, it was not as extensively studied as its binary counterpart. Therefore, many notions such as supports, preferences, weights or ranking-based semantics have not been implemented yet for this framework. Against this background, we thrive to generalise the ranking-based semantics for argumentation hypergraphs by (1) translating desirable properties for ranking-based semantics on binary argumentation frameworks for hypergraph argumentation framework and (2) defining the first ranking-based semantics for hypergraph argumentation framework.

The structure of this section is as follows. In Section 5.2.1, we show how the existing properties for ranking-based semantics can translate in the context of hypergraph argumentation framework. In Section 5.2.2, we introduce the nh-categoriser as the first ranking-based semantics for hypergraph argumentation framework.

Example 5.7. *In this section, we will consider the hypergraph argumentation framework* $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ *such that:*

- $\mathscr{A}^\star = \{a, b, c, d, e\}$
- $\mathscr{R}^\star = \{(\{d\}, a), (\{b\}, a), (\{b\}, c), (\{b, e\}, d), (\{d, c\}, e), (\{a, c\}, e)\}$.

A ranking-based semantics $\sigma^\star$ is a function that returns a ranking on argument for every hypergraph argumentation framework.

Definition 5.13 (Ranking-based semantics $\sigma^\star$). *A ranking-based semantics* $\sigma^\star$ *associates to any hypergraph argumentation framework* $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ *a ranking* $\geq^{\sigma^\star}_{\mathfrak{F}^\star}$ *on* $\mathscr{A}^\star$ *where* $\geq^{\sigma^\star}_{\mathfrak{F}^\star}$ *is a total preorder (reflexive and transitive relation) on* $\mathscr{A}^\star$*. The notation* $a \geq^{\sigma^\star}_{\mathfrak{F}^\star} b$ *means that* a *is at least as acceptable as* b*.*

Notation 5.1. *As usual, the notation* $a \succ^{\sigma^\star}_{\mathfrak{F}^\star} b$ *is used for* $a \geq^{\sigma^\star}_{\mathfrak{F}^\star} b$ *and* $b \not\geq^{\sigma^\star}_{\mathfrak{F}^\star} a$*. Likewise, we use* $a \approx^{\sigma^\star}_{\mathfrak{F}^\star} b$ *for* $a \geq^{\sigma^\star}_{\mathfrak{F}^\star} b$ *and* $b \geq^{\sigma^\star}_{\mathfrak{F}^\star} a$*. By abuse of notation, if* S *and* S' *are two sets of arguments, we write that* $S \geq^{\sigma^\star}_{\mathfrak{F}^\star} S'$ *if and only if for every* $s \in S$ *and every* $s' \in S'$*, we have* $s \geq^{\sigma^\star}_{\mathfrak{F}^\star} s'$*. Let* S *be a set in* $(2^{\mathscr{A}^\star} \setminus \emptyset)$ *and* $\sigma^\star$ *be a ranking-based semantics, the set* $\min^{\sigma^\star}_{\mathfrak{F}^\star}(S)$ *is the set* $\{s \in S \mid \text{for every } s' \in S, s' \geq^{\sigma^\star}_{\mathfrak{F}^\star} s\}$*.*

Definition 5.14 (Path). *Let* $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ *and* $a \in \mathscr{A}^\star$*. We say that a sequence of attacks* $((S_1, t_1), \ldots, (S_n, t_n))$ *is a path of size* n *from* S_1 *to* a *if and only if for every* $i \in \{1, \ldots, n\}, (S_i, t_i) \in \mathscr{R}^\star$*,* $t_n = a$ *and for every* $i \in \{1, \ldots, n-1\}$*, it holds that* $t_i \in S_{i+1}$*.*

Notation 5.2. *Let* $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}, S \in 2^{\mathscr{A}^\star}$ *and* $a \in \mathscr{A}^\star$*, we say that* $S \in \mathscr{R}^{\star}{}^{-}_{n}(a)$ *if and only if there exists a path of size* n *from* S *to* a*. We say that* $S \in \mathscr{R}^{\star}{}^{-}_{1}(a)$ *is a direct attacker of* a *and* $S \in \mathscr{R}^{\star}{}^{-}_{2}(a)$ *is a direct defender of* a*.*

Example 5.8 (Example 5.7 cont'd). *The sequence* $((\{d\}, a), (\{a, c\}, e))$ *is a path of size 2 from* $\{d\}$ *to* e*. Thus,* $\{d\} \in \mathscr{R}^{\star}{}^{-}_{2}(e)$ *and* $\{d\}$ *is a direct defender of* e*.*

5.2.1 Properties for ranking-based semantics on hypergraphs

We first introduce the definition of an isomorphism between two argumentation hypergraphs.

Definition 5.15 (Isomorphism). *An isomorphism between two argumentation hypergraphs* $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ *and* $\mathfrak{F}^{\star\prime} = (\mathscr{A}^{\star\prime}, \mathscr{R}^{\star\prime})$ *is a bijective function* $\gamma : \mathscr{A}^\star \to \mathscr{A}^{\star\prime}$ *such that for every* $S \in 2^{\mathscr{A}^\star}$ *and every* $a \in \mathscr{A}^\star, (S, a) \in \mathscr{R}^\star$ *if and only if* $(\{\gamma(s) \mid s \in S\}, \gamma(a)) \in \mathscr{R}^{\star\prime}$*. With a slight abuse of notation, we will note* $\mathfrak{F}^{\star\prime} = \gamma(\mathfrak{F}^\star)$*.*

In the rest of this section, we recall the properties for a ranking-based semantics $\sigma^\star$ that have been defined in the literature [Amgoud and Ben-Naim, 2013; Bonzon et al., 2016] and we translate them for hypergraph argumentation frameworks. We now begin with the properties that can be translated straightforwardly.

The Abstraction property states that the name of the arguments should not be taken into account for the ranking.

Property 5.1 (Abstraction). *We say that* $\sigma^\star$ *satisfies* Abstraction *if and only if for any* $\mathfrak{F}^\star, \mathfrak{F}^{\star\prime} \in \mathfrak{K}_{\mathfrak{F}^\star}$ *and isomorphism* γ *such that* $\mathfrak{F}^{\star\prime} = \gamma(\mathfrak{F}^\star)$*, we have* $a \geq^{\sigma^\star}_{\mathfrak{F}^\star} b$ *if and only if* $\gamma(a) \geq^{\sigma^\star}_{\mathfrak{F}^{\star\prime}} \gamma(b)$

The Independence property states that two arguments with no paths connecting them should not influence each other.

Property 5.2 (Independence). *We say that* $\sigma^\star$ *satisfies* Independence *if and only if for any* $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star), \mathfrak{F}^{\star\prime} = (\mathscr{A}^{\star\prime}, \mathscr{R}^{\star\prime})$ *in* $\mathfrak{K}_{\mathfrak{F}^\star}$ *such that* $\mathscr{A}^\star \cap \mathscr{A}^{\star\prime} = \emptyset$ *and every* $a, b \in \mathscr{A}^\star$ *we have* $a \geq^{\sigma^\star}_{\mathfrak{F}^\star} b$ *if and only if* $a \geq^{\sigma^\star}_{\mathfrak{F}^\star \oplus \mathfrak{F}^{\star\prime}} b$*.*

The Void precedence property states that non-attacked arguments should be ranked higher than attacked arguments.

Property 5.3 (Void precedence). *We say that* $\sigma^\star$ *satisfies* Void precedence *if and only if for any* $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ *and* $a, b \in \mathscr{A}^\star$ *such that* $\mathscr{R}^{\star}{}^{-}_{1}(a) = \emptyset$ *and* $\mathscr{R}^{\star}{}^{-}_{1}(b) \neq \emptyset$ *we have* $a >^{\sigma^\star}_{\mathfrak{F}^\star} b$*.*

The Self-contradiction property states that self-contradicting arguments should be ranked lower than non self-contradicting arguments.

Property 5.4 (Self-contradiction). *We say that a ranking-based semantics $\sigma^\star$ satisfies* Self-contradiction *if and only if for any $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and $a, b \in \mathscr{A}^\star$ such that there exists $S_1 \in \mathscr{R}^{\star -}_1(a)$ with $a \in S_1$ and there exists no $S_2 \in \mathscr{R}^{\star -}_1(b)$ with $b \in S_2$ we have $b \succ^{\sigma^\star}_{\mathfrak{F}^\star} a$.*

The Cardinality precedence property states that if an argument a has more attackers than an argument b than it should be ranked lower than b.

Property 5.5 (Cardinality precedence). *We say that $\sigma^\star$ satisfies* Cardinality precedence *if and only if for any $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and $a, b \in \mathscr{A}^\star$ such that $|\mathscr{R}^{\star -}_1(a)| > |\mathscr{R}^{\star -}_1(b)|$ we have $b \succ^{\sigma^\star}_{\mathfrak{F}^\star} a$.*

The Defense precedence property states that if a and b are two arguments with the same number of attackers and a is defended but b is not then a should be ranked higher than b.

Property 5.6 (Defense precedence). *We say that $\sigma^\star$ satisfies* Defense precedence *if and only if for any $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and $a, b \in \mathscr{A}^\star$ such that $|\mathscr{R}^{\star -}_1(a)| = |\mathscr{R}^{\star -}_1(b)|, \mathscr{R}^{\star -}_2(a) = \emptyset$ and $\mathscr{R}^{\star -}_2(b) \neq \emptyset$ we have $b \succ^{\sigma^\star}_{\mathfrak{F}^\star} a$.*

The Total property states that two arguments should always be comparable.

Property 5.7 (Total). *We say that $\sigma^\star$ satisfies* Total *if and only if for any $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and $a, b \in \mathscr{A}^\star$ we have $a \succeq^{\sigma^\star}_{\mathfrak{F}^\star} b$ or $b \succ^{\sigma^\star}_{\mathfrak{F}^\star} a$.*

The Non-attacked equivalence property states that two non-attacked arguments should be ranked equivalently.

Property 5.8 (Non-attacked equivalence). *We say that $\sigma^\star$ satisfies* Non-attacked equivalence *if and only if for any $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and $a, b \in \mathscr{A}^\star$ such that $\mathscr{R}^{\star -}_1(a) = \mathscr{R}^{\star -}_1(b) = \emptyset$ we have $a \approx^{\sigma^\star}_{\mathfrak{F}^\star} b$*

Definition 5.16 (Simple and distributed defense). *Let $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and $a \in \mathscr{A}^\star$. The defense of a is simple if and only if for every direct defender S of a, there is a unique $S' \in \mathscr{R}^{\star -}_1(a)$ and $s' \in S'$ such that $(S, s') \in \mathscr{R}^\star$. The defense of a is distributed if and only if for every direct attacker S of a, there is at most one direct defender S' of a such that (S', s) where $s \in S$.*

The Distributed-defense precedence property states that if a has a simple and distributed defense and b has a simple but not distributed defense than a should be higher than b.

Property 5.9 (Distributed-defense precedence). *$\sigma^\star$ satisfies* Distributed-defense precedence *if and only if for any $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$, for any $a, b \in \mathscr{A}^\star$ such that $|\mathscr{R}^{\star -}_1(a)| = |\mathscr{R}^{\star -}_1(b)|$ and $|\mathscr{R}^{\star -}_2(a)| = |\mathscr{R}^{\star -}_2(b)|$ and that the defense of a is simple and distributed and that the defense of b is simple but not distributed we have $a \succ^{\sigma^\star}_{\mathfrak{F}^\star} b$.*

An attack (respectively defense) branch of an argument a is an hypergraph argumentation framework such that its fusion with the original argumentation framework should result in the addition of an even (respectively odd) path to a.

Definition 5.17 (Added attack and defense branch). *Let $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ be an hypergraph argumentation framework and $a \in \mathscr{A}^\star$. $P_+(a)$ (respectively $P_-(a)$) is a pair $(\mathscr{A}^\star{}_0, \mathscr{R}^\star{}_0)$ called a defense (respectively attack) branch added to a if and only if the three following items are satisfied:*

- $\mathscr{A}^\star{}_0 = \{a, x_1, \ldots, x_k\}$ *and* $\mathscr{A}^\star \cap \mathscr{A}^\star{}_0 = \{a\}$
- *there exists* $n \in 2\mathbb{N}$ *(respectively* $n \in 2\mathbb{N}+1$*)*, $S_1 \in (2^{\mathscr{A}^\star{}_0} \setminus \emptyset)$ *and a path* $((S_1, t_1), \ldots, (S_n, t_n))$ *of size* n *such that* $t_n = a$.
- $\mathscr{R}^\star{}_0 = \{(S_i, t_i) \mid i \in \{1, \ldots, n\}\}$ *and* $(\bigcup_1^n S_i) \cup \{a\} = \mathscr{A}^\star{}_0$

The Strict addition of defense branch property states that adding a defense branch to an argument should increase its rank.

Property 5.10 (Strict addition of defense branch). *We say that $\sigma^\star$ satisfies* Strict addition of defense branch *if and only if for any $\mathfrak{F}^{\star\prime}, \mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and any $a \in \mathscr{A}^\star$ such that there exists an isomorphism γ with $\mathfrak{F}^{\star\prime} = \gamma(\mathfrak{F}^\star)$ we have $\gamma(a) \succ^{\sigma^\star}_{\mathfrak{F}^\star \oplus \mathfrak{F}^{\star\prime} \oplus P_+(\gamma(a))} a$.*

The Addition of defense branch property states that adding a defense branch to an attacked argument should increase its rank.

Property 5.11 (Addition of defense branch). *We say that $\sigma^\star$ satisfies* Addition of defense branch *if and only if for any $\mathfrak{F}^{\star\prime}, \mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and any $a \in \mathscr{A}^\star$ such that there exists an isomorphism γ with $\mathfrak{F}^{\star\prime} = \gamma(\mathfrak{F}^\star)$ and $\mathscr{R}^{\star -}_1(a) \neq \emptyset$ we have $\gamma(a) \succ^{\sigma^\star}_{\mathfrak{F}^\star \oplus \mathfrak{F}^{\star\prime} \oplus P_+(\gamma(a))} a$.*

The Addition of attack branch property states that adding an attack branch to an argument should decrease its rank.

Property 5.12 (Addition of attack branch). *We say that $\sigma^\star$ satisfies* Addition of attack branch *if and only if for any $\mathfrak{F}^{\star\prime}, \mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and any $a \in \mathscr{A}^\star$ such that there exists an isomorphism γ with $\mathfrak{F}^{\star\prime} = \gamma(\mathfrak{F}^\star)$ we have $a \succ^{\sigma^\star}_{\mathfrak{F}^\star \oplus \mathfrak{F}^{\star\prime} \oplus P_-(\gamma(a))} \gamma(a)$.*

The Increase of attack branch property states that increasing the length of an attack branch of an argument a should decrease the ranking of a.

Property 5.13 (Increase of attack branch). *We say that $\sigma^\star$ satisfies* Increase of attack branch *if and only if for any $\mathfrak{F}^{\star\prime}, \mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and any $a \in \mathscr{A}^\star$ such that there exists an isomorphism γ with $\mathfrak{F}^{\star\prime} = \gamma(\mathfrak{F}^\star)$, for every $S \in (2^{\mathscr{A}} \setminus \emptyset)$ such that for every $s \in S, \mathscr{R}^{\star -}_1(s) = \emptyset$, there exists a path of size n from S to a with $n \in 2\mathbb{N}+1$ and there is no path of size m from S to a with $m \in 2\mathbb{N}$ then for every $s \in S$, $\gamma(a) \succ^{\sigma^\star}_{\mathfrak{F}^\star \oplus \mathfrak{F}^{\star\prime} \oplus P_+(\gamma(s))} a$.*

The Increase of defense branch property states that increasing the length of a defense branch of an argument a should increase the ranking of a.

Property 5.14 (Increase of defense branch). *We say that $\sigma^\star$ satisfies* Increase of defense branch *if and only if for any $\mathfrak{F}^{\star\prime}, \mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and any $a \in \mathscr{A}^\star$ such that there exists an isomorphism γ with $\mathfrak{F}^{\star\prime} = \gamma(\mathfrak{F}^\star)$, for every $S \in (2^{\mathscr{A}} \setminus \emptyset)$ such that for every $s \in S, \mathscr{R}^{\star}{}_1^-(s) = \emptyset$, there exists a path of size n from S to a with $n \in 2\mathbb{N}$ and there is no path of size m from S to a with $m \in 2\mathbb{N}+1$ then for every $s \in S$, $a \succ^{\sigma^\star}_{\mathfrak{F}^\star \oplus \mathfrak{F}^{\star\prime} \oplus P_+(\gamma(s))} \gamma(a)$.*

Definition 5.18 (Cycle). *Let $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ be an hypergraph argumentation framework, $S \in (2^{\mathscr{A}^\star} \setminus \emptyset)$ and $s \in \mathscr{A}^\star$. We say that a path from S to s is a cycle if and only if $s \in S$. An argumentation framework is called acyclic if and only if there is no cycle.*

The Attack vs full defense property states that, in an acyclic hypergraph argumentation framework, an argument with only one direct attacker should be ranked lower than an argument without any attack branch.

Property 5.15 (Attack vs full defense). *We say that $\sigma^\star$ satisfies* Attack vs full defense *if and only if for any $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ that is acyclic and $a, b \in \mathscr{A}^\star$ such that $|\mathscr{R}^{\star}{}_1^-(b)| = 1, \mathscr{R}^{\star}{}_2^-(b) = \emptyset$, there exists no $S \in (2^{\mathscr{A}^\star} \setminus \emptyset)$ with for every $s \in S, \mathscr{R}^{\star}{}_1^-(s) = \emptyset$ and a path of size n from S to a with $n \in 2\mathbb{N}+1$ then $a \succ^{\sigma^\star}_{\mathfrak{F}^\star} b$.*

In the next properties, we make the assumption that the set of arguments should be compared with respect to their weakest arguments. We work under this hypothesis since, for an attack to exist, all arguments from S are necessary. Of course, another aggregation function could be used and most of the corresponding definitions could easily be changed to take this modification into account.

The Quality precedence property states that if an argument a has an attacker S such that the weakest element of S is ranked higher than every weakest element of any attacker of b than a should be ranked lower than b.

Property 5.16 (Quality precedence). *We say that $\sigma^\star$ satisfies* Quality precedence *if and only if for any $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and $a, b \in \mathscr{A}^\star$ such that there exists $S \in \mathscr{R}^{\star}{}_1^-(a)$ and for every $S' \in \mathscr{R}^{\star}{}_1^-(b)$ we have $\min^{\sigma^\star}_{\mathfrak{F}^\star}(S) \succ^{\sigma^\star}_{\mathfrak{F}^\star} \min^{\sigma^\star}_{\mathfrak{F}^\star}(S')$ then $b \succ^{\sigma^\star}_{\mathfrak{F}^\star} a$.*

Before introducing the next properties, we need to introduce a relation that compares sets of sets of arguments by inspiring ourselves from [Amgoud and Ben-Naim, 2013].

Definition 5.19 (Group comparison). *Let $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ and $G, G' \subseteq (2^{\mathscr{A}^\star} \setminus \emptyset)$. We say that $G \geq G'$ if and only if there exists an injective function $f : G' \to G$ such that for every $g' \in G', \min^{\sigma^\star}_{\mathfrak{F}^\star}(f(g')) \succeq^{\sigma^\star}_{\mathfrak{F}^\star} \min^{\sigma^\star}_{\mathfrak{F}^\star}(g')$.*

Notation 5.3. *As usual, we use the notation $G > G'$ if and only if $|G'| < |G|$ or there exists $g' \in G'$ such that $\min^{\sigma^\star}_{\mathfrak{F}^\star}(f(g')) >^{\sigma^\star}_{\mathfrak{F}^\star} \min^{\sigma^\star}_{\mathfrak{F}^\star}(g')$.*

The Counter-transitivity property states that if the attackers of a are better than the attackers of b with respect to the group comparison then b should be ranked higher than a.

Property 5.17 (Counter-transitivity). *We say that $\sigma^\star$ satisfies* Counter-transitivity *if and only if for any $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and $a, b \in \mathscr{A}^\star$ such that $\mathscr{R}^{\star\,-}_1(a) \geq \mathscr{R}^{\star\,-}_1(b)$ then $b \geq^{\sigma^\star}_{\mathfrak{F}^\star} a$.*

The Strict counter-transitivity property states that if the attackers of a are better than the attackers of b with respect to the group comparison then b should be ranked strictly higher than a.

Property 5.18 (Strict counter-transitivity). *We say that $\sigma^\star$ satisfies* Strict counter-transitivity *if and only if for any $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star) \in \mathfrak{K}_{\mathfrak{F}^\star}$ and $a, b \in \mathscr{A}^\star$ such that $\mathscr{R}^{\star\,-}_1(a) > \mathscr{R}^{\star\,-}_1(b)$ then $b >^{\sigma^\star}_{\mathfrak{F}^\star} a$.*

5.2.2 The nh-categoriser

As mentioned before, in each set of attacking arguments, all the components are necessary. Thus, removing one argument from the set of attacking arguments would make the attack void. In the definition of the nh-categoriser, we thus consider the force of the set of attacking arguments to be the force of the weakest argument of the set. Of course, the approach can be generalised with other aggregating methods.

Definition 5.20 (Nh-categoriser function). *Let $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ be an hypergraph argumentation framework. The nh-categoriser function is $C : \mathscr{A}^\star \to [0,1]$ defined as, for all $a \in \mathscr{A}^\star$:*

$$C(a) = \begin{cases} 1 & \text{if } \mathscr{R}^{\star\,-}(a) = \emptyset \\ \dfrac{1}{1 + \sum\limits_{S \in \mathscr{R}^{\star\,-}_1(a)} \min\limits_{s \in S} C(s)} & \text{otherwise} \end{cases}$$

In the rest of this section, we consider the argumentation framework $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ where $\mathscr{A}^\star = \{a_1, \ldots, a_n\}$. We now answer the two following questions for the nh-categoriser function: (1) "How many solutions exists?" and (2) "how to find them?"

We first transform the problem into a fixed point form. Let us consider $v \in [0,1]^n$ such that:

$$v = F(v) = [f_1(v), f_2(v), \ldots, f_n(v)]^T \tag{5.1}$$

where the function F maps $[0,1]^n$ to $[0,1]^n$, and for every $i \in \{0,\dots,n\}$, the function f_i from $[0,1]^n$ to $[0,1]$ is defined by the nh-categoriser function:

$$f_i(v) = \begin{cases} 1 & \text{if } \mathscr{R}^{\star -}(a_i) = \emptyset \\ \frac{1}{1+\sum\limits_{S \in \mathscr{R}^{\star}_{1}(a_i)} \min\limits_{a_j \in S} f_j(v)} & \text{else} \end{cases} \tag{5.2}$$

The function F is continuous and non-increasing as for every two vectors $u = (u_1,\dots,u_n)$ and $u' = (u'_1,\dots u'_n)$ of $[0,1]^n$ with $u_1 \leq u'_1,\dots,u_n \leq u'_n$, $F(u) \geq F(u')$ holds.

Proposition 5.13 (Solution existence). *For any argumentation framework $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$, the nh-categoriser valuation defined in (5.2) has at least one solution in $[0,1]^n$.*

Proof. The proof is similar to the one of Pu et al. [2014] and relies on the equivalence result that function F has at least one fixed point. The proof uses Brouwer's fixed point theorem and the fact that $[0,1]^n$ is homeomorphic to a closed ball and function that F is continuous on it. □

Proposition 5.14 (Uniqueness of nh-categoriser valuation). *Let $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ be an hypergraph argumentation framework with $\mathscr{A}^\star = \{a_1,\dots,a_n\}$ and $\mathscr{R}^\star$ with sets of attacking arguments. Then, the scores of the nh-categoriser function converge toward a unique solution $v^* \in [0,1]^n$, which is the limit of the sequence of $\{v^{(k)}\}_{k=0}^{\infty}$ defined from an arbitrary selected $v^{(0)}$ in $[0,1]^n$ and , for each $k \geq 1$, generated by:*

$$v^{(k)} = F(v^{(k-1)}) \tag{5.3}$$

Proof. First, let us consider that $u^{(0)} = (0,\dots,0), u^{(1)} = F(u^{(0)}) = (1,\dots,1)$ and $u^{(k)} = F(u^{(k-1)})$ for each $k \geq 2$. We can easily check that:

$$u^{(0)} \leq u^{(2)} \leq u^{(1)} \tag{5.4}$$

and that there exists $0 < \varphi < 1$ such that:

$$\varphi u^{(1)} \leq u^{(2)} \tag{5.5}$$

This is true because every element of $u^{(2)}$ is strictly positive and φ can be initialised to the minimum element.

Now, let us prove by induction that for all $k \geq 0$, the following statement holds:

$$u^{(0)} \leq u^{(2)} \leq \cdots \leq u^{(2k)} \leq \cdots \leq u^{(2k+1)} \leq \cdots \leq u^{(3)} \leq u^{(1)} \tag{5.6}$$

Base case: We showed in (5.4) that $u^{(0)} \leq u^{(2)} \leq u^{(1)}$.

Inductive step: Suppose that $u^{(0)} \leq u^{(2)} \leq \cdots \leq u^{(2k)} \leq \cdots \leq u^{(2k+1)} \leq \cdots \leq u^{(3)} \leq u^{(1)}$ is true. We need to prove that $u^{(2k)} \leq u^{(2k+2)} \leq u^{(2k+1)} \leq u^{(2k+1)}$.

First, we show that $u^{(2k+2)} \leq u^{(2k+1)}$. Since $u^{(2k)} \leq u^{(2k+1)}$ and that F is non increasing, we deduce that $F(u^{(2k+1)}) \leq F(u^{(2k)})$ and that $u^{(2k+2)} \leq u^{(2k+1)}$. Likewise, we show $u^{(2k)} \leq u^{(2k+2)}$ using the same reasoning.

Second, we show that $u^{(2k+2)} \leq u^{(2k+3)} \leq u^{(2k+1)}$. Since we prove in the previous step that $u^{(2k+2)} \leq u^{(2k+1)}$ and $u^{(2k)} \leq u^{(2k+2)}$, we deduce using the fact that F is non increasing that $u^{(2k+2)} \leq u^{(2k+3)}$ and $u^{(2k+3)} \leq u^{(2k+1)}$ respectively. This concludes the proof by induction.

From (5.5) and (5.6), we find that there exists φ such that $\varphi u^{(2k-1)} \leq u^{(2k)}$ for each $k \geq 1$. Now, let us denote $\pi_k = \sup\{\pi \text{ such that } \pi u^{(2k-1)} \leq u^{2k)}\}$. Then, $0 < \varphi \leq \pi_1 \leq \cdots \leq \pi_k \leq \cdots \leq 1$. We now show that $\lim\limits_{k\to\infty} \pi_k = 1$.

We first show $f_i(\pi u) = \frac{1}{\pi + f_i(u)(1-\pi)} f_i(u)$ for all $i \in \{1, 2, \ldots, n\}$.

$$
\begin{aligned}
f_i(\pi u) &= \frac{1}{1 + \sum\limits_{S \in \mathscr{R}^{\star -}_1(a_i)} \min\limits_{a_j \in S} f_j(\pi u)} \\
&= \frac{1}{1 + \pi \sum\limits_{S \in \mathscr{R}^{\star -}_1(a_i)} \min\limits_{a_j \in S} f_j(u)} \\
&= \frac{1 + \sum\limits_{S \in \mathscr{R}^{\star -}_1(a_i)} \min\limits_{a_j \in S} f_j(u)}{1 + \pi \sum\limits_{S \in \mathscr{R}^{\star -}_1(a_i)} \min\limits_{a_j \in S} f_j(u)} \times f_i(u) \\
&= \frac{\frac{1}{1+\pi \sum\limits_{S \in \mathscr{R}^{\star -}_1(a_i)} \min\limits_{a_j \in S} f_j(u)}}{\frac{1+ \sum\limits_{S \in \mathscr{R}^{\star -}_1(a_i)} \min\limits_{a_j \in S} f_j(u)}{}} \times f_i(u) \\
&= \frac{\frac{1}{(1+ \sum\limits_{S \in \mathscr{R}^{\star -}_1(a_i)} \min\limits_{a_j \in S} f_j(u))\pi - \pi + 1}}{\frac{1+ \sum\limits_{S \in \mathscr{R}^{\star -}_1(a_i)} \min\limits_{a_j \in S} f_j(u)}{}} \times f_i(u) \\
&= \frac{1}{\pi + \frac{1-\pi}{1+ \sum\limits_{S \in \mathscr{R}^{\star -}_1(a_i)} \min\limits_{a_j \in S} f_j(u)}} \times f_i(u) \\
f_i(\pi u) &= \frac{1}{\pi + (1-\pi) f_i(u)} \times f_i(u)
\end{aligned}
$$

Then, there exists $0 < \alpha < 1$ and a continuous function $\psi(\pi) = \frac{1}{\pi + \alpha(1-\pi)}$

such that:

$$F(\pi u) \leq \psi(\pi)F(u), \forall \pi \in [\varphi, 1[, u \in [\varphi, 1]^n \tag{5.7}$$

Then, we show that:

$$u^{(2k+1)} = F(u^{(2k)}) \leq F(\pi_k u^{(2k-1)}) \leq \psi(\pi_k)u^{(2k)} \leq \psi(\pi_k)u^{(2k+2)} \tag{5.8}$$

By definition of π_k, it holds that $\pi_k u^{(2k-1)} \leq u^{(2k)}$ and since F is non decreasing, we deduce that $F(u^{(2k)}) \leq F(\pi_k u^{(2k-1)})$. Using (5.7), we conclude that $F(\pi_k u^{(2k-1)}) \leq \pi_k F(u^{(2k-1)}) = \pi_k u^{(2k)}$. Using (5.6), we have that $u^{(2k)} \leq u^{(2k+2)}$ and $\pi_k u^{(2k)} \leq \pi_k u^{(2k+2)}$.

Now, we show that $\pi_{k+1} \geq \frac{1}{\psi(\pi_k)}$ by contradiction. Suppose that this $\pi_{k+1} < \frac{1}{\psi(\pi_k)}$. Using (5.8), we have that $u^{(2k+1)} \leq \psi(\pi_k)u^{(2k+2)}$ and thus $\frac{1}{\psi(\pi_k)}u^{(2k+1)} \leq u^{(2k+2)}$. Contradiction with the definition of π_{k+1}.

We now show that:

$$1 - \pi_{k+1} \leq (1-\alpha)(1-\pi_k) \leq \cdots \leq (1-\alpha)^k(1-\pi_1) \leq (1-\alpha)^k(1-\varphi) \tag{5.9}$$

Using the previous result, we have that:

$$\begin{aligned} \pi_{k+1} \geq \frac{1}{\psi(\pi_k)} &= \pi_k + \alpha(1-\pi_k) \\ \alpha\pi_k - \alpha - \pi_k + 1 &\geq 1 - \pi_{k+1} \\ (1-\alpha)(1-\pi_k) &\geq 1 - \pi_{k+1} \end{aligned}$$

This is sufficient to prove (5.9) by noticing that $\varphi \leq \pi_1$.

As $0 < \alpha < 1$, thus by (5.9) we have:

$$\lim_{k\to\infty}(1-\pi_{k+1}) = 0 \quad \Rightarrow \quad \lim_{k\to\infty}\pi_k = 1 \tag{5.10}$$

There by (5.6), we get for any integer $p \geq 1$:

$$0 \leq u^{(2k+2p)} - u^{(2k)} \leq u^{(2k+1)} - u^{(2k)} \leq (1-\pi_k)u^{(2k+1)} \leq (1-\pi_k)u^{(1)} \tag{5.11}$$

Since $[0,1]^n$ is normal, both $\{u^{(2k+1)}\}_{k=0}^{\infty}$ and $\{u^{(2k)}\}_{k=1}^{\infty}$ are convergence sequences. By (5.10) and (5.11), thus, there exists $u^* \in [0,1]^n$ such that:

$$\lim_{k\to\infty} u^{(2k+1)} = \lim_{k\to\infty} u^{(2k)} = u^* \tag{5.12}$$

Hence, $u^{(2k)} \leq u^* \leq u^{(2k+1)}$ and $u^{(2k)} \leq F(u^*) \leq u^{(2k+1)}$. Letting $k \to \infty$ and combining with (5.12), it follows that $F(u^*) = u^*$ (it is a fixed point of

F. We now show that the result holds for any arbitrary $v^{(0)} \in [0,1]^n$. By induction, we have that for any $k \geq 1, u^{(2k)} \leq v^{(2k)} \leq u^{(2k-1)}$ and $u^{(2k)} \leq v^{(2k+1)} \leq u^{(2k+1)}$. Then $v^{(k)} \to v^* = u^*$ as $k \to \infty$. In particular, let $v^{(0)} = w^*$, where w^* is any fixed point of F in $[0,1]^n$, then $v^{(k)} = w^*$ for all $k \geq 1$, and we get $w^* = u^*$. So F has a unique fixed point in $[0,1]^n$.

□

Now that we showed the existence and uniqueness of the values returned by the nh-categoriser function for any argumentation hypergraphs, we show how the nh-categoriser ranking-based semantics is constructed from the scores returned by nh-categoriser function.

Definition 5.21 (Nh-categoriser ranking-based semantics). *Let $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ be an argumentation hypergraph. The nh-categoriser ranking-based semantics on $\mathfrak{F}^\star$ returns a ranking $\geq^{nh}_{\mathfrak{F}^\star}$ on $\mathscr{A}^\star$ such that for every $a, b \in \mathscr{A}^\star$, $a \geq^{nh}_{\mathfrak{F}^\star} b$ if and only if $C(a) \geq C(b)$.*

Example 5.9 (Example 5.7 cont'd). *The nh-categoriser scores of arguments are $C(a) \approx 0.38, C(b) = 1, C(c) = 0.5, C(d) \approx 0.65$ and $C(e) \approx 0.53$. We obtain the ranking: $b \geq^{nh}_{\mathfrak{F}^\star} d \geq^{nh}_{\mathfrak{F}^\star} e \geq^{nh}_{\mathfrak{F}^\star} c \geq^{nh}_{\mathfrak{F}^\star} a$.*

We now show that nh-categoriser satisfies the same properties as h-categoriser.

Proposition 5.15 (Property satisfaction). *The nh-categoriser satisfies Abstraction, Independence, Void precedence, Defense precedence, Counter-transitivity, Strict counter-transitivity, Increase of attack branch, Increase of defense branch, Addition of attack branch, Total and Non-attacked equivalence.*

In the next proposition, we show that there always exists a regular Dung's abstract argumentation framework such that the score on arguments with the h-categoriser function (see Definition 2.54 on page 41) are the same as the scores on arguments in the hypergraph argumentation framework with the nh-categoriser function.

Proposition 5.16 (Dung equivalent existence). *Let $\mathfrak{F} = (\mathscr{A}^\star, \mathscr{R}^\star)$ be an hypergraph argumentation frameworks. Then, there exists an abstract argumentation framework $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$, a subset $\{a_1, \ldots, a_m\} \subseteq \mathscr{A}$ and a bijection $\gamma : \mathscr{A}^\star \to \{a_1, \ldots, a_m\}$ such that for every $a \in \mathscr{A}^\star$, $C(a) = C'(\gamma(a))$.*

Proof. The detailed proof can be found in Section 7.2.3 on page xv. □

Example 5.10 (Example 5.9 cont'd). *Let us consider the abstract argumentation framework $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ with $\mathscr{A} = \{a', b', c', d', e'\} \cup \{d_a, b_a, ac_e, b_c, be_d, dc_e\}$ and $\mathscr{R} = \{(d_a, a), (d_a, ac_e), (b_a, a), (b_a, ac_e), (b_c, c), (b_c, dc_e), (be_d, d),$*

(be_d, d_a), $(dc_e, e), (dc_e, be_d), (ac_e, e), (ac_e, be_d)\}$ *depicted in Figure 5.4. If* γ *is the bijection from* $\mathscr{A}^\star$ *to* $\mathscr{A}$ *with* $\gamma(a) = a', \gamma(b) = b', \gamma(c) = c', \gamma(d) = d'$ *and* $\gamma(e) = e'$ *then it holds that for every* $a \in \mathscr{A}^\star$, $C(a) = C'(\gamma(a))$.

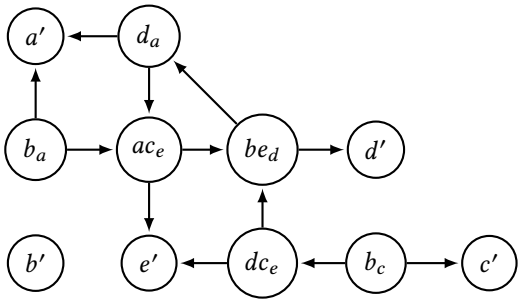

Figure 5.4: Argumentation framework with equal h-categoriser scores

We would like to point out that for a particular hypergraph argumentation framework, there might be several binary argumentation frameworks such that Proposition 5.16 is satisfied. For instance, in Example 5.11, we show another binary argumentation framework that satisfies Proposition 5.16.

Example 5.11 (Example 5.10 cont'd). *Let us consider the binary argumentation framework* $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ *with* $\mathscr{A} = \{a', b', c', d', e'\}$ *and* $\mathscr{R} = \{(d', a'), (b', a'), (e', d'), (a', e'), (b', c'), (c', e')\}$, *depicted in Figure 5.5 and the bijection* γ *from* $\mathscr{A}^\star$ *to* $\mathscr{A}$ *such that* $\gamma(a) = a', \gamma(b) = b', \gamma(c) = c', \gamma(d) = d'$ *and* $\gamma(e) = e'$. *It also holds that for every* $a \in \mathscr{A}^\star$, $C(a) = C'(\gamma(a))$.

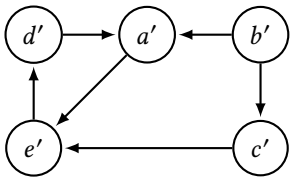

Figure 5.5: Another argumentation framework with equal h-categoriser scores

5.3 Ranking facts in inconsistent knowledge bases

We now place ourselves in the Ontology-Based Data Access (OBDA) setting where one wants to "access" the data stored in different source. In this setting, the aggregation of all the databases often result in an inconsistent knowledge base due to conflicting data or incompatible vocabulary. As a result, the main research avenue is to investigate query answering over a set of fact bases enriched by the ontology [Poggi et al., 2008]. One of the main challenges of reasoning in OBDA applications is handling the inherent inconsistency that might occur amongst independently built data sources partially describing the same knowledge of interest [Benferhat et al., 1997; Lukasiewicz et al., 2015; Lembo et al., 2015; Hecham et al., 2017b]. Classically inconsistent tolerant semantics consider all maximally consistent subsets of a fact base (called repairs) that they manipulate using a modifier (expansion, splitting, etc.) and an inference strategy (intersection, universality, etc.) [Baget et al., 2016a].

Using all repairs might be inappropriate for certain applications that would rather focus on particular sources of knowledge. For instance, when considering more reliable knowledge (i.e. sensor information, provenance data etc.) one could only consider repairs using mostly facts from such sources. Preferences on facts have been used for inconsistency-tolerant reasoning in the work of Staworko et al. [2012]. In that setting, the authors suppose that the preference order on the facts is given but unfortunately, this is not always the case. In such cases, we propose to use the inconsistency of the elements of the knowledge base as an intrinsic preference on the facts. Such inherent preference on the facts (i.e. facts that are more or less responsible for the inconsistencies) generates a preference on the repairs that are containing these facts (i.e. repairs that contain more or less controversial facts). In this section, we propose a framework that takes into consideration the inconsistency on the facts when using the repairs for query answering and restricts the set of repairs to the "best" with respect to inconsistency values. Since we consider a subset of repairs we obtain more answers than classical inconsistency-tolerant query answering.

In this section, we characterise desirable properties of such frameworks like free facts entailment, syntax independence and reliability preservation. We also provide an implementation of our approach and discuss its performance. The salient point of this section lies in it being the *first approach in the literature capable of ranking repairs using only the inherent structure of the knowledge base*. This is a significant result as our approach is applicable on a large variety of domains without requiring additional preference information. Furthermore, we show the significance and the practical interest of our approach using the real data collected in the framework of the Pack4Fresh project for reducing food wastes. During this project, we collected data using an online poll from a set of professionals of the food in-

dustry, including wholesalers, quality managers, floorwalkers and warehouse managers, about food packagings and their characteristics. The framework was able to rank the repairs efficiently and the results were then analysed and evaluated by experts from the packaging industry.

5.3.1 The ranking-based inference framework

We introduce the Ranking-based Inference Framework (RIF) and its three main components: the inconsistency value, the lifting function and the inconsistency-tolerant inference. It is similar to the work in argumentation by Konieczny et al. [2015], where only the best extensions are used for reasoning. The section is organised as follows: in Section 5.3.1.1, we recall the notion of Drastic and MI Shapley inconsistency values, in Section 5.3.1.2, we give examples of lifting functions and in Section 5.3.1.3, we show how inconsistency-tolerant inferences are modified in order to be used in the framework.

An *inconsistency measure* according to Grant and Hunter [2011] is a function that, given a knowledge base $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$, associates a number to each set of facts.

Definition 5.22 (Inconsistency measure). *An inconsistency measure is a function $I : \mathcal{KB}_s \times 2^{\mathcal{L}} \to \mathbb{R}$ such that for every $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N}) \in \mathcal{KB}_s$ and $C, C' \in 2^{\mathcal{L}}$:*

- *$I(\mathcal{KB}, C) = 0$ if and only if C is $\mathcal{R}$-consistent*
- *$I(\mathcal{KB}, C \cup C') \geq I(C)$*
- *if α is a free fact of $\mathcal{KB}$, then $I(\mathcal{KB}, \mathcal{F}) = I(\mathcal{KB}, \mathcal{F} \backslash \{\alpha\})$.*

For readability purposes, we will use the notation $I^{\mathcal{KB}}(C)$ instead of $I(\mathcal{KB}, C)$ and $I(C)$ if the working knowledge base $\mathcal{KB}$ is obvious.

An *inconsistency value* is a function that associates a number to each fact of a knowledge base $\mathcal{KB}$. Many inconsistency values were defined by Hunter and Konieczny [2010] using existing inconsistency measures and the Shapley value from coalitional game theory. We introduce a framework that makes use of these inconsistency values together with a lifting function and an inconsistency-tolerant inference relation to improve the productivity of query answering for an inconsistent knowledge base.

Our framework is based on three layers. First, an *inconsistency value* is used to calculate the score of each fact of $\mathcal{KB}$. We previously mentioned Shapley inconsistency values, but any function returning a score for each fact of $\mathcal{KB}$ can be used.

Definition 5.23 (Inconsistency value). *An inconsistency value is a function* $\mathfrak{V} : \mathcal{KB}_s \times \mathcal{L} \to \mathbb{R}$. *Let* $\geq^{\mathfrak{V}}_{\mathcal{L}}$ *be the total, reflexive and transitive binary order on* $\mathcal{L}$ *with respect to* $\mathcal{KB}$ *and* $\mathfrak{V}$ *defined as: for every* $a, b \in \mathcal{L}, a \geq^{\mathfrak{V}}_{\mathcal{L}} b$ *if and only if* $\mathfrak{V}(\mathcal{KB}, a) \leq \mathfrak{V}(\mathcal{KB}, b)$.

For readability purposes, we write $\mathfrak{V}_a(\mathcal{KB})$ *instead of* $\mathfrak{V}(\mathcal{KB}, a)$. *Moreover, we write* $\geq$ *instead of* $\geq^{\mathfrak{V}}_{\mathcal{L}}$ *when* $\mathcal{L}$ *and* $\mathfrak{V}$ *are obvious.*

Second, we need a *lifting function*, i.e. a function that compares the set of repairs, based on the individual scores of facts with respect to an inconsistency value. A criterion of comparison would be to evaluate the "strongest" fact of each set. A generalisation of this criterion is the so-called *leximax* which, in the case where the best facts are equally strong, proceeds to compare the next best fact of each set. Please note that the set of all total, reflexive and transitive binary orders on X is denoted by $\geq_X$.

Definition 5.24 (Lifting function). *A lifting function is a function* $\mathtt{L} : 2^{\mathcal{L}} \times \geq_{\mathcal{L}} \to \geq_{2^{\mathcal{L}}}$.

For readability purposes, we use the notation $\mathtt{L}^{\geq}(X)$ *for* $\mathtt{L}(X, \geq)$. *Furthermore,* $(E, E') \in \mathtt{L}^{\geq}(X)$ *means that* E *is better than or equal to* E'.

Third, we use an *inconsistency-tolerant inference relation* restricted to the best repairs sets ranked by the lifting function to answer the query. At this step, one can use the usual inconsistency-tolerant inference relations such as AR, IAR, ICR or any of the modifier-based semantics of Baget et al. [2016a]

Definition 5.25 (Inference). *An inconsistency-tolerant inference relation is a function* $\models : \mathcal{KB}_s \times \mathcal{Q} \to \{True, False\}$.

Based on the previous notions, we define our framework (see Figure 5.6).

Definition 5.26 (RIF). *A* ranking-based inference framework *(RIF) is a tuple* $\mathcal{RIF} = (\mathfrak{V}, \mathtt{L}, \models)$ *where* $\mathfrak{V}$ *is an inconsistency value,* $\mathtt{L}$ *is a lifting function and* $\models$ *is an inconsistency-tolerant inference. The* top result *of* $\mathcal{RIF} = (\mathfrak{V}, \mathtt{L}, \models)$ *on a knowledge base* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *is* $OUT_{\mathcal{RIF}}(\mathcal{KB}) = \{E \in repairs(\mathcal{KB}) \mid for\ all\ E' \in repairs(\mathcal{KB}), (E, E') \in \mathtt{L}^{\geq^{\mathfrak{V}}_{\mathcal{F}}}(\mathcal{F})\}$.

5.3.1.1 $\mathcal{RIF}$ inconsistency value

An inconsistency value is a function that associates a value to each fact of the knowledge base. This value is supposed to be higher the more a fact is conflicting with the other facts. In this thesis, we make the choice to focus on the Shapley inconsistency value introduced by Hunter and Konieczny [2010] because it possesses many desirable properties as will be shown in Proposition 5.17 below. The Shapley inconsistency value uses notions from game theory to measure the responsibility of each fact to the overall inconsistency of the knowledge base.

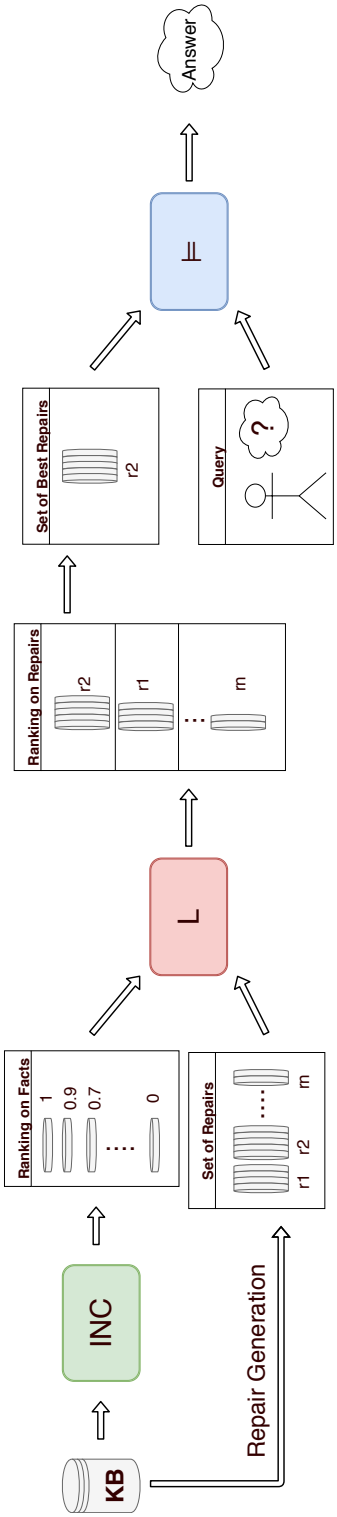

Figure 5.6: Representation of the $\mathcal{RIF}$ workflow

Definition 5.27 (Shapley inconsistency value). *Let I be an inconsistency measure, $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ a knowledge base and $f \in \mathcal{F}$, the Shapley inconsistency value corresponding to I, noted $\mathfrak{V}^I$ is defined as:*

$$\mathfrak{V}^I_f(\mathcal{KB}) = \sum_{C \subseteq \mathcal{F}} \frac{(|C|-1)!(|\mathcal{F}|-|C|)!}{|\mathcal{F}|!}(I^{\mathcal{KB}}(C) - I^{\mathcal{KB}}(C \backslash \{f\}))$$

Note that if one considers $\mathcal{F}$ as the vector $(f_1, f_2, \ldots, f_n)$, then $\mathfrak{V}^I(\mathcal{KB})$ is the vector of corresponding Shapley inconsistency values, i.e. $\mathfrak{V}^I(\mathcal{KB}) = (\mathfrak{V}^I_{f_1}(\mathcal{KB}), \mathfrak{V}^I_{f_2}(\mathcal{KB}), \ldots, \mathfrak{V}^I_{f_n}(\mathcal{KB}))$.

Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, the inconsistency values investigated in this thesis are the following:

- *The drastic Shapley inconsistency value* is computed by using the following inconsistency measure:

$$I^{\mathcal{KB}}_d(X) = \begin{cases} 0 & \text{if } X \text{ is } \mathcal{R}\text{-consistent with respect to } \mathcal{KB} \\ 1 & \text{otherwise} \end{cases}$$

- *The MI Shapley inconsistency value* is computed by using the following inconsistency measure:

$$I^{\mathcal{KB}}_{MI}(X) = |MI((X, \mathcal{R}, \mathcal{N}))|$$

We now show that every Shapley inconsistency value satisfies Distribution, Symmetry and Minimality. The result and its proof are similar to that of Hunter and Konieczny [2010].

Proposition 5.17 (Shapley I.V. property satisfaction). *Let I be an arbitrary inconsistency measure and $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ a knowledge base, the Shapley inconsistency value corresponding to I satisfies:*

- ***(Distribution)*** $\sum_{a \in \mathcal{F}} \mathfrak{V}^I_a(\mathcal{KB}) = I^{\mathcal{KB}}(\mathcal{F})$
- ***(Symmetry)*** *If $a, b \in \mathcal{F}$ such that for all $X' \subseteq \mathcal{F}, a, b \notin X'$ we have $I^{\mathcal{KB}}(X' \cup \{a\}) = I^{\mathcal{KB}}(X' \cup \{b\})$ then it holds that $\mathfrak{V}^I_a(\mathcal{KB}) = \mathfrak{V}^I_b(\mathcal{KB})$*
- ***(Minimality)*** *If a is a free fact of $\mathcal{KB}$ then $\mathfrak{V}^I_a(\mathcal{KB}) = 0$*

In Example 5.12, we show how the MI and drastic Shapley inconsistent values are computed from a simple knowledge base.

Example 5.12 (MI and drastic Shapley inconsistency values). *Let us consider the knowledge base* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *where* $\mathcal{F} = \{d(m), a(m), c(m), b(m,s)\}$, $\mathcal{R} = \{\forall X(c(X) \wedge b(X,s) \rightarrow u(X))\}$ *and* $\mathcal{N} = \{\forall X\ (d(X) \wedge a(X) \rightarrow \bot), \forall X\ (u(X) \wedge d(X) \rightarrow \bot), \forall X\ (u(X) \wedge a(X) \rightarrow \bot)\}$.

We have that $\mathfrak{V}^{I_d}_{d(m)}(\mathcal{KB}) = 4 \times \frac{1}{12} = \frac{1}{3}$ *and* $\mathfrak{V}^{I_{MI}}_{d(m)}(\mathcal{KB}) = 4 \times \frac{1}{12} + \frac{1}{4} \times 2 = \frac{5}{6}$. *Thus, here we have that* $\mathcal{F} = \{d(m), a(m), c(m), b(m,s)\}$ *and* $\mathfrak{V}^{I_d}(\mathcal{KB}) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{6}, \frac{1}{6})$ *and* $\mathfrak{V}^{I_{MI}}(\mathcal{KB}) = (\frac{5}{6}, \frac{5}{6}, \frac{2}{3}, \frac{2}{3})$. *Since a higher score means being more inconsistent, the resulting ranking on facts, for both inconsistency values, is* $c(m) \sim b(m,s) \succ d(m) \sim a(m)$.

We recall that we work with the total, reflexive and transitive ranking $\succeq^{\mathfrak{V}}_{\mathcal{F}}$ on $\mathcal{F}$ extracted from the inconsistency value.

5.3.1.2 $\mathcal{RIF}$ lifting

A *lifting function* $\mathtt{L}^{\succeq}$ compares sets of elements with respect to the ranking $\succeq$ and returns a total order on the sets.

Let us first introduce the `sort` relation that will be used in order to define the $\mathtt{L}^{\succeq}_{leximax}$ notion below. Given a set of elements $X = \{x_1, x_2, \ldots, x_n\}$ and a total, reflexive and transitive binary relation $\succeq$ on X, $\mathtt{sort}(X, \succeq)$ returns a sorted vector $(x_1, x_2, \ldots, x_n)$ such that for every x_i, x_j, we have that $x_i \succeq x_j$ if and only if $i \leq j$. The element at position i in the vector $\mathtt{sort}(X, \succeq)$ is denoted by $\mathtt{sort}_i(X, \succeq)$. Note that the returned vector is not necessarily unique due to the fact that some elements might be equivalent, i.e. $x_i \sim x_j$.

In this thesis, we consider two possible instantiations of the lifting function $\mathtt{L}$. The $\mathtt{L}^{\succeq}_{max}$ lifting function compares the subsets with respect to their maximal elements and $\mathtt{L}^{\succeq}_{leximax}$ compares the elements after sorting them in decreasing order.

Let Y be a set of elements, $\succeq$ be a ranking on Y, $E, E' \in 2^Y$, $\mathtt{sort}(E, \succeq) = (x_1, x_2, \ldots, x_n)$ and $\mathtt{sort}(E', \succeq) = (x'_1, x'_2, \ldots, x'_m)$. We say that:

- $(E, E') \in \mathtt{L}^{\succeq}_{max}(Y)$ if and only if $max(E) \succeq max(E')$, where $max(X) = \mathtt{sort}_1(X, \succeq)$.

- $(E, E') \in \mathtt{L}^{\succeq}_{leximax}(Y)$ if and only if one of the following holds: (1) $m = n$ and for every $i \in \{1, \ldots, n\}, x_i \sim x'_i$, (2) there exists $i \in \{1, \ldots, min(m,n)\}$ such that $x_i \succ x'_i$ and for every $j \in \{1, \ldots, i-1\}, x_j \sim x'_j$ or (3) $n > m$ and for every $i \in \{1, \ldots, m\}$, $x_i \sim x'_i$.

Example 5.13 (Example 5.12 cont'd). *Let* $\mathcal{RIF} = (\mathfrak{V}^{Id}, \mathtt{L}^{\succeq}_{leximax}, \models)$ *be a RIF. It holds that for every* $R \in repairs(\mathcal{KB}) \setminus \{c(m), b(m,s)\}$, *we have* $(\{c(m), b(m,s)\}, R) \in \mathtt{L}^{\succeq}_{leximax}(\mathcal{F})$ *but* $(R, \{c(m), b(m,s)\}) \notin \mathtt{L}^{\succeq}_{leximax}(\mathcal{F})$ *and thus,* $OUT_{\mathcal{RIF}}(\mathcal{KB}) = \{\{c(m), b(m,s)\}\}$

5.3.1.3 $\mathcal{RIF}$ inference

Inconsistency-tolerant query answering is a challenging problem that received a lot of attention recently. We recall that we place ourselves in the context of OBDA, where the ontology is assumed to be satisfiable and fully reliable. In the following, we recall some of the most well-known inconsistency-tolerant inferences that have been proposed in the literature. Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base and q be a boolean conjunctive query. Then:

- q is said to be AR entailed by $\mathcal{KB}$ denoted by $\mathcal{KB} \models_{AR} q$ if and only if for every $R \in repairs(\mathcal{KB}), \mathcal{C}\ell_{\mathcal{R}}(R) \models q$
- q is said to be IAR entailed by $\mathcal{KB}$ denoted by $\mathcal{KB} \models_{IAR} q$ if and only if $\mathcal{C}\ell_{\mathcal{R}}\left(\bigcap_{R \in repairs(\mathcal{KB})} R\right) \models q$
- q is said to be ICR entailed by $\mathcal{KB}$ denoted by $\mathcal{KB} \models_{ICR} q$ if and only if $\bigcap_{R \in repairs(\mathcal{KB})} \mathcal{C}\ell_{\mathcal{R}}(R) \models q$

Example 5.14 (Example 5.12 cont'd). *A query $q = \exists x(c(x))$ is not AR, IAR nor ICR entailed. Indeed, we cannot entail q from the closure of all the repairs, the intersection of the closure of all the repairs nor the closure of the intersection of all repairs.*

We propose here to reuse AR, IAR, ICR by restricting them to the top result of a $\mathcal{RIF}$ instead of the whole set of repairs.

Definition 5.28 (Restricted inference). *Let $x \in \{AR, IAR, ICR\}$. We denote the restriction of $\models_x$ to the top result of $\mathcal{RIF}$ instead of the whole set of repairs by $\models_x^{\mathcal{RIF}}$.*

For instance, the restricted version of AR will be denoted by $\models_{AR}^{\mathcal{RIF}}$ and defined as $\mathcal{KB} \models_{AR}^{\mathcal{RIF}} q$ if and only if for every $R \in \mathrm{OUT}_{\mathcal{RIF}}(\mathcal{KB}), \mathcal{C}\ell_{\mathcal{R}}(R) \models q$.

Example 5.15 (Example 5.13 cont'd). *Let us consider the query $q = \exists X(c(X))$. The query q is AR, IAR and ICR entailed with respect to $\mathcal{RIF}$ since $OUT_{\mathcal{RIF}}(\mathcal{KB}) = \{\{c(m), b(m,s)\}\}$.*

5.3.2 $\mathcal{RIF}$ results

This section presents a characterisation of the framework in terms of properties and general productivity results. In Section 5.3.2.1, we show some desirable properties of the framework and how such component properties relate to framework properties. In Section 5.3.2.2, we show an algorithm for computing the top result of the framework and its performance on a given set of data. In Section 5.3.2.3, we explicit the use of our framework on a real life scenario.

5.3.2.1 $\mathcal{RIF}$ properties

In this section, we show that the desirable properties on the components can lead to desirable properties on the entire framework. First, we introduce desirable properties for **general inconsistency values**. The *minimality* property states that a free fact should have the lowest score. The *flawed* property conveys the idea that a non-free fact should have a strictly positive score. Lastly, the *bottom facts* property states that an $\mathcal{R}$-inconsistent fact should have the score 1.

Definition 5.29 (Minimality). *We say that an inconsistency value $\mathfrak{V}$ satisfies* minimality *if and only if for any knowledge base $\mathcal{KB}$ and every free fact a of $\mathcal{KB}$, it holds that $\mathfrak{V}_a(\mathcal{KB}) = 0$.*

Definition 5.30 (Flawed). *We say that an inconsistency value $\mathfrak{V}$ satisfies* flawed *if and only if for any knowledge base $\mathcal{KB}$ and every non free fact a of $\mathcal{KB}$, it holds that $\mathfrak{V}_a(\mathcal{KB}) > 0$.*

Definition 5.31 (Bottom facts). *We say that an inconsistency value $\mathfrak{V}$ satisfies* bottom facts *if and only if for any knowledge base $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ and $a \in \mathcal{F}$ such that $\{a\}$ is $\mathcal{R}$-inconsistent, it holds that $\mathfrak{V}_a(\mathcal{KB}) = 1$*

The *R-append* and *N-append* properties are satisfied if the addition of a rule or a negative constraint to a knowledge base cannot decrease the score of any fact.

Definition 5.32 (R-append). *We say that an inconsistency value $\mathfrak{V}$ satisfies* R-append *if and only if for any knowledge base $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ and any positive rule $r \notin \mathcal{R}$ such that $\mathcal{KB}' = (\mathcal{F}, \mathcal{R} \cup \{r\}, \mathcal{N})$, it holds that for every $f \in \mathcal{F}, \mathfrak{V}_f(\mathcal{KB}) \leq \mathfrak{V}_f(\mathcal{KB}')$.*

Definition 5.33 (N-append). *We say that an inconsistency value $\mathfrak{V}$ satisfies* N-append *if and only if for any knowledge base $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ and any negative constraint $n \notin \mathcal{N}$ such that $\mathcal{KB}' = (\mathcal{F}, \mathcal{R}, \mathcal{N} \cup \{n\})$, it holds that for every $f \in \mathcal{F}, \mathfrak{V}_f(\mathcal{KB}) \leq \mathfrak{V}_f(\mathcal{KB}')$.*

The *abstraction-I* property states that an inconsistency value should not rely on the names of constants or predicates.

Definition 5.34 (Abstraction-I). *We say that an inconsistency value $\mathfrak{V}$ satisfies* abstraction-I *if and only if for any two knowledge bases $\mathcal{KB}, \mathcal{KB}'$ and any isomorphism[3] γ such that $\gamma(\mathcal{KB}) = \mathcal{KB}'$, it holds that for every $f \in \mathcal{F}$, $\mathfrak{V}_f(\mathcal{KB}) = \mathfrak{V}_{\gamma(f)}(\mathcal{KB}')$.*

[3]The isomorphism γ renames the predicates and the constants. We use an abuse of notation and apply γ to sets of facts, sets of rules, negative constraints and to knowledge bases (meaning we apply to all three).

The *cardinality-MI* property says that the score is based on minimal inconsistent subsets. Namely, if the score of a fact is strictly inferior to the score of an other fact, it means that the number of minimal inconsistent sets the first fact belongs to is strictly lower than the number of minimal inconsistent sets the second facts belongs to.

Definition 5.35 (Cardinality-MI). *We say that an inconsistency value* $\mathfrak{V}$ *satisfies* cardinality-MI *if and only if for any knowledge base* $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ *and any two facts* $f, f' \in \mathcal{F}$ *such that* $\mathfrak{V}_f(\mathcal{KB}) < \mathfrak{V}_{f'}(\mathcal{KB})$*, it holds that* $|\{X \in MI(\mathcal{KB}) \mid f \in X\}| < |\{X \in MI(\mathcal{KB}) \mid f' \in X\}|$.

Proposition 5.18 (MI and drastic I.V. property satisfaction). *It holds that:*

- $\mathfrak{V}^{I_{MI}}$ ***satisfies*** *minimality, flawed, abstraction-I, bottom facts and* ***does not satisfy*** *cardinality-MI, R-append and N-append.*
- $\mathfrak{V}^{I_d}$ ***satisfies*** *minimality, flawed, abstraction-I and* ***does not satisfy*** *cardinality-MI, R-append, N-append and bottom facts.*

The lifting function uses a ranking on elements in order to provide a ranking on sets. We introduce some basic desirable **properties for lifting functions** below.

The *data sensitive* property ensures that element ranking is taken into account by the lifting function. The *abstraction-L* states that the ranking returned by the lifting function should not be concerned with the names of the elements.

Definition 5.36 (Data sensitive). *We say that a lifting function* $\mathtt{L}$ *satisfies* data sensitive *if and only if for any set of elements* Y *such that* $|Y| > 1$ *and any two sets* $E, E' \in 2^Y$*, there exist two different total, reflexive and transitive binary relations* $\succeq, \succeq'$ *on* Y *such that* $(E, E') \in \mathtt{L}^{\succeq}(Y)$ *and* $(E, E') \notin \mathtt{L}^{\succeq'}(Y)$.

Definition 5.37 (Abstraction-L). *We say that a lifting function* $\mathtt{L}$ *satisfies* abstraction-L *if and only if for any for any two set of elements* Y, Y'*, any total, reflexive and transitive binary relation* $\succeq$ *on* Y*, any two sets* $E, E' \in 2^Y$ *and any isomorphism* γ *such that* $\gamma(Y) = Y'$ *and* $\gamma(\succeq) = \succeq'$*, it holds that* $(E, E') \in \mathtt{L}^{\succeq}(Y)$ *if and only if* $(\gamma(E), \gamma(E')) \in \mathtt{L}^{\succeq'}(Y')$.

Proposition 5.19 (Max and leximax property satisfaction). *It holds that* $\mathtt{L}^{\succeq}_{max}, \mathtt{L}^{\succeq}_{leximax}$ ***satisfy*** data sensitive *and* abstraction-L.

Please note that $\models^{\mathcal{RIF}}_{x}$ with $x \in \{AR, IAR, ICR\}$ satisfies the QCE, QCI, Cons, ConsS, ConsC properties from the work of Baget et al. [2016b].

We now introduce some **properties on the whole framework**. A desirable property is the entailment of free facts (the *free* property). The

supremacy property states that if a fact is strictly less controversial than any other fact, then it should be entailed. *non entailment ejection* states that if a fact is only entailed by the closure of a repair which is not ranked amongst the best repairs, then it will not be entailed by the knowledge base. The *abstraction* property states that if there is a renaming of the constants and predicates in a knowledge base, the answers of our framework should remain unchanged.

Definition 5.38 (Free). *We say that a RIF* $\mathcal{RIF} = (\mathfrak{V}, \mathtt{L}, \models)$ *satisfies* free *if and only if for any knowledge base* $\mathcal{KB}$ *and any free fact* a *of* $\mathcal{KB}$*, it holds that* $\mathcal{KB} \models^{\mathcal{RIF}} a$.

Definition 5.39 (Supremacy). *We say that a RIF* $\mathcal{RIF} = (\mathfrak{V}, \mathtt{L}, \models)$ *satisfies* supremacy *if and only if for any knowledge base* $\mathcal{KB}$ *such that there exists* $f \in \mathcal{F}$ *with* $\mathfrak{V}_f(\mathcal{KB}) \neq 1$ *and for every* $f' \in \mathcal{F} \setminus \{f\}$*, we have* $\mathfrak{V}_f(\mathcal{KB}) < \mathfrak{V}_{f'}(\mathcal{KB})$ *then it holds that* $\mathcal{KB} \models f$.

Definition 5.40 (Non entailment ejection). *We say that a RIF* $\mathcal{RIF} = (\mathfrak{V}, \mathtt{L}, \models)$ *satisfies* non entailment ejection *if and only if for any knowledge base* $\mathcal{KB}$*, any* $R \in repairs(\mathcal{KB}) \setminus OUT_{\mathcal{RIF}}(\mathcal{KB})$ *and* $f \in \mathcal{C}\ell_{\mathcal{R}}(R)$ *such that there is no* $R' \in repairs(\mathcal{KB}) \setminus \{R\}$ *with* $f \in \mathcal{C}\ell_{\mathcal{R}}(R')$ *then* $\mathcal{KB} \not\models f$.

Definition 5.41 (Abstraction). *We say that a RIF* $\mathcal{RIF}$ *satisfies* abstraction *if and only if for any two knowledge base* $\mathcal{KB}, \mathcal{KB}'$*, any query* q*, any isomorphism* γ *such that* $\gamma(\mathcal{KB}) = \mathcal{KB}'$ *and* $x \in \{AR, IAR, ICR\}$*, it holds that* $\mathcal{KB} \models_x^{\mathcal{RIF}} q$ *if and only if* $\gamma(\mathcal{KB}) \models_x^{\mathcal{RIF}} \gamma(q)$.

Proposition 5.20 ($\mathcal{RIF}$ property satisfaction). *It holds that:*

- Abstraction-I *and abstraction-L imply* abstraction.
- *Let* $\mathcal{RIF} = (\mathfrak{V}, \mathtt{L}^{\geq}, \models_x^{\mathcal{RIF}})$ *where* $x \in \{AR, IAR, ICR\}$ *then* $\mathcal{RIF}$ *satisfies* free *and* non entailment ejection.
- *Let* $\mathcal{RIF} = (\mathfrak{V}, \mathtt{L}_y^{\geq}, \models_x^{\mathcal{RIF}})$ *where* $x \in \{AR, IAR, ICR\}$*,* $y \in \{leximax, max\}$ *and* $\mathfrak{V}$ *satisfies* bottom facts *then* $\mathcal{RIF}$ *satisfies* supremacy.

Although it is not always better to be more productive for all applications, the following result shows that the RIF is more productive than the usual IAR, AR and ICR semantics.

Proposition 5.21 (Productivity). *Let* $\mathcal{RIF}$ *be a RIF and be* q *a query, then:*

- *If* $\mathcal{KB} \models_x q$ *then* $\mathcal{KB} \models_x^{\mathcal{RIF}} q$*, where* $x \in \{AR, IAR, ICR\}$
- *If* $\mathcal{KB} \models_{IAR}^{\mathcal{RIF}} q$ *then* $\mathcal{KB} \models_{ICR}^{\mathcal{RIF}} q$
- *If* $\mathcal{KB} \models_{ICR}^{\mathcal{RIF}} q$ *then* $\mathcal{KB} \models_{AR}^{\mathcal{RIF}} q$

However, it is worth noting that our framework does not make every inconsistency-tolerant inference relation more productive. For instance, that is not the case with the non objection semantics defined by Benferhat et al. [2016].

5.3.2.2 Algorithmic results

In this section, we show a simple recursive algorithm that uses minimal inconsistent sets to compute the top result of our framework and we study the behaviour of this algorithm thanks to an experiment. Since finding each minimal inconsistent set has been proven to be practically feasible as it is polynomial for data complexity and exponential for combined complexity [Lembo et al., 2010; Hecham et al., 2017a], we make the assumption that this set is given. The top result of our framework is obtained by calling Algorithm 5.1 with X and Z initialised to $\emptyset$ and M to the set of minimal inconsistent sets of $\mathcal{KB}$. The parameter X is the set considered for building the result and Z is the set used for avoiding redundancies by memorising encountered sets. In Algorithm 5.1, we begin by checking if the set considered was already encountered (see line l_1). Then, we proceed by finding facts with a minimal score that can be added to X without triggering a minimal inconsistent set (see line l_2). If such facts cannot be found, it means that the considered set X is indeed a repair and should be returned (see line l_3). Otherwise, the above mentioned process is repeated by augmenting the set of the considered set X with the facts found in line l_2 (see line l_4). The set E contains repairs with maximal elements with respect to $\mathfrak{B}^{I_{MI}}$.[4] However, although these repairs contain maximal elements, they are not equivalent with respect to leximax and this is why we have to refine the set E (see line l_5). Please note that an alternate definition of $\mathfrak{B}^{I_{MI}}$ found in the work of Hunter and Konieczny [2010] can be used for a faster computation.

We ran the algorithm on the knowledge bases described by Yun et al. [2017b] and compared its performance with a basic algorithm for computing the RIF, namely naively finding all repairs, computing the inconsistency values and refining them by using leximax. The knowledge bases were split in two sets: A first set A of 108 knowledge bases with 2 to 7 facts, 0 to 6 rules and 1 to 4 binary or ternary negative constraints and a second set B of 26 knowledge bases with 8 facts, 6 rules and between 1 and 2 binary or ternary negative constraints. For further details about the knowledge bases, the reader is invited to consult the original paper of Yun et al. [2017b]. The results were as follows:

- For the set A, the average number of repairs per knowledge base was 2.89 and the average number of repairs in the top result of the RIF

[4] The obtained set is similar to the notion of preferred sub-theories of a stratification defined by Brewka [1989].

Algorithm 5.1 OFRecc

Data: Two sets of sets of facts M, Z and a set of facts X

Result: A set of repairs $\mathrm{OUT}_{\mathcal{RIF}}(\mathcal{KB})$ where $\mathcal{RIF} = (\mathfrak{V}_{I^{MI}}, \mathtt{L}^{\geq}_{leximax}, \models^{\mathcal{RIF}})$

begin

l_1 **if** $X \notin Z$ **then**

$Z \longleftarrow Z \cup \{X\}$

$Y \longleftarrow \{f \in \mathcal{F} \setminus X \mid \text{for all } m \subseteq X, m \cup \{f\} \notin M\}$

l_2 $Y' \longleftarrow \{y \in Y \mid \mathfrak{V}^{I_{MI}}_{y} \text{ is minimal}\}$

if $Y' = \emptyset$ **then**

l_3 **return** $\{X\}$

else

$E \longleftarrow \emptyset$ **for** $y' \in Y'$ **do**

l_4 $E \longleftarrow E \cup OFRecc(M, X \cup \{y'\})$

l_5 $E' \longleftarrow$ top sets of E with $\mathtt{L}^{\geq}_{leximax}$ with respect to $\mathfrak{V}^{I_{MI}}$

return E'

else

return $\emptyset$

was 2.11. It means that the average number of repair was reduced by 26.92%. Moreover, it takes an average of 543ms per knowledge base to find the top result of the RIF with the basic algorithm whereas it takes an average of 601ms per knowledge base to find the same top result with our algorithm.

- For the set B, the average number of repairs per knowledge base was 4.5 and the average number of repairs in the top result of the RIF was 3.62. It means that the average number of repair was reduced by of 19.66%. Moreover, it takes an average of 1.764s per knowledge base to find the top result of the RIF with the basic algorithm whereas it takes an average of 1.617s per knowledge base to find the top result with our algorithm.

Although finding the set Y' and E' (see l_2 and l_5) can be found in polynomial time, in the worst case scenario, we would still have to search throughout all the subsets of $\mathcal{F}$ which would be exponential with respect to $|\mathcal{F}|$. All experiments were performed on a Mac machine running on macOS High Sierra with an Intel core i5 2.8 GHz and 8GB of RAM and were reproduced multiple times.

5.3.2.3 Application scenario

We now consider an application scenario constructed in the setting of the Pack4Fresh project [Yun et al., 2016] which was aimed at choosing the

Figure 5.7: Representation of the considered packagings (from left to right): the opened plastic packaging without lid (opl), the wooden packaging (wp), the plastic packaging with a rigid lid (prl) and the plastic packaging with a plastic film (ppf).

best packaging for strawberries. In this project, an online poll consisting of 66 questions was submitted to an audience of 21 professionals from the food industry. We distinguished four kinds of professionals: the wholesalers, the floorwalkers, the quality managers and the warehouse managers. The questions were aimed at collecting the individual vision of each person about the characteristics of four packagings: the wooden packaging (wp), the plastic packaging with a plastic film (ppf), the plastic packaging with a rigid lid (prl) and the opened plastic packaging without lid (opl) (see Figure 5.7). The answers of this poll were formalised into a set of 50 facts and 160 rules. In our application scenario, the inconsistency of the knowledge base comes from the fusion of the divergent visions of the several professionals about the four aforementioned packagings. These visions were explicitly expressed using the rules. For instance, the rule $\forall X(ppf(X, wholesaler0) \rightarrow cheapCost(X))$ conveys the idea that the entity $wholesaler0$ believes that the plastic packaging with a plastic film is cheap. A group of packaging experts constructed another set of 18 rules constituting expert knowledge. For instance, the rule $\forall X(keepHumidity(X) \rightarrow badFridgeConservation(X))$ states that if X is a packaging that keeps humidity then X is a bad packaging for fridge conservation whereas the rule $\forall X(accelerateDecay(X) \rightarrow badEffectOnFruits(X))$ states that if X is a packaging that accelerates decay then it is considered as having a bad effect on fruits. Lastly, a set of 34 negative constraints representing conflicting atoms, such as $\forall X(notBadEffectOnFruits(X) \wedge badEffectOnFruits(X) \rightarrow \bot)$, and incompatibilities between packagings, such as $\forall X, Y, Z, T(opl(X, Y) \wedge prl(Z, T) \rightarrow \bot)$, was added.

The formalisation yielded a set of 33 repairs where each repair corresponds to the vision of a collection of individuals about a single packaging. For instance, we have that the repair $opl_2 = \{opl(po, floorwalker0), opl(po, wharehouse_manager0)\}$ corresponds to the vision of the two agents $floorwalker0$ and $wharehouse_manager0$ about the opened plastic packaging without lid. Amongst the 33 repairs, 16 concerned the wooden packaging, 6 concerned the plastic packaging with a rigid lid, 9 concerned the plastic packaging with a plastic film and 2 concerned the opened plastic packaging without lid. The different number of repairs is explained by the

diverse quantity of disagreements amongst individuals. For instance, only two wholesalers disagreed about the characteristics of the opened packaging without lid whereas eight wholesalers disagreed about the characteristics of wooden packaging. In our model, the size of the repair corresponds to the number of individuals that agreed on all the characteristics of a specific packaging. Please note that the repairs are not ranked solely based on their cardinality.

Surprisingly, the ranking on repairs was extremely clear as it showed that $wp > prl \sim ppf > opl$ (see Table 5.6). Indeed, the repairs about the wooden packaging were ranked above the other repairs. The repairs about the plastic packaging with a rigid lid were ranked roughly equally with the repairs about plastic packaging with a plastic film and the repairs about opened packaging without lid were last. The ranking was evaluated by a group of packaging experts which confirmed that the ranking on packagings was intuitive with respect to the data of the knowledge base. Indeed, the experts acknowledged that wooden packaging was ranked first because its characteristics were slightly less contested by the experts.

The knowledge base in DLGP format as well as a JAVA implementation of the tool for computing the top result of our framework is accessible at: `https://gite.lirmm.fr/yun/IJCAI2018`.

5.4 Summary

In this chapter, we first instantiated the argumentation-based ranking logic, defined by Amgoud et al. [2014], with the existential rules framework and showed that it is impacted by the notion of cores. Indeed, we highlighted that the ranking of some arguments in an argumentation graph and in a core of this graph can be different. The main reason for this change in the ranking of arguments is that redundant arguments (and thus the attacks from them) are removed. In the case where the argumentation graph is generated from a knowledge base, we defined two sufficient properties for the cores of the graph to be not different from the original graph. Moreover, we proposed several properties for characterising the ranking changes.

In the second part of this chapter, we studied how ranking-based semantics can be transposed in the setting of hypergraph argumentation frameworks. We translated the 18 desirable properties for ranking-based semantics on binary argumentation framework defined in the literature for argumentation hypergraphs. Then, we introduced the *nh-categoriser* as the first ranking-based semantics for argumentation hypergraphs and showed the existence and uniqueness of its solution. As we see it, those results will enable the argumentation community to further consider new kind of frameworks such as argumentation hypergraphs in the future.

Lastly, we presented the Ranking-based Inference Framework (RIF) that

wp_1
$\vdots$
wp_{16}
ppf_1, prl_1
ppf_2, prl_2
ppf_3
ppf_4
ppf_5, prl_3, prl_4
ppf_6
prl_5
prl_6
ppf_7
$\vdots$
ppf_9
opl_1
opl_2

From "worst" to "best" repairs

Table 5.6: Ranking on repairs. For simplicity, repairs are denoted by the packaging they are referencing

takes into consideration the inconsistency on the facts when using the repairs for query answering and restricts the set of repairs to the best with respect to inconsistency values. Since we consider a subset of repairs, we obtain more answers than classical inconsistency-tolerant query answering in most cases. We characterised the desirable properties of such a framework with respect to the following properties: Abstraction, Free, Non Entailment Ejection and Supremacy. We also introduced an algorithm for computing the top result of such a framework and showed its results on a real-life scenario about packagings. Please note that the ranking-based inference framework is more abstract than the work of Staworko et al. [2012] as we can consider an inconsistency value returning the same preferences on facts, a lifting function that uses these preferences such that the completion optimal repairs are ranked first, Globally optimal repairs second, Pareto optimal repairs third and followed by the other repairs.

Chapter 5 in a Nutshell

- We studied the impact of cores on the rankings returned by ranking-based semantics and proposed several properties for characterising ranking changes.
- We introduced the notion of ranking-based semantics for argumentation hypergraph and translated all of the existing properties for the binary argumentation graphs to argumentation hypergraphs. We also defined the *nh-categoriser* as the first ranking-based semantics for hypergraphs.
- We introduced the ranking-based inference framework for ranking repairs of an inconsistent knowledge base by using inconsistency values. We proposed several properties for the framework and describe a real-life application scenario.

6

Conclusion

6.1 Scope **160**

6.2 Summary and contributions **161**

6.3 Perspectives **163**

In this thesis, we answered a research question that originated from the necessity to preserve the ability to reason in presence of inconsistencies in a knowledge base expressed using the existential rules framework. The inconsistencies are conflicts that stem from incorrect factual knowledge. The problem of inconsistence has been addressed for existential rules using the various Repair Semantics. However, those approaches are lacking in many aspects, such that explanatory power, compared to logic-based argumentation approaches.

Research Question

How can we reason using logic-based argumentation in the context of the existential rules framework?

Our research hypothesis is that logic-based argumentation can be practically used to reason in presence of inconsistencies in real-life databases. However, although there has been many theoretical works on the structure of arguments, there is a lack of studies about the structural properties of the generated argumentation frameworks and the practical generation of such graphs. Our research question can be refined as follows:

1. What are the particularities of argumentation frameworks generated from existential rules knowledge bases?
2. How can we generate logic-based argumentation frameworks for the existential rules framework? Is this framework usable for large set of data and can we improve the generation process?
3. Can we provide tools that allow for an automatic generation of argumentation graphs?
4. What are the reasoning techniques that can be used with logic-based argumentation?

6.1 Scope

Since our research problem was vast, we restricted its scope in order to address it in a more focused manner:

- *Datalog+* is a first order logical language that has the ability to express knowledge about unknown individuals using value invention. The level of expressiveness brought by this ability comes at the high cost of computation tractability since the reasoning mechanism of *Datalog+* can be infinite. That is the reason why several decidable fragments of *Datalog+* have been defined: Finite Expansion Set (FES) [Baget et al., 2014b] guaranteeing a finite forward chaining mechanism (chase), Finite Unification Set (FUS) [Baget et al., 2011] guaranteeing a finite backward chaining mechanism, and Greedy Bounded Treewidth Set (GBTS) [Baget et al., 2011] guaranteeing a finite forward-like inference mechanism. We decided to work with the family of these decidable fragments which is called *Datalog±* or equivalently, the *existential rules framework*. As arguments are often composed of a conclusion that is deduced from a support, we choose to focus to *focus on the forward chaining mechanism* which is arguably the most intuitive one given its ability to handle transitive rules [Rocher, 2016]. The second question that we needed to answer is *what type of chase do we want to use?* There are four kinds of chases: Oblivious, Frontier/Skolem, Restricted, and Core. In this thesis, we use the Frontier/Skolem chase which is the most used chase given its relatively low cost and its ability to stay decidable for all known concrete classes of the FES fragment [Baget et al., 2011].

- *Logic-based argumentation* is an approach that consists in building arguments from a knowledge base using a given logic. There has been

many major logical approaches that have been studied in the literature: Assumption-Based Argumentation (ABA) [Toni, 2014], ASPIC+ [Modgil and Prakken, 2014], Defeasible Logic Programming (DeLP) [García and Simari, 2004] or Deductive argumentation [Besnard and Hunter, 2008]. However, since none of these previous approaches can be directly applied to a knowledge base expressed using the existential rules framework, we decided to focus on logic-based formalisms that were dedicated to this language, i.e. the argumentation frameworks defined by Croitoru and Vesic [2013] and Arioua [2016].

- *Inconsistency-tolerant techniques* are techniques that can be used for handling inconsistencies in knowledge bases expressed in the existential rules framework. In this thesis, we consider two of these techniques: repair semantics and argumentation semantics. For the former, focus on the most well-known repair semantics, namely AR, IAR and ICR [Lembo et al., 2010]. For the later, we focus only on extension-based [Dung, 1995] and ranking-based approaches [Amgoud and Ben-Naim, 2015].

6.2 Summary and contributions

In order to achieve our goals, we provided three main contributions:

1. We revisited the logic-based argumentation framework defined by Arioua [2016] and showed many of its structural properties. We proved that this argumentation framework has an exponential number of arguments with respect to the size of the knowledge base which makes it difficult to be used in real-life applications.

2. We provided optimisations in order to accelerate the generation and limit the number of arguments. We also defined a new logic-based hypergraph argumentation framework for the existential rules framework and empirically showed that it is more efficient that the binary argumentation framework.

3. On the one hand, we studied ranking-based approaches for binary argumentation frameworks and introduced how this notion can be applied for hypergraph argumentation frameworks. On the other hand, we showed that ranking-based approaches can also be applied directly on the factual knowledge in order to improve repair semantics.

In Chapter 2, we introduced the existential rules framework and reviewed the various reasoning techniques that are used in the context of the abstract argumentation model proposed by Dung [1995].

In Chapter 3, we studied the specially crafted deductive logic-based argumentation framework proposed by Arioua et al. [2017] and Yun et al. [2017a]. In this framework, we highlighted several structural properties such as the presence of particular arguments and subgraphs, the presence of symmetries, the characterisation of strongly connected components or the explosion of the number of arguments when free facts are added. To solve the problem of the huge amount of arguments generated, we defined optimisation methods in both the case of knowledge bases with or without positive rules. In the first case, we showed that the generation of the argumentation graph can be fastened using a pre-processing of the underlying knowledge base. In the second case, we showed that the number of arguments can be reduced by filtering "redundant" arguments and replacing the binary attack relation by sets of attacking arguments. Using these insights, we developed the first tool for generating argumentation frameworks from existential rules knowledge bases (DAGGER) and we compared the results of the best solvers for argumentation semantics on the generated graphs. The benchmark provided a clear view that logically generated argumentation graphs are very different from randomly generated graphs usually used in benchmarks. Moreover, the benchmark can be used by a data engineer to choose the best solver for logically generated graphs.

Based on the insights that we uncovered when optimising the existing argumentation framework for existential rules, we introduced in Chapter 4, a new logic-based argumentation framework which is an instantiation of the framework defined by Nielsen and Parsons [2007] and showed in an experiment that it is more efficient both with respect to the size of the argumentation graph and with respect to the computational time. Indeed, we showed that this new argumentation framework does not suffer from the aforementioned problem of explosion of the number of arguments. Implementing this new framework in a tool (NAKED) enabled for the generation, visualisation and export of an argumentation hypergraph generated from an existential rules knowledge base.

After solving the problem of the generation of the argumentation graph, we investigated, in Chapter 5, how ranking-based techniques can be used for handling inconsistencies in both the logic-based argumentation theory and the logic setting. First, we studied the behaviour of the ranking on arguments returned by classical ranking-based semantics, in the context of the framework of Arioua et al. [2017] and Yun et al. [2017a], when redundant knowledge is present in the underlying knowledge base. We provided sufficient conditions on the knowledge base for preventing the apparition of redundant arguments and showed several examples of ranking modifications caused by redundancies. Second, we provided an extension of the notion of ranking-based semantics to the special case of hypergraph argumentation by converting existing desirable properties and ranking-based se-

mantics. Lastly, we showed that ranking-based techniques can also be used directly on knowledge bases using the ontological layer in order to compute the contribution of each fact in the overall inconsistency of the knowledge base. This approach is not only able to rank pieces of knowledge with respect to their "*inconsistency value*" but can also be used to rank repairs using the same approach as Konieczny et al. [2015] and Bonzon et al. [2018].

6.3 Perspectives

This thesis answered the research question we set out to address and our contributions opened interesting avenues for future work.

1. **Ranking-based techniques.**

   Ranking-based techniques are often used in the logical setting in order to get a degree of inconsistency or acceptability. In this thesis, we studied ranking-based techniques from two perspectives: the argumentation and the knowledge base perspectives. In the argumentation perspective, ranking-based semantics are used for ranking arguments with respect to the structure of the underlying argumentation framework. The ranking on arguments obtained by these ranking-based semantics can be used in many ways. As we showed in Section 5.1, the approach of Amgoud and Ben-Naim [2015] is able to rank formulas using the output of ranking-based semantics. In the work of Yun et al. [2018h], the output of ranking-based semantics is "lifted" in order to rank sets of arguments or extensions. Likewise, in the knowledge base perspective, we showed in Section 5.3 that the approach of Yun et al. [2018g] is also able to rank formulas based on their contributions in the overall inconsistency of the knowledge base and that this ranking can also be "lifted" to repairs.

   - **Correspondence between perspectives.** The first question that we ask here is: *Can we find a correspondence between ranking-based techniques in the two perspectives?* This is a natural question as ranking-based techniques in the two perspectives can yield rankings on formulas and rankings on extensions or repairs. Moreover, since there is an equivalence between repairs and preferred (respectively stable) extensions in most logic-based argumentation frameworks with existential rules (see Chapter 3 and Chapter 4), a ranking on extensions can be converted into a ranking on repairs and vice versa. A positive answer to this question would provide a link between argumentation theory and the results in the knowledge representation community and allow practical applications to explain to users why a particular for-

mula holds by constructing and highlighting several arguments in favour or against this formula.

- **Improving rankings.** The second question that we ask is: *Can the ranking-based techniques be improved?* Ranking techniques in argumentation make use of the structure of an argumentation framework whereas in the knowledge base perspective, ranking techniques are based on the notion of minimal inconsistent sets. Following the work of Bonzon et al. [2018], it would be possible to combine these two techniques. It would be interesting to study what properties are produced by the combination of these two techniques.

2. **Efficient generation.** In Chapter 4, we introduced a new framework for logic-based argumentation that is based on instantiating hypergraphs. As we showed in an experiment, this new framework is much more efficient than those of Croitoru and Vesic [2013] and Arioua et al. [2017]. Against this background, it would be interesting to scale up the experiment and generate argumentation frameworks using large existential rules knowledge bases such as LUBM [Bourgaux, 2016].

3. **Incoherence.** We showed in Chapter 2 that there are two types of conflicts: inconsistencies and incoherences. In this thesis, we focused on how to solve inconsistencies using logic-based argumentation techniques. However, it would be interesting to study how incoherences can be handled in the context of the existential rules framework. In the work of Martinez et al. [2014], the author proposed a logic-based argumentation framework for the existential rules framework inspired by DeLP. They introduced *Defeasible Datalog*± which added a defeasible layer composed of defeasible atoms (weaker statements) and rules (weaker inferences). In the work of Hecham [2018], defeater rules (rules that prevent the application of defeasible rules) were added to the defeasible layer of defeasible *Datalog*±. In this setting, the author introduced the *Statement Graph* and the *Graph of Atom Dependency* [Hecham et al., 2017b] as two new logic-based formalisms for reasoning with incoherent knowledge. To the best of our knowledge, the three aforementioned formalisms are currently the only works that study how to handle incoherences in the context of the existential rules framework. In the future, it would be interesting to explore how other argumentation frameworks such as ASPIC+ and ABA can be instantiated with the existential rules framework [Yun and Croitoru, 2016; Lam et al., 2016] and to show that there is potential for cross-fertilisation for research on the relationship between defeasible reasoning in existential rules and major logic-based argumentation frameworks.

4. **Argumentation & human reasoning.** Computer supported collaborative decision-making is a task that consists in collectively taking a decision via social interaction using a computer. Previous research gives grounds to believe that groups of students that are more efficient than others for collective tasks use specific speech and argumentation typologies in their interaction during the social interaction. In the PEPS S2IH APOLONIO (Argument Patterns in computer supported cOllaborative LearNIng) project, we aimed at collecting the interaction data and applying the existing approaches from the argumentation literature. In this project, several student classes were divided into groups for or against a specific topic. Each student and group had to rank arguments while their interactions was recorded. Then, different pairs of groups for or against the topic were asked to find a common ranking on the whole set of arguments. In the future, we plan to publish our results on this experiment. This cross-fertilisation will be beneficial to both computer scientists and psychologists. On one hand, the human sciences will use the approaches developed in the computer science community in order to identify the argument patterns used by efficient groups. On the other hand, computer sciences will benefit from the real-world data coming from other domains in order to compare and evaluate existing theoretical approaches.

5. **Practical applications.**

   In the setting of the Pack4Fresh project for reducing post-harvest wastes, we developed a decision-support system (DSS) for ranking strawberry packaging alternatives based on poll results and argumentation. In the real world, choosing a packaging alternative can be an arduous task as it requires a balance between pros and cons. For instance minimising food waste and losses could also significantly contribute to decrease the overall environmental impact of the food itself. In our workflow, we collected answers to a poll from a large number of consumers in order to model the overall viewpoint of consumers about the characteristics of each packaging in propositional logics. Then, we instantiated an ASPIC+ argumentation framework based the aforementioned knowledge base. The extensions are computed for a specific semantics and ranked with respect to sets of preferences on packaging characteristics [Yun et al., 2018a]. In this system, the ranking on alternatives given by the DSS is greatly affected by the user's preferences. In the future, we plan on further implementing (1) automatic rankings that rely on the structure of the argumentation framework itself [Yun et al., 2018h], (2) automatic rankings that are based on the knowledge base itself (see Section 5.3) and (3) switching to a more expressive language such as the existential rules framework or description logics.

7 Appendix

7.1 Miscellaneous . . . . . **i**
7.2 Proofs . . . . . **ii**
7.2.1 Chapter 3 ii
7.2.2 Chapter 4 viii
7.2.3 Chapter 5 xii

This chapter contains the proofs for the propositions and examples of the several chapters of this thesis.

7.1 Miscellaneous

Example 3.19 Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that $\mathcal{F} = \{a(m), b(m), c(m), d(m), e(m)\}$, $\mathcal{R} = \emptyset$ and $\mathcal{N} = \{\forall X(a(X) \wedge b(X) \wedge c(X) \rightarrow \bot), \forall X(e(X) \wedge d(X) \rightarrow \bot)\}$. The corresponding argumentation framework is composed of 161 attacks and the following 20 arguments:

- $a_0 : (\{a(m)\}, \{a(m)\})$
- $a_1 : (\{b(m)\}, \{b(m)\})$
- $a_2 : (\{a(m), b(m)\}, \{a(m), b(m)\})$
- $a_3 : (\{c(m)\}, \{c(m)\})$
- $a_4 : (\{a(m), c(m)\}, \{a(m), c(m)\})$
- $a_5 : (\{b(m), c(m)\}, \{b(m), c(m)\})$
- $a_6 : (\{d(m)\}, \{d(m)\})$
- $a_7 : (\{a(m), d(m)\}, \{a(m), d(m)\})$
- $a_8 : (\{b(m), d(m)\}, \{b(m), d(m)\})), d(m)\})$

- $a_9 : (\{a(m), b(m), d(m)\}, \{a(m), b(m), d(m)\})$
- $a_{10} : (\{c(m), d(m)\}, \{c(m), d(m)\})$
- $a_{11} : (\{a(m), c(m), d(m)\}, \{a(m), c(m), d(m)\})$
- $a_{12} : (\{b(m), c(m), d(m)\}, \{b(m), c(m), d(m)\})$
- $a_{13} : (\{e(m)\}, \{e(m)\})$
- $a_{14} : (\{a(m), e(m)\}, \{a(m), e(m)\})$
- $a_{15} : (\{b(m), e(m)\}, \{b(m), e(m)\})$
- $a_{16} : (\{a(m), b(m), e(m)\}, \{a(m), b(m), e(m)\})$
- $a_{17} : (\{c(m), e(m)\}, \{c(m), e(m)\})$
- $a_{18} : (\{a(m), c(m), e(m)\}, \{a(m), c(m), e(m)\})$
- $a_{19} : (\{b(m), c(m), e(m)\}, \{b(m), c(m), e(m)\})$

The preferred extensions will be composed of the following sets:

- $\varepsilon_1 = \{a_0, a_1, a_2, a_6, a_7, a_8, a_9\}$
- $\varepsilon_2 = \{a_0, a_3, a_4, a_6, a_7, a_{10}, a_{11}\}$
- $\varepsilon_3 = \{a_1, a_3, a_5, a_6, a_8, a_{10}, a_{12}\}$
- $\varepsilon_4 = \{a_0, a_1, a_2, a_{13}, a_{14}, a_{15}, a_{16}\}$
- $\varepsilon_5 = \{a_0, a_3, a_4, a_{13}, a_{14}, a_{17}, a_{18}\}$
- $\varepsilon_6 = \{a_1, a_3, a_5, a_{13}, a_{15}, a_{17}, a_{19}\}$

The set of cf2 extensions is the set $\{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5, \varepsilon_6, \varepsilon_7, \varepsilon_8\}$ with:

- $\varepsilon_7 = \{a_0, a_1, a_3, a_6, a_7, a_8, a_{10}\}$
- $\varepsilon_8 = \{a_0, a_1, a_3, a_{13}, a_{14}, a_{15}, a_{17}\}$

7.2 Proofs

7.2.1 Chapter 3

Proposition 3.11 (Characterisation of dummy arguments) Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be knowledge base such that $\mathcal{R} = \emptyset$ and $|\mathcal{F}| = n$. There are exactly $2^k - 1$ dummy arguments in $\mathfrak{F}_{\mathcal{KB}}$ where $k = |Free(\mathcal{KB})|$.

Proof 3.11. Let us consider $Unn = \{a \in \mathscr{A} \mid Att^-(a) = Att^+(a) = \emptyset\}$, the set of dummy arguments.

1. Let us prove that $|Unn| \geq 2^k - 1$. The set $Free(\mathcal{KB})$ corresponds to the set of facts that are not in any conflict. Since $k = |Free(\mathcal{KB})|$, we conclude that there are at least $2^k - 1$ arguments of the form (X, X) that have a non empty subset X of $Free(\mathcal{KB})$ as support. These arguments are not attacked and do not attack other arguments as the elements of their supports and conclusions are not in any conflict.

2. Let us prove that $|Unn| \leq 2^k - 1$. By means of contradiction, we suppose that $|Unn| > 2^k - 1$. It means that that there is a dummy argument $a \in Unn$ such that $Supp(a) \nsubseteq Free(\mathcal{KB})$. Thus, there exists a minimal inconsistent set $X \in MI(\mathcal{KB})$ such that $X \cap Supp(a) \neq \emptyset$. Now, let us consider $Y = X \setminus Supp(a)$. We know that Y is not empty otherwise there is a contradiction with the consistency of the support of a. Furthermore, Y is $\mathcal{R}$-consistent since $Y \subset X$. Thus, there is an argument $b = (Y, Y)$ such that $(b, a) \in \mathscr{R}$, contradiction.

□

Proposition 3.12 (Number of arguments) Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base with $\mathcal{R} = \emptyset$. If $|Free(\mathcal{KB})| = k$ then there is a subgraph of $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ that is a (2^k)-*copy* graph of $\mathfrak{F}_{\mathcal{KB}'} = (\mathscr{A}', \mathscr{R}')$ where $\mathcal{KB}' = (\mathcal{F} \setminus Free(\mathcal{KB}), \mathcal{R}, \mathcal{N})$ and $|\mathscr{A}| = (|\mathscr{A}'| + 1) * 2^k - 1$.

Proof 3.12. If $Free(\mathcal{KB}) = \emptyset$, then it is obvious that $\mathfrak{F}_{\mathcal{KB}}$ is a 1-*copy* graph of itself. Suppose now that $Free(\mathcal{KB}) \neq \emptyset$. We denote by $\mathfrak{F}_{\mathcal{KB}'} = (\mathscr{A}', \mathscr{R}')$ the argumentation framework from the knowledge base $\mathcal{KB}' = (\mathcal{F} \setminus Free(\mathcal{KB}), \mathcal{R}, \mathcal{N})$. Moreover, since $\mathcal{R} = \emptyset$, the arguments can only be of the form (X, X) where X is an $\mathcal{R}$-consistent subset of $\mathcal{F} \setminus Free(\mathcal{KB})$. Hence, $|\mathscr{A}'| = |\{X \mid X$ is a non empty $\mathcal{R}$-consistent subset of $\mathcal{F} \setminus Free(\mathcal{KB})\}|$.

Now, let us consider $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$, the argumentation framework corresponding to the knowledge base $\mathcal{KB}$. We show that the subgraph $\mathfrak{F}''_{\mathcal{KB}} = (\mathscr{A}'', \mathscr{R}'')$ of $\mathfrak{F}_{\mathcal{KB}}$ where $\mathscr{A}'' = \{a \in \mathscr{A} \mid Supp(a) \nsubseteq Free(\mathcal{KB})\}$ and $\mathscr{R}'' = \mathscr{R}_{|\mathscr{A}''}$ is a $(2^{|Free(\mathcal{KB})|})$-*copy* graph of $\mathfrak{F}_{\mathcal{KB}'}$:

- We know that for any set X that is an $\mathcal{R}$-consistent subset of $\mathcal{F} \setminus Free(\mathcal{KB})$, $X \cup X'$, where X' is a subset of $Free(\mathcal{KB})$, is an $\mathcal{R}$-consistent set. Thus $|\mathscr{A}''| = |\{X \cup X' \mid X' \subseteq Free(\mathcal{KB})$ and X is a non empty $\mathcal{R}$-consistent subset of $\mathcal{F} \setminus Free(\mathcal{KB})\}|$. Since the number of subsets of $Free(\mathcal{KB})$ is $2^{|Free(\mathcal{KB})|}$, then $|\mathscr{A}''| = |\mathscr{A}'| * 2^{|Free(\mathcal{KB})|}$.

- We denote by f the function from $\mathscr{A}''$ to $\mathscr{A}'$ such that $f(a'') = a'$ iff $Supp(a') = Supp(a'') \cap (\mathcal{F} \setminus Free(\mathcal{KB}))$. We now show that this function

is surjective. Let a' be an argument of $\mathscr{A}'$ and c an arbitrary element of $Free(\mathcal{KB})$ (it exists since $Free(\mathcal{KB}) \neq \emptyset$). As mentioned before, we know that $E = Supp(a') \cup \{c\}$ is $\mathcal{R}$-consistent. Therefore $a'' = (E, E)$ is an argument of $\mathscr{A}''$ and $f(a'') = a'$.

- Let $a' \in \mathscr{A}'$ and $W_{a'} = \{a'' \in \mathscr{A}'' \mid f(a'') = a'\}$. For every subset X of $Free(\mathcal{KB})$, $L = X \cup Supp(a')$, $(L, L) \in W_{a'}$. Since the number of different subsets of $Free(\mathcal{KB})$ is $2^{|Free(\mathcal{KB})|}$, we have $|W_{a'}| \geq 2^{|Free(\mathcal{KB})|}$. Since for every $a'_1, a'_2 \in A'$, $W_{a'_1} \cap W_{a'_2} = \emptyset$, then for every $a' \in \mathscr{A}', |W_{a'}| = 2^{|Free(\mathcal{KB})|}$ because $|\mathscr{A}''| = |\mathscr{A}'| * 2^{|Free(\mathcal{KB})|}$.

- Let $(a''_1, a''_2) \in \mathscr{R}''$, by definition, we have that there exists $\phi \in Supp(a''_2)$ s.t. $Conc(a''_1) \cup \{\phi\}$ is $\mathcal{R}$-inconsistent. Since there are no rules, it is true that $Supp(a''_1) \cup \{\phi\}$ is also $\mathcal{R}$-inconsistent. However, this inconsistency cannot come from elements of $Free(\mathcal{KB})$. Thus, there exists $\phi \in Supp(a''_2) \cap (\mathcal{F} \setminus Free(\mathcal{KB}))$ such that $(Supp(a''_1) \cap (\mathcal{F} \setminus Free(\mathcal{KB})) \cup \{\phi\}$ is $\mathcal{R}$-inconsistent. Therefore $(f(a''_1), f(a''_2)) \in \mathscr{R}'$ since $Supp(f(a''_1)) = Supp(a''_1) \setminus Free(\mathcal{KB})$ and $Supp(f(a''_2)) = Supp(a''_2) \setminus Free(\mathcal{KB})$.

- Let $a''_1, a''_2 \in \mathscr{A}''$ such that $(f(a''_1), f(a''_2)) \in \mathscr{R}'$. It means that there exists $\phi \in Supp(f(a''_2))$ s.t. $Conc(f(a''_1)) \cup \{\phi\}$ is $\mathcal{R}$-inconsistent. By definition, we have that $Supp(f(a''_2)) = Supp(a''_2) \setminus Free(\mathcal{KB})$, thus $\phi \in Supp(a''_2)$. Likewise, we have that $Conc(f(a''_1)) = Supp(f(a''_1)) = Supp(a''_1) \setminus Free(\mathcal{KB}) = Conc(a''_1) \setminus Free(\mathcal{KB})$. We conclude that $(Conc(a''_1) \setminus Free(\mathcal{KB})) \cup \{\phi\}$ is $\mathcal{R}$-inconsistent. Therefore $Conc(a''_1) \cup \{\phi\}$ is $\mathcal{R}$-inconsistent and $(a''_1, a''_2) \in \mathscr{R}''$.

Finally, we have that $|\mathscr{A}| = |\{X \mid X \text{ is an } \mathcal{R}\text{-consistent subset of } \mathcal{F}\}| = |\{X \mid X \not\subseteq Free(\mathcal{KB}) \text{ and } X \text{ is an } \mathcal{R}\text{-consistent subset of } \mathcal{F}\} \cup \{X \mid X \subseteq Free(\mathcal{KB})\}| - 1 = |\mathscr{A}'| * 2^{|Free(\mathcal{KB})|} + 2^{|Free(\mathcal{KB})|} - 1 = (|\mathscr{A}'| + 1) * 2^{|Free(\mathcal{KB})|} - 1$. This concludes the proof. □

Proposition 3.15 (SCC characterisation) Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base such that $\mathcal{R} = \emptyset$ and $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ be the corresponding argumentation framework. We have that:

1. $s_i \in SCC(\mathfrak{F}_{\mathcal{KB}})$ where $s_i = \{(X_i, X_i)\}$ with $X_i \in 2^{\mathcal{F}} \setminus \bigcup_{C \in MI(\mathcal{KB})} Imp(C)$
2. $(\mathscr{A} \setminus \bigcup_i s_i) \in SCC(\mathfrak{F}_{\mathcal{KB}})$

Proof 3.15. We split the proof in two parts:

1. Suppose that s_i is not a strongly connected component by itself, it means that there is another argument a such that there is a path from $s_i = \{X_i, X_i\}$ to a and inversely. Let us denote by a_1, the first argument attacked by s_i on a path from s_i to a. By definition, it means that there

exists $\phi \in Supp(a_1)$ such that $X_i \cup \{\phi\}$ is $\mathcal{R}$-inconsistent. Since X_i is $\mathcal{R}$-consistent, it means that $X_i \cup \{\phi\}$ is a minimal conflict and that $X_i \in Imp(X_i \cup \{\phi\})$. Then, $Xi \notin 2^{\mathcal{F}} \setminus \bigcup_{C \in MI(\mathcal{KB})} Imp(C)$, contradiction.

2. Let a, b be two arguments in $(\mathscr{A} \setminus \bigcup_i s_i)$, we show here that there is a path from a to b. From the definitions, we know that a (resp. b) is of the form (X, X) (resp.(X', X')) such that there exists a minimal conflict C (resp. C') and $W \subseteq C$ (resp. $W' \subseteq C'$) with $|W| = |C - 1|$ (resp. $|W'| = |C' - 1|$) and $W \subseteq X$ (resp. $W' \subseteq X'$).

   Let $H = C \setminus X$, $X'' = X' \setminus H$, $W'' \subseteq X''$ with $|W''| = |X'' - 1|$ and $J = H \cup W'' \cup (C' \setminus X')$.

   - If J is $\mathcal{R}$-consistent, we denote by u, the argument (J, J). We have that u belongs to $(\mathscr{A} \setminus \bigcup_i s_i)$ because $J = |C' - 1|$ and $J \subseteq C'$. We have that a attacks u and u attacks b.
   - If J is $\mathcal{R}$-inconsistent, it means that there is a minimal conflict $C'' \subseteq J$ such that $C'' \nsubseteq C'$ and $C'' \nsubseteq C$. Let us consider $K, L \subseteq J$ such that $|K| = |L| = |J - 1|$, $H \subseteq K$ and $H \nsubseteq L$. By definition, K and L are $\mathcal{R}$-consistent, thus the arguments $c = (K, K)$ and $d = (L, L)$ exist. We have that a attacks c, c attacks d and d attacks b.

□

Proposition 3.17 (Repair equivalence) Let $\mathfrak{F}^*_{\mathcal{KB}} = (\mathscr{A}^*, \mathscr{R}^*)$, it holds that $Ext_x(\mathfrak{F}^*_{\mathcal{KB}}) = \{Arg(A', \mathscr{A}^*) \mid A' \in repairs(\mathcal{KB})\}$ for $i \in \{s, p\}$.

Proof 3.17. The proof is split in two parts. Let $\mathfrak{F}^*_{\mathcal{KB}} = (\mathscr{A}^*, \mathscr{R}^*)$ and $N = \{Arg(A', \mathscr{A}^*) \mid A' \in repairs(\mathcal{KB})\}$. We first show that $N \subseteq Ext_s(\mathfrak{F}^*_{\mathcal{KB}})$. Then since, the set of stable extensions is included in the set of preferred extensions Nielsen and Parsons [2006], we have that $N \subseteq Ext_p(\mathfrak{F}^*_{\mathcal{KB}})$. In the second part of the proof, we show that $Ext_p(\mathfrak{F}^*_{\mathcal{KB}}) \subseteq N$.

- We first show $N \subseteq Ext_s(\mathfrak{F}^*_{\mathcal{KB}})$. Let A' be a repair of $\mathcal{KB}$ and let $E = Arg(A', \mathscr{A}^*)$. Let us prove that E is a stable extension of $\mathfrak{F}^*_{\mathcal{KB}}$. We first prove that E is conflict-free. By means of contradiction we suppose the contrary, i.e. let $X \subseteq E$, $b \in E$ such that $(X, b) \in \mathscr{R}^*$. From the definition of attack, there exists $\phi \in Supp(b)$ such that $(\bigcup_{a \in X} Conc(a)) \cup \{\phi\}$ is $\mathcal{R}$-inconsistent. Thus $(\bigcup_{x \in X} Supp(x)) \cup \{\phi\}$ is $\mathcal{R}$-inconsistent and A' is $\mathcal{R}$-inconsistent, contradiction. Therefore E is conflict-free.

  Let us now prove that E attacks all arguments outside the set E. Let $b \in \mathscr{A}^* \setminus Arg(A', \mathscr{A}^*)$ and let $\phi \in Supp(b)$ such that $\phi \notin A'$. By definition, we know that there is $S' \subseteq A'$ such that $c = (S', A')$ is an argument in $\mathfrak{F}_{\mathcal{KB}}$. If $c \in \mathscr{A}^*$, we have that $(\{c\}, b) \in \mathscr{R}^*$. Otherwise, there

exists a set of arguments $M = \{a_1, a_2, \ldots, a_n\} \subseteq \mathscr{A}^*$ with $Supp(a_i) = S'$ such that $\bigcup_{a_i \in M} Conc(a_i) = A'$. Hence, $(M, b) \in \mathscr{R}^*$ and E is a stable extension.

- We now show that $Ext_p(\mathfrak{F}^*_{\mathcal{KB}}) \subseteq N$. Let $E \in Ext_p(\mathfrak{F}^*_{\mathcal{KB}})$ and let us prove that there exists a repair A' such that $E = Arg(A', \mathscr{A}^*)$. Let $S = Base(E)$. Let us prove that S is $\mathcal{R}$-consistent. Aiming to a contradiction, suppose that S is $\mathcal{R}$-inconsistent. Let $S' \subseteq S$ be such that:

  1. S' is $\mathcal{R}$-inconsistent
  2. Every proper set of S' is $\mathcal{R}$-consistent. Let us denote by S' the set $\{\phi_1, \phi_2, \ldots, \phi_n\}$.

  Let $a \in E$ be an argument such that $\phi_n \in Supp(a)$. Let $S'' \subseteq S' \setminus \{\phi_n\}$ such that $c = (S'', S' \setminus \{\phi_n\})$ is an argument of $\mathfrak{F}_{\mathcal{KB}}$. Then, if $c \in \mathscr{A}^*$, we have that $(\{c\}, a) \in \mathscr{R}^*$. Otherwise, there exists a set of arguments $M = \{a_1, a_2, \ldots, a_n\} \subseteq \mathscr{A}^*$ with $Supp(a_i) = S''$ such that $\bigcup_{a_i \in M} Conc(a_i) = S' \setminus \{\phi_n\}$. Hence, a is attacked by c or by M. Since E is conflict-free, we have that $c \notin E$ (resp $M \nsubseteq E$). However, since E is admissible, there exists $B = \{b_1, \ldots b_n\}$ such that $B \subseteq E$ and $(B, c) \in \mathscr{R}^*$ (respectively there exists $a_i \in M \setminus E$ and $(B, a_i) \in \mathscr{R}^*$). By definition, this means that there exists $\phi_j \in Supp(c)$ (resp $\phi_j \in Supp(a_i)$) such that $(\bigcup_{b_i \in B} Conc(b_i)) \cup \{\phi_j\}$ is $\mathcal{R}$-inconsistent. Since $\phi_j \in S$, there is an argument $d \in E$ such that $\phi_j \in Supp(d)$. Therefore, $(B, d) \in \mathscr{R}^*$. Contradiction and E is $\mathcal{R}$-consistent.

  Let us now prove that there exists no $S' \subseteq \mathcal{F}$ such that $S \subset S'$ and S' is $\mathcal{R}$-consistent. We use the proof by contradiction. Thus, suppose that S is not a maximal $\mathcal{R}$-consistent subset of $\mathcal{F}$. Then, there exists a repair S' of $\mathcal{KB}$, such that $S \subset S'$. We have that $E \subseteq Arg(S, \mathscr{A}^*)$. Denote $E' = Arg(S', \mathscr{A}^*)$. Since $S \subset S'$ then $Arg(S, \mathscr{A}^*) \subset E$. Thus, $E \subset E'$. From the first part of the proof, $E' \in Ext_s(\mathfrak{F}^*_{\mathcal{KB}})$. Consequently, $E' \in Ext_p(\mathfrak{F}^*_{\mathcal{KB}})$. We also know that $E \in Ext_p(\mathfrak{F}^*_{\mathcal{KB}})$. Contradiction, since no preferred set can be a proper subset of another preferred set. Thus, we conclude that $Base(E)$ is a repair of $\mathcal{KB}$.

  Let us show that $E = Arg(Base(E), \mathscr{A}^*)$. It must be that $E \subseteq Arg(S, \mathscr{A}^*)$. Also, we know (from the first part) that $Arg(S, \mathscr{A}^*)$ is a stable and a preferred extension, thus the case $E \nsubseteq Arg(S, \mathscr{A}^*)$ is not possible

□

Proposition 3.19 (Attack properties) Let $\mathcal{KB}$ be a knowledge base, $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ be the corresponding argumentation framework and $\mathfrak{F}^*_{\mathcal{KB}} = (\mathscr{A}^*, \mathscr{R}^*)$ be the filtrated argumentation framework. It holds that:

1. $a \in \mathscr{A}^*$ is not attacked in $\mathfrak{F}_{\mathcal{KB}}$ if and only if a is not attacked in $\mathfrak{F}^*_{\mathcal{KB}}$

2. if $a \in \mathscr{A}^*$ is attacked in $\mathfrak{F}_{\mathcal{KB}}$ then $|Att^-_{\mathfrak{F}_{\mathcal{KB}}}(a)| \leq |Att^-_{\mathfrak{F}^*_{\mathcal{KB}}}(a)|$

Proof 3.19. The proofs are as follows:

1. We show that $a \in \mathscr{A}^*$ is not attacked in $\mathfrak{F}_{\mathcal{KB}}$ if and only if a is not attacked in $\mathfrak{F}^*_{\mathcal{KB}}$. The proof is split in two parts:

   - ($\Rightarrow$) Let $a \in \mathscr{A}^*$ and suppose that a is not attacked in $\mathfrak{F}_{\mathcal{KB}}$ but a is attacked in $\mathfrak{F}^*_{\mathcal{KB}}$. If there exists $b \in \mathscr{A}^*$ such that $(\{b\}, a) \in \mathscr{R}^*$. Then, we have that $(b, a) \in \mathscr{R}$, hence a is attacked in $\mathfrak{F}_{\mathcal{KB}}$. Contradiction. Otherwise, there exists $B = \{b_1, \ldots, b_n\} \subseteq \mathscr{A}^*$ such that $(B, a) \in \mathscr{R}^*$. Let $CB = \bigcup_{b_i \in B} Conc(b_i)$. Let $SB = Supp(b_1) = \cdots = Supp(b_n)$. We now show that $b = (SB, CB)$ is an argument in $\mathfrak{F}_{\mathcal{KB}}$. Indeed, SB is consistent, $CB \subseteq Sat_{\mathfrak{R}}(SB)$. By means of contradiction, suppose that SB is not a minimal set satisfying the previous two conditions and let $SB' \subset SB$ such that $CB \subseteq Sat_{\mathfrak{R}}(SB')$. Then, b_1 is not an argument since $(SB', Conc(b_1))$ is an argument. Thus, b is an argument in $\mathfrak{F}_{\mathcal{KB}}$ and we have that $(b, a) \in \mathscr{R}$.
   - ($\Leftarrow$) Suppose now that a is attacked in $\mathfrak{F}_{\mathcal{KB}}$ but that a is not attacked in $\mathfrak{F}^*_{\mathcal{KB}}$. Let $(b, a) \in \mathscr{R}$. The case $b \in \mathscr{A}^*$ is trivial because $(\{b\}, a) \in \mathscr{R}^*$. Suppose that $b \notin \mathscr{A}^*$. This means that there exists $B = \{b_1, \ldots, b_n\} \subseteq \mathscr{A}^*$ such that $Supp(b_1) = \cdots = Supp(b_n)$ and $Conc(b) = \bigcup_{b_i \in B} Conc(b_i)$. So either $(B, a) \in \mathscr{R}^*$ or there exists $B' \subset B$ such that $(B', a) \in \mathscr{R}^*$.

2. We show that if $a \in \mathscr{A}^*$ is attacked in $\mathfrak{F}_{\mathcal{KB}}$ then $|Att^-_{\mathfrak{F}_{\mathcal{KB}}}(a)| \leq |Att^-_{\mathfrak{F}^*_{\mathcal{KB}}}(a)|$. Suppose that $a \in \mathscr{A}^*$ is attacked in $\mathfrak{F}_{\mathcal{KB}}$. Let $B = \{b_1, \ldots, b_n\}$ be the set of all attackers of a in $\mathfrak{F}_{\mathcal{KB}}$. Without loss of generality, let $B \cap \mathscr{A}^* = \{b_k, \ldots, b_n\}$. Let us define a function $f : B \to \{B' \subseteq \mathscr{A}^* \mid (B', a) \in \mathscr{R}^*\}$ as follows:

$$f(b_i) = \begin{cases} \{b_i\} & \text{if } b_i \in \mathscr{A}^* \\ B' = \{b_i^1, \ldots, b_i^k\} \text{ where } B \text{ is an arbitrary set such} & \\ \text{that for every } b_i^j \in B', Supp(b_i^j) = Supp(b_i) \text{ and} & \\ \bigcup_{b_i^j \in B'} Conc(b_i^j) = Conc(b_i) \text{ and } B' \text{ is a minimal} & \\ \text{set satisfying the previous two conditions} & \text{otherwise} \end{cases}$$

f is well defined since such B' exists if $b_i \notin \mathscr{A}^*$, by definition of $\mathscr{A}^*$. Let us prove that f is an injective function. Let $b_i, b_j \in B$. The case $i \geq k$ or $j \geq k$ is obvious. In the remainder of the proof, we suppose that $i < k$ and $j < k$. Denote $B' = f(b_i)$ and $B'' = f(b_j)$. Suppose that $B' = B''$. Thus $Supp(b_i) = Supp(b_j)$. Note that $Conc(b_i) =$

$\bigcup_{b \in B'} Conc(b) = \bigcup_{b \in B''} Conc(b) = Conc(b_j)$. Thus $b_i = b_j$. This shows that f is injective. We conclude that $|Att^-_{\mathfrak{F}_{\mathcal{KB}}}(a)| \leq |Att^-_{\mathfrak{F}^*_{\mathcal{KB}}}(a)|$.

□

7.2.2 Chapter 4

Proposition 4.1 (Preferred & Stable Characterisation) Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, $\mathfrak{F}^\star{}_{\mathcal{KB}}$ be the corresponding argumentation framework and $x \in \{s, p\}$. It holds that:

$$Ext_x(\mathfrak{F}^\star{}_{\mathcal{KB}}) = \{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\}$$

Proof 4.1. The plan of the proof is as follows:

1. We prove that $\{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\} \subseteq Ext_s(\mathfrak{F}^\star{}_{\mathcal{KB}})$.
2. We prove that $Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}}) \subseteq \{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\}$.
3. Nielsen and Parsons [2007] proved that every stable extension is a preferred one, we can thus proceed as follows. From the first item, we have that $\{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\} \subseteq Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}})$, thus the theorem holds for preferred semantics. From the second item we have that $Ext_s(\mathfrak{F}^\star{}_{\mathcal{KB}}) \subseteq \{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\}$, thus the theorem holds for stable semantics.

1. We first show $\{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\} \subseteq Ext_s(\mathfrak{F}^\star{}_{\mathcal{KB}})$. Let $A' \in repairs(\mathcal{KB})$ and let $E = Arg^\star(A')$. Let us prove that $E \in Ext_s(\mathfrak{F}^\star{}_{\mathcal{KB}})$. We first prove that E is conflict-free. By means of contradiction we suppose the contrary, i.e. let $b \in E$ such that $(E, b) \in \mathscr{R}^\circ$. From the definition of $\mathscr{R}^\circ$, there exists $E' \subseteq E$ such that $(E', b) \in \mathscr{R}^\star$, i.e. $\varphi \in Prem(b)$ such that $(\bigcup_{e' \in E'} Conc(e')) \cup \{\varphi\}$ is R-inconsistent. Thus $(\bigcup_{e' \in E'} Prem(e')) \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent; consequently A' is $\mathcal{R}$-inconsistent, contradiction. Therefore E is conflict-free. Let us now prove that E attacks all arguments outside the set. Let $b \in Arg^\star(\mathcal{F}) \setminus Arg^\star(A')$ and let $\varphi \in Prem(b)$, such that $\varphi \notin A'$. Let $A'_c = \{a_1, \ldots, a_n\}$ be the set of all arguments of the form $a_i = \varphi_i$ where $\varphi_i \in A'$. We have $\varphi \notin A'$, so, due to the set inclusion maximality for the repairs, $A' \cup \{\varphi\}$ is $\mathcal{R}$-inconsistent. Therefore, $(A'_c, b) \in \mathscr{R}^\star$. Consequently, E is a stable extension.

2. We now need to prove that $Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}}) \subseteq \{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\}$. Let $E \in Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}})$ and let us prove that there exists a repair A' such that $E = Arg^\star(A')$. Let $S = Base^\star(E)$. Let us prove that S is $\mathcal{R}$-consistent. Aiming to a contradiction, suppose that S is $\mathcal{R}$-inconsistent. Let $S' \subseteq S$ be such that (1) S' is $\mathcal{R}$-inconsistent and (2) every proper set of S' is $\mathcal{R}$-consistent. Let us denote $S' = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$. Let

$a \in E$ be an argument such that $\varphi_n \in Prem(a)$. Let $Y = \{a_1, \ldots, a_{n-1}\}$ be a set of arguments of the form $a_i = \varphi_i$. It holds that $(Y, a) \in \mathscr{R}^\star$ because $\bigcup_{a_i \in Y} Prem(a_i)$ is $\mathcal{R}$-inconsistent (by definition of S') and $(\bigcup_{a_i \in Y} Conc(a_i)) \cup \{\varphi_n\}$ is $\mathcal{R}$-inconsistent. However, since E is an admissible set, it means that a is defended by E. Thus, we know that $(E, Y) \in \mathscr{R}^\circ$, i.e. there exists $a_j \in Y$ such that $(E, a_j) \in \mathscr{R}^\circ$. Since E is conflict-free, we deduce that $a_j \notin E$. Now consider $a' \in E$ such that $\varphi_j \in Prem(a')$, it holds that $(E, a') \in \mathscr{R}^\circ$, contradiction. So it must be that S is $\mathcal{R}$-consistent.

Let us now prove that there exists no $S' \subseteq \mathcal{F}$ such that $S \subset S'$ and S' is $\mathcal{R}$- consistent. We use the proof by contradiction. Thus, suppose that S is not a maximal $\mathcal{R}$-consistent subset of $\mathcal{F}$. Then, there exists $S' \in repairs(\mathcal{KB})$, such that $S \subset S'$.

We have that $E \subseteq Arg^\star(S)$, since $S = Base^\star(E)$. Denote $E' = Arg^\star(S')$. Since $S \subset S'$ then $Arg^\star(S) \subset E'$. Thus, $E \subset E'$. From the first part of the proof, $E' \in Ext_s(\mathfrak{F}^\star{}_{\mathcal{KB}})$. Consequently, $E' \in Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}})$. We also know that $E \in Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}})$. Contradiction, since no preferred set can be a proper subset of another preferred set. Thus, we conclude that $Base^\star(E) \in repairs(\mathcal{KB})$. Let us show that $E = Arg^\star(Base^\star(E))$. It must be that $E \subseteq Arg^\star(S)$. Also, we know (from the first part) that $Arg^\star(S)$ is a stable and a preferred extension, thus the case $E \subset Arg^\star(s)$ is not possible.

3. Now we know that the set $\{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\}$ in included in $Ext_s(\mathfrak{F}^\star{}_{\mathcal{KB}})$ and $Ext_p(\mathfrak{F}^\star{}_{\mathcal{KB}}) \subseteq \{Arg^\star(A') \mid A' \in repairs(\mathcal{KB})\}$. The theorem follows from those two facts, as explained at the beginning of the proof.

□

Proposition 4.7 (Attack upper-bound) Let $\mathcal{KB}$ be a knowledge base and $\mathfrak{F}^\star{}_{\mathcal{KB}} = (\mathscr{A}^\star, \mathscr{R}^\star)$ be the corresponding argumentation framework. If $|\mathscr{A}^\star| = n$ then $|\mathscr{R}^\star| \leq \sum_{i=1}^{n-1} \binom{n}{i}(n-i)$.

Proof 4.7. We show by induction that for every $|\mathscr{A}^\star| \geq 2$, we have $|\mathscr{R}^\star| \leq \sum_{i=1}^{n-1} \binom{n}{i}(n-i)$.

- **Base:** If $|\mathscr{A}^\star| = 2$ then we have that $|\mathscr{R}^\star| \leq \binom{2}{1}(2-1) = 2$
- **Inductive Step:** Let us denote by $E(\mathscr{A}^\star)$, the maximum number of attacks on the set of arguments $\mathscr{A}^\star$. Suppose that $\mathscr{A}^\star = \{a_1, \ldots, a_k\}$

and $E(\mathscr{A}^\star)$ is $\sum_{i=1}^{k-1}\binom{k}{i}(k-i)$. We show that $E(\mathscr{A}^\star \cup \{a_{k+1}\}) = \sum_{i=1}^{k}\binom{k+1}{i}(k+1-i)$. We have that $E(\mathscr{A}^\star \cup \{a_{k+1}\}) = E(\mathscr{A}^\star) + x + y$ where x is the maximum number of attacks from a subset of $\{a_1, \ldots, a_k\}$ to a_{k+1} and y is the maximum number of attacks from a subset of $\{a_1, \ldots, a_{k+1}\}$ containing a_{k+1} to an argument of $\{a_1, \ldots, a_k\}$. We have that $x = 2^k - 1$ and $y = \sum_{i=1}^{k}\binom{k}{i}i$. Thus:

$$E(\mathscr{A}^\star \cup \{a_{k+1}\}) = E(\mathscr{A}^\star) + 2^k - 1 + \sum_{i=1}^{k}\binom{k}{i}i$$

$$E(\mathscr{A}^\star \cup \{a_{k+1}\}) = \left[\sum_{i=1}^{k-1}\binom{k}{i}(k-i)\right] + 2^k - 1 + \sum_{i=1}^{k}\binom{k}{i}i$$

$$E(\mathscr{A}^\star \cup \{a_{k+1}\}) = \left[\sum_{i=1}^{k-1}\binom{k}{i}k\right] + k + 2^k - 1$$

$$E(\mathscr{A}^\star \cup \{a_{k+1}\}) = \left[\sum_{i=1}^{k-1}\binom{k}{i}k\right] + k + \left[\sum_{i=1}^{k}\binom{k}{i}\right]$$

$$E(\mathscr{A}^\star \cup \{a_{k+1}\}) = \left[\sum_{i=1}^{k-1}\binom{k}{i}(k+1)\right] + k + 1$$

$$E(\mathscr{A}^\star \cup \{a_{k+1}\}) = \left[\sum_{i=1}^{k-1}\binom{k+1}{i}(k+1-i)\right] + k + 1$$

$$E(\mathscr{A}^\star \cup \{a_{k+1}\}) = \left[\sum_{i=1}^{k}\binom{k+1}{i}(k+1-i)\right]$$

This concludes the proof by induction. □

Proposition 4.8 ($\mathfrak{F}^\star$ Indirect consistency) Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, ${\mathfrak{F}^\star}_{\mathcal{KB}}$ be the corresponding argumentation framework and $x \in \{s, p, g\}$. Then:

- for every $E \in Ext_x({\mathfrak{F}^\star}_{\mathcal{KB}}), Concs(E)$ is a $\mathcal{R}$-consistent.
- $Output_x({\mathfrak{F}^\star}_{\mathcal{KB}})$ is $\mathcal{R}$-consistent.

Proof 4.8. Let E be a stable or a preferred extension of ${\mathfrak{F}^\star}_{\mathcal{KB}}$. From Proposition 4.1, there exists a repair $A' \in repairs(\mathcal{KB})$ such that $E = Arg^\star(A')$.

Note that $Concs(E) = Sat_{\mathcal{R}\cup\mathcal{N}}(A')$ (this follows from the definition of the arguments). Formally, $Sat_{\mathcal{R}\cup\mathcal{N}}(Sat_{\mathcal{R}\cup\mathcal{N}}(A')) = Sat_{\mathcal{R}\cup\mathcal{N}}(Concs(E))$. Since $Sat_{\mathcal{R}\cup\mathcal{N}}$ is idempotent, this means that we have $Sat_{\mathcal{R}\cup\mathcal{N}}(A') = Sat_{\mathcal{R}\cup\mathcal{N}}(Concs(E))$. Since $Sat_{\mathcal{R}\cup\mathcal{N}}(A') \not\models \bot$, then $Sat_{\mathcal{R}\cup\mathcal{N}}(Concs(E)) \not\models \bot$ and $Concs(E)$ is $\mathcal{R}$-consistent.

Let us now consider the case of grounded semantics. Denote E_{GE} the grounded extension of $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$. We have seen that for every $E \in Ext_p(\mathfrak{F}^{\star}{}_{\mathcal{KB}})$, it holds that $Sat_{\mathcal{R}\cup\mathcal{N}}(Concs(E)) \not\models \bot$. Since the grounded extension is a subset of the intersection of all the preferred extensions, and since there is at least one preferred extension Nielsen and Parsons [2007], say E_1, then $E_{GE} \subseteq E_1$. Since $Sat_{\mathcal{R}\cup\mathcal{N}}(Concs(E)) \not\models \bot$ then $Sat_{\mathcal{R}\cup\mathcal{N}}(Concs(E_{GE})) \not\models \bot$ and $Concs(E_{GE})$ is $\mathcal{R}$-consistent.

Consider the case of stable or preferred semantics. Let us prove that $Sat_{\mathcal{R}\cup\mathcal{N}}(Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}})) \not\models \bot$. From the definition of the output, we have $Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}}) = \bigcap_{E\in Ext_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}})} Concs(E)$. Since every knowledge base has at least one repair then, there is at least one stable or preferred extension E. From the definition of the output, we have that $Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}}) \subseteq Concs(E)$. Since $Concs(E)$ is $\mathcal{R}$-consistent thus $Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}})$ is $\mathcal{R}$-consistent.

Note that since there is only one grounded extension, we can deduce that $Sat_{\mathcal{R}}(Output_g(\mathfrak{F}^{\star}{}_{\mathcal{KB}})) = Sat_{\mathcal{R}}(Concs(E_{GE}))$.

□

Proposition 4.9 ($\mathfrak{F}^{\star}$ Closure) Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base, $\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ be the corresponding argumentation framework and $x \in \{s, p, g\}$. Then:

- for every $E \in Ext_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}}), Concs(E) = Sat_{\mathcal{R}}(Concs(E)$
- $Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}}) = Sat_{\mathcal{R}}(Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}}))$

Proof 4.9. Let E be a preferred, a stable or the grounded extension. The proof is split in two parts:

- From the definition of the saturation, $Concs(E) \subseteq Sat_{\mathcal{R}}(Concs(E))$. We prove that $Sat_{\mathcal{R}}(Concs(E)) \subseteq Concs(E)$. Suppose that $\alpha \in Sat_{\mathcal{R}}(Concs(E))$. This means that there exists a minimal set $\{\alpha_1, \ldots, \alpha_k\} \subseteq Conc(E)$ and a sequence of rule applications such that α is produced from $\{\alpha_1, \ldots, \alpha_k\}$.

  Note that from Proposition 4.8, $\{\alpha_1, \ldots, \alpha_k\}$ is $\mathcal{R}$-consistent. Since $\alpha_1, \ldots, \alpha_k \in Concs(E)$ then there exist $a_1, \ldots, a_k \in E$ such that we have $Conc(a_1) = \alpha_1, \ldots, Conc(a_k) = \alpha_k$. Thus, there exists an argument a such that $Prem(a) = Prem(a_1) \cup \cdots \cup Prem(a_k)$ and $Conc(a) = \alpha$. Since E is a preferred, a stable or the grounded extension, Theorems 4.1 and 4.3 imply that there exists a set $S \subseteq \mathcal{F}$ such that $E = Arg^{\star}(S) = Arg^{\star}(Base^{\star}(E))$. From this observation and since $Supp(a) \subseteq Base^{\star}(E)$, we conclude that $a \in E$. Thus, $\alpha \in Concs(E)$, which ends the proof.

- In the case of grounded semantics, the result holds directly from the first part of the proposition. The reminder of the proof considers stable or preferred semantics. From the definition of $Sat_{\mathcal{R}}, Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}}) \subseteq Sat_{\mathcal{R}}(Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}}))$. So we need to prove that $Sat_{\mathcal{R}}(Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}})) \subseteq Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}})$. Let $\alpha \in Sat_{\mathcal{R}}(Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}}))$. Then there exist a minimal set $\{\alpha_1, \ldots, \alpha_k\} \subseteq Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}})$ and a sequence of rule applications such that α is produced from $\{\alpha_1, \ldots, \alpha_k\}$. Since $\alpha_1, \ldots, \alpha_k \in Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}})$ then for every $E \in Ext_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}})$, we have $\alpha_1, \ldots, \alpha_k \in Concs(E)$. Therefore for every $E \in Ext_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}}), \alpha \in Sat_{\mathcal{R}}(Concs(E))$. From the first part of the proof, $Sat_{\mathcal{R}}(Concs(E)) = Concs(E)$. Thus, for every $E \in Ext_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}}), \alpha \in Concs(E)$. Thus, $\alpha \in Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}})$.

□

7.2.3 Chapter 5

Proposition 5.2 (Attackers equivalence) Given a logical argumentation framework $\mathfrak{F}_{\mathcal{K}} = (\mathscr{A}, \mathscr{R})$ with $\mathscr{A}$ being the set of arguments defined by Definition 3.1 and $\mathscr{R}$ the set of attacks defined according to Definition 3.2, the set $\mathscr{R}$ enjoys the following properties:

1. $C1b$: for every $a, b, c \in \mathscr{A}$, if $Conc(a) \cong Conc(b)$ then $((a, c) \in \mathscr{R}$ if and only if $(b, c) \in \mathscr{R})$.

2. $C2b$: for every $a, b, c \in \mathscr{A}$, if $Supp(a) \cong Supp(b)$ then $((c, a) \in \mathscr{R}$ if and only if $(c, b) \in \mathscr{R})$.

Proof 5.2. Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base expressed using existential rules and $\mathfrak{F}_{\mathcal{KB}} = (\mathscr{A}, \mathscr{R})$ the corresponding argumentation framework. Now, we consider $a, b, c \in \mathscr{A}$.

1. Suppose that $Conc(a) \cong Conc(b)$. If $(a, c) \in \mathscr{R}$, it means that there exists $\phi \in Supp(c)$ such that $Sat_{\mathcal{R}\cup\mathcal{N}}(Conc(a) \cup \{\phi\}) \models \bot$. However, since $Conc(a) \cong Conc(b)$, we can infer that $Sat_{\mathcal{R}}(Conc(a)) = Sat_{\mathcal{R}}(Conc(b))$, thus $Sat_{\mathcal{R}\cup\mathcal{N}}(Conc(b) \cup \{\phi\}) \models \bot$ and $(b, c) \in \mathscr{R}$. Likewise, $(b, c) \in \mathscr{R}$ implies $(a, c) \in \mathscr{R}$ which ends the proof.

2. Suppose now that $Supp(a) \cong Supp(b)$. If $(c, a) \in \mathscr{R}$, it means that there exists $\phi \in Supp(a)$ such that $Sat_{\mathcal{R}\cup\mathcal{N}}(Conc(c) \cup \{\phi\}) \models \bot$. However, since $Supp(a) \cong Supp(b)$, by definition, we have that there exists $\phi' \in Supp(b)$ s.t. $\phi' \asymp \phi$, i.e. $Sat_{\mathcal{R}}(\phi') = Sat_{\mathcal{R}}(\phi)$. Therefore, we can infer that $Sat_{\mathcal{R}}(Conc(c) \cup \{\phi'\}) \models \bot$ and $(c, b) \in \mathscr{R}$. Likewise, $(c, b) \in \mathscr{R}$ implies $(c, a) \in \mathscr{R}$ which ends the proof.

□

Proposition 5.4 (Core construction equivalence) Let $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ be an argumentation framework and $\bowtie_1, \bowtie_2$ be the equivalence relations defined in Definition 5.2. It holds that:

$$Core_{\bowtie_2}(\mathfrak{F}) = \bigcup_{c_1 \in Core_{\bowtie_1}(\mathfrak{F})} Core_{\bowtie_2}(c_1).$$

Proof 5.4. This proof will be split in two parts:

- ($\subseteq$) We prove this inclusion by construction. Let $c' = (\mathscr{A}', \mathscr{R}') \in Core_{\bowtie_2}(\mathfrak{F})$, by definition, for every $G \in \mathscr{A}/\bowtie_1$, we chose a unique x in G for c_1. Here, if there exists $a \in G \cap A'$ then we choose $x = a$, otherwise we choose a random element of G. Now that we have a specific core c_1 of $\mathscr{A}$ for $\bowtie_1$, we repeat the process and construct c_2 from c_1. In the end, $c_2 = c'$.

- ($\supseteq$) Let $c_1 = (\mathscr{A}_1, \mathscr{R}_1) \in Core_{\bowtie_1}(\mathfrak{F})$ and $c_2 = (\mathscr{A}_2, \mathscr{R}_2) \in Core_{\bowtie_2}(c_1)$. We prove that $c_2 \in Core_{\bowtie_2}(\mathfrak{F})$. We will proceed by proving each parts of Definition 5.4.

  - Since c_2 is a core of c_1 for equivalence relation $\bowtie_2$, then $\mathscr{A}_2 \subseteq \mathscr{A}_1$. Likewise, since c_1 is a core of $\mathfrak{F}$, we have $\mathscr{A}_1 \subseteq \mathscr{A}$. Finally, we have that $\mathscr{A}_2 \subseteq \mathscr{A}$.
  - Let $f : \mathscr{A}_1/\bowtie_2 \rightarrow \mathscr{A}/\bowtie_2$ a function that takes as input an element $\bar{x}$ of $\mathscr{A}_1/\bowtie_2$ and returns an element $\bar{\bar{x}}$ of $\mathscr{A}/\bowtie_2$ s.t. $\bar{x} \cap \bar{\bar{x}} \neq \emptyset$. We will show that this function is a bijection.
    * *Injective:* Suppose that there exists $\bar{x}, \bar{y} \in \mathscr{A}_1/\bowtie_2$ s.t. $f(\bar{x}) = f(\bar{y}) = \bar{\bar{z}}$ and $\bar{x} \neq \bar{y}$. By definition, it means that $\bar{\bar{z}} \cap \bar{x} \neq \emptyset$ and $\bar{\bar{z}} \cap \bar{y} \neq \emptyset$. Let $z_1 \in \bar{\bar{z}} \cap \bar{x}$ and $z_2 \in \bar{\bar{z}} \cap \bar{y}$. Since $z_1, z_2 \in \bar{\bar{z}}$, we have that $z_1 \bowtie_2 z_2$, contradiction with $\bar{x} \neq \bar{y}$.
    * *Surjective:* We have to prove that for every $\bar{\bar{x}} \in \mathscr{A}/\bowtie_2$, there exists $\bar{x} \in \mathscr{A}_1/\bowtie_2$ s.t. $f(\bar{x}) = \bar{\bar{x}}$. Suppose that there is no $\bar{x} \in \mathscr{A}_1/\bowtie_2$ s.t. $f(\bar{x}) = \bar{\bar{x}}$. Let us consider an argument $x \in \bar{\bar{x}}$. Then, there exists $G_1 \in \mathscr{A}/\bowtie_1$ s.t. $x \in G_1$. Furthermore, there exists $x_1 \in G_1$ s.t. $x_1 \in \mathscr{A}_1$. Keep in mind that $x_1 \in \bar{\bar{x}}$ since $x_1 \bowtie_1 x$ and thus $x_1 \bowtie_2 x$. Now, let $G \in A_1/\bowtie_2$ s.t. $x_1 \in G$. By definition of the core, there is a unique $x_2 \in G \cap A_2$ and $x_2 \bowtie_2 x_1$ thus, $x_2 \in \bar{\bar{x}}$, contradiction.

    Since c_2 is a core of c_1 for $\bowtie_2$, we have that for every $\bar{x} \in \mathscr{A}_1/\bowtie_2$, there is a unique $x \in \bar{x} \cap \mathscr{A}_2$. But now, since f is a bijection, we can easily conclude that for every $\bar{\bar{x}} \in \mathscr{A}/\bowtie_2$, there is a unique $x \in \bar{\bar{x}} \cap \mathscr{A}_2$.
  - The final point is obvious since it is only a restriction of attacks.

This ends the proof. □

Proposition 5.6 (Not different core characterisation $\bowtie_1$) Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a KB and $\mathfrak{F}_{\mathcal{KB}}$ be the corresponding argumentation framework. We have that $Core_{\bowtie_1}(\mathfrak{F}_{\mathcal{KB}}) = \{\mathfrak{F}_{\mathcal{KB}}\}$ iff for all $\mathcal{R}$-consistent subset Y of $\mathcal{F}$, there is no y_1, y_2 such that $Sat_{\mathcal{R}}(Y) \models y_1, Sat_{\mathcal{R}}(Y) \models y_2$, $y_1 \neq y_2$ and $y_1 \asymp y_2$.

Proof 5.6. We divide this proof in two parts:

- ($\Rightarrow$) We show that contrapositive of this implication is true by *reductio ad absurdum*. Suppose that there is a $\mathcal{R}$-consistent subset Y of $\mathcal{F}$ and there exists y_1, y_2 such that $Sat_{\mathcal{R}}(Y) \models y_1, Sat_{\mathcal{R}}(Y) \models y_2$, $y_1 \neq y_2$, $y_1 \asymp y_2$ and $Core_{\bowtie_1}(\mathfrak{F}_{\mathcal{KB}}) = \{\mathfrak{F}_{\mathcal{KB}}\}$. Let us consider $Y'' \subseteq Y$ s.t. there is no $Y' \subset Y''$ and $Sat_{\mathcal{R}} \models y_1$. We have that $a = (Y'', \{y_1\})$ and $b = (Y'', \{y_2\})$ are two arguments of $\mathfrak{F}_{\mathcal{KB}}$. Furthermore, we have that $a \bowtie_1 b$ meaning that $\mathfrak{F}_{\mathcal{KB}} \notin Core_{\bowtie_1}(\mathfrak{F}_{\mathcal{KB}})$, contradiction.

- ($\Leftarrow$) We show that this implication is true by *reductio ad absurdum*. Suppose that $Core_{\bowtie_1}(\mathfrak{F}_{\mathcal{KB}}) \neq \{\mathfrak{F}_{\mathcal{KB}}\}$. It means that there exists $c_1 = (\mathscr{A}_1, \mathscr{R}_1) \in Core_{\bowtie_1}(\mathfrak{F}_{\mathcal{KB}})$ with $X_{c_1} \neq \emptyset$. Therefore, it exists an argument $x \in X_{c_1}$ s.t. $x \in \mathscr{A}$ and $x \notin \mathscr{A}_1$. We deduce that there exists $x' \in \mathscr{A}_1$ s.t. $Conc(x) \cong Conc(x'), Supp(x) = Supp(x')$. By definition of an argument, we have that $Conc(x) \subseteq Sat_{\mathcal{R}}(Supp(x))$ and $Conc(x') \subseteq Sat_{\mathcal{R}}(Supp(x'))$, contradiction.

This ends the proof. □

Proposition 5.7 (Not different core characterisation $\bowtie_2$) Let $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a KB and $\mathfrak{F}_{\mathcal{KB}}$ be the corresponding argumentation framework. We have $Core_{\bowtie_2}(\mathfrak{F}_{\mathcal{KB}}) = \{\mathfrak{F}_{\mathcal{KB}}\}$ iff the two following items are satisfied:

- there is no $f_1, f_2 \in \mathcal{F}$ such that $f_1 \asymp f_2$ and $f_1 \neq f_2$
- for all $\mathcal{R}$-consistent subset $Y \subseteq \mathcal{F}$, there is no y_1, y_2 such that $Sat_{\mathcal{R}}(Y) \models y_1, Sat_{\mathcal{R}}(Y) \models y_2, y_1 \neq y_2$ and $y_1 \asymp y_2$.

Proof 5.7. We divide this proof in two parts:

- ($\Rightarrow$) We show that contrapositive of this implication is true by *reductio ad absurdum* in the same fashion as the proof of Proposition 5.6. Indeed, following the same reasoning, we can deduce that there exists two arguments $a, b \in \mathfrak{F}_{\mathcal{KB}}$ s.t. $a \bowtie_1 b$ and thus $a \bowtie_2 b$. It means that $\mathfrak{F}_{\mathcal{KB}} \notin Core_{\bowtie_2}(\mathfrak{F}_{\mathcal{KB}})$, contradiction.

- ($\Leftarrow$) We show that this implication is true by *reductio ad absurdum*. Suppose that $Core_{\bowtie_2}(\mathfrak{F}_{\mathcal{KB}}) \neq \{\mathfrak{F}_{\mathcal{KB}}\}$. It means that there exists $c_2 = (\mathscr{A}_2, \mathscr{R}_2) \in Core_{\bowtie_2}(\mathfrak{F}_{\mathcal{KB}})$ with $X_{c_2} \neq \emptyset$. Therefore, there exists an argument $x \in X_{c_2}$ s.t. $x \in \mathscr{A}$ and $x \notin \mathscr{A}_2$. It means that there exists $x' \in \mathscr{A}_2$ s.t. $Conc(x) \cong Conc(x'), Supp(x) \cong Supp(x')$. We can consider two cases which both lead to contradictions:

- If $Supp(x) \neq Supp(x')$, there exists $a \in Supp(x)$ and $a \notin Supp(x')$ (resp. $a \notin Supp(x)$ and $a \in Supp(x')$). Since we have $Supp(x) \cong Supp(x')$, there exists $a' \in Supp(x')$ (resp. $a' \in Supp(x)$) s.t. $a \asymp a'$, contradiction.
- If $Supp(x) = Supp(x')$, then since $Conc(x) \subseteq Sat_{\mathfrak{R}}(Supp(x))$ and $Conc(x') \subseteq Sat_{\mathfrak{R}}(Supp(x'))$, contradiction

This ends the proof. □

Proposition 5.16 (Dung Equivalent Existence) Let $\mathfrak{F} = (\mathscr{A}^\star, \mathscr{R}^\star)$ be an hypergraph argumentation frameworks. Then, there exists an abstract argumentation framework $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$, a subset $\{a_1, \ldots, a_m\} \subseteq \mathscr{A}$ and a bijection $\gamma : \mathscr{A}^\star \to \{a_1, \ldots, a_m\}$ such that for every $a \in \mathscr{A}^\star$, $C(a) = C'(\gamma(a))$.

Proof 5.16. This proof is split in two parts:

1. First, we exhibit how to build an abstract argumentation framework $\mathfrak{F}$ from an arbitrary hypergraph argumentation framework $\mathfrak{F}^\star$.
2. Second, we show that the aforementioned $\mathfrak{F}$ is such that Proposition 5.16 is satisfied.

1- Let $\mathfrak{F}^\star = (\mathscr{A}^\star, \mathscr{R}^\star)$ be an arbitrary hypergraph argumentation framework. We build the abstract argumentation $\mathfrak{F} = (\mathscr{A}, \mathscr{R})$ such that:

- $\mathscr{A} = \{a_1, \ldots, a_m\} \cup \{a_{m+1}, \ldots, a_n\}$
- There is a bijection γ_1 from $\mathscr{A}^\star$ to $\{a_1, \ldots, a_m\}$ and a bijection γ_2 from $\mathscr{R}^\star$ to $\{a_{m+1}, \ldots, a_n\}$.
- For every $(S, t) \in \mathscr{R}^\star$, we create the attack $(\gamma_2((S, t)), \gamma_1(t))$ in $\mathscr{R}$.
- For every $(S, t) \in \mathscr{R}^\star$, we consider the set $X_S = \{s \in S \mid$ for every $s' \in S, s' \geq^{nh}_{\mathfrak{F}^\star} s\}$ and we chose $x_S \in X_S$. Then, for every $S' \in \mathscr{R}^{\star -}_1(x_S)$, we create the attack $(\gamma_2((S', x_S)), \gamma_2((S, t)))$ in $\mathscr{R}$.

2- We now show that the constructed argumentation framework $\mathfrak{F}$ is such that Proposition 5.16 is satisfied. Let $X' = \{a_{m+1}, \ldots, a_n\}$ and $\mathfrak{F}' = (\mathscr{A}', \mathscr{R}')$ be the restriction of $\mathfrak{F}$ to X', i.e. $\mathscr{A}' = X'$ and $\mathscr{R}' = \{(a, b) \in \mathscr{R} \mid a \in X'$ and $b \in X'\}$.

We now define the function $f : X' \to \mathbb{R}$ that returns for every element a in X', the minimum score of the attackers associated with the attack corresponding to a. Namely, for every $a \in X'$, $f(a) = \min_{a' \in W} C(a')$ where $(W, t) = \gamma_2^{-1}(a)$ and γ_2^{-1} is the inverse function of γ_2.

We now prove that for every $a \in \mathscr{A}'$, it holds that $f(a) = \frac{1}{1 + \sum_{(b,a) \in \mathscr{R}'} f(b)}$.

Let a be an arbitrary argument in $\mathscr{A}'$, $\{b_1, \ldots, b_p\}$ the set of attackers

of a in $\mathfrak{F}'$ and $(W,t) = \gamma_2^{-1}(a)$. By construction, we know that there exists $x_W \in W$ such that $\{b_1,\ldots,b_p\} = \{\gamma_2((S',x_W))|S' \in \mathscr{R}^{\star}{}_1^{-}(x_W)\}$. Thus,

$$
\begin{aligned}
f(a) &= \min_{a' \in W} C(a') \\
f(a) &= C(x_W) \\
f(a) &= \frac{1}{1 + \sum\limits_{S' \in \mathscr{R}^{\star}{}_1^{-}(x_W)} \min\limits_{s' \in S'} C(s')} \\
f(a) &= \frac{1}{1 + \sum\limits_{(b,a) \in \mathscr{R}'} f(b)}
\end{aligned}
$$

Since f satisfies the same formula as h-categoriser then f is identical to h-categoriser on $\mathfrak{F}'$ Pu et al. [2014].

Let us now consider the whole argumentation graph $\mathfrak{F}$. Since there are no attacks from arguments in $\{a_1,\ldots,a_m\}$ to arguments in $\{a_{m+1},\ldots,a_n\}$, the h-categoriser gives the same scores to $\{a_{m+1},\ldots,a_n\}$ in $\mathfrak{F}$ and in $\mathfrak{F}'$. Since for every argument a in $\{a_1,\ldots,a_m\}$, the set of attackers of a in $\mathfrak{F}$ is $\{b \in \{a_{m+1},\ldots,a_n\} \mid \gamma_2^{-1}(b) = (S,\gamma_1^{-1}(a))$ with $S \in \mathscr{R}^{\star}{}_1^{-}(\gamma_1^{-1}(a))\}$, we have that for every $a \in \{a_1,\ldots,a_m\}$:

$$
\begin{aligned}
C'(a) &= \frac{1}{1 + \sum\limits_{b \in \mathscr{R}_1^{-}(a)} C'(b)} \\
C'(a) &= \frac{1}{1 + \sum\limits_{b \in \mathscr{R}_1^{-}(a)} f(b)} \\
C'(a) &= \frac{1}{1 + \sum\limits_{b \in \mathscr{R}_1^{-}(a)} \min\limits_{a' \in W} C(a')} \text{ where } (W,t) = \gamma_2^{-1}(b) \\
C'(a) &= \frac{1}{1 + \sum\limits_{S \in \mathscr{R}^{\star}{}_1^{-}(\gamma_1^{-1}(a))} \min\limits_{s \in S} C(s)} \\
C'(a) &= C(\gamma_1^{-1}(a))
\end{aligned}
$$

This concludes the proof.

□

Index

Symbols

$\mathscr{R}^{\star}{}^{-}_{n}(a)$ 132
$Arg(X, \mathscr{A}^*)$ 69
Att^+ 33
$Att^-_{\mathfrak{F}}$ 33
$Base(X)$ ..... *see* Base
Bur ..... *see* Burden vector
C ..... *see* nh-categorise function
C' ..... *see* h-categoriser function
$Conc(a)$ ..... *see* Conclusion
$Concs(X)$ 99
$Core_{\bowtie}(\mathfrak{F})$ 119
$D(\mathfrak{F}_{\mathcal{KB}})$ 67, 72
$D_{\mathfrak{F}}$ 36
Dis ..... *see* Discussion count
$Ext_x(\mathfrak{F})$ 36
$Ext_x(\mathfrak{F}^{\star})$ 95
$Ext_x(\mathfrak{F}^*_{\mathcal{KB}})$ 69
$Free(\mathcal{KB})$ 31
$I(\mathcal{KB}, C)$ *see* Inconsistency measure
$I^{\mathcal{KB}}(C)$ .. *see* Inconsistency measure
$I^{\mathcal{KB}}_d(X)$ 146
$I^{\mathcal{KB}}_{MI}(X)$ 146
$Imp(C)$ 61
$J_{c'}$ 128
$K(X)$ 125
$Output_x(\mathfrak{F}^{\star}{}_{\mathcal{KB}})$ 99
$Q()$. *see* Boolean conjunctive query
$Q(\vec{X})$ ..... *see* Conjunctive query
$SCC(\mathfrak{F})$ 36
$Supp(a)$ ..... *see* Support
$W_{a'}$ 57
$X/\bowtie$ ..... *see* Equivalence classes
X_c 126
$\mathscr{A}$ 32
AC_0 26
$\mathcal{ARL}$ ..... *see* ARL
$\mathfrak{F}$ 32
$\mathfrak{F}_{\mathcal{KB}}$ 51
$\mathfrak{F}^{\star}$ 92
$\mathfrak{F}^{\star}{}_{\mathcal{KB}}$ 92
$\mathfrak{F}^*_{\mathcal{KB}}$ 68
$\mathscr{A}^*$ .. *see* Filtrated set of arguments
$\mathscr{A}^{\star}$ 92
$\mathfrak{K}_{\mathfrak{F}^{\star}}$ 92
$Arg^{\star}(X)$ 93
$Base^{\star}(X)$ 93
$\mathscr{R}$ 32
$\mathscr{R}^*$ 68
$\mathscr{R}^-_n(a)$ 33
$\mathscr{R}^{\star}$ 92
$\mathscr{R}^{\circ}$ 94
Sat ..... *see* Saturated set of facts
$\mathcal{C}$ ..... *see* Constant
$\mathcal{D}$ 15
$repairs(\mathcal{KB})$ 31
$MI(\mathcal{KB})$ 31
$\mathcal{F}$ 18
$\mathcal{F}^*$ ..... *see* Saturated set of facts
$\mathcal{I}$ 15
$\mathcal{KB}$ 30
$\mathcal{KB}_s$ 30
$\mathcal{L}$ ..... *see* Language
Λ 38
$\mathrm{L}^{\geq}_{leximax}$ 147
$\mathrm{L}(X, \geq)$ ..... *see* Lifting function
$\mathrm{L}^{\geq}(X)$ ..... *see* Lifting function
$\mathcal{N}$ 28
$Nulls$ 14
$\mathcal{P}$ ..... *see* Predicate
$\mathcal{R}$ 19
$\mathcal{R}$-inconsistence 30
$\mathcal{RIF}$ ..... *see* RIF
$\mathcal{RIF}$ abstraction 151
$\mathfrak{V}(\mathcal{KB}, b)$ ... *see* Inconsistency value
$\mathfrak{V}^I(\mathcal{KB})$ 146
$\mathfrak{V}^I_f(\mathcal{KB})$. *see* Shapley inconsistency value
$\mathfrak{V}_a(\mathcal{KB})$ .... *see* Inconsistency value
$\mathcal{V}$ ..... *see* Variable
$\mathcal{V}oc$ ..... *see* Vocabulary
$\approx^{\sigma^{\star}}_{\mathfrak{F}^{\star}}$ 131
$\approx^{\sigma}_{\mathfrak{F}}$ 40
$\bowtie_1$ 118
$\bowtie_2$ 118
$Body(r)$ ..... *see* Body

$\perp$ 15
σ-chase 20
$\cong$ 117
δ ..... *see* Derivation
$\equiv$ ..... *see* Logical equivalence
$\asymp$ 117
$\exists$ 14
Facts 19
$\forall$ 14
Null ..... *see* Fresh variable
σ_{fr}*-chase* ..... *see* Frontier chase
σ_{fr} . *see* Frontier derivation reducer
Head(r) ..... *see* Head
Homo 19
$\mathcal{O}_x$ ..... *see* Orbit
$\mathfrak{C}'(X)$ 125
$\models_x^{\mathcal{RIF}}$ ..... *see* Restricted inference
$\models_{AR}$ ..... *see* AR
$\models_{IAR}$ ..... *see* IAR
$\models_{ICR}$ ..... *see* ICR
$\oplus$ 32, 92
π ..... *see* Homomorphism
π^{safe} 18
Rule 19
$\sigma^\star$ 131
`sort` 147
$\rightarrow$ 14
$\succ_{lex}$ ..... *see* Lexicographical order
$\succeq_{\mathfrak{F}^\star}^{\sigma^\star}$ 131
$\succeq_{\mathfrak{F}}^{\sigma}$ 40
$\succeq_{\mathfrak{F}}^{bur}$ ..... *see* Burden-based ranking semantics
$\succeq_{\mathfrak{F}}^{dis}$ .. *see* Discussion-based ranking semantics
$\succeq_{\mathfrak{F}}^{hcat}$ ..... *see* h-categoriser ranking-based semantics
$\succeq_{\mathcal{L}}^{\mathfrak{B}}$ 144
$\top$ 15
$\vDash$ ..... *see* Logical consequence
$\vec{a}$ 16
$\wedge$ 14
$a\mathscr{R}b$ 51
$fr(r)$ ..... *see* Frontier
terms(Φ) 15
vars(Φ) 15
coNP 26
ExpTime 26
NP 26
PTime 26
Datalog± 2
Datalog+ 2, 14

A

ABA 7, 47
Abstract argumentation framework 32
Abstraction 132
Abstraction-I 149
Abstraction-L 150
Acceptability 33, 68
Acyclic graph of rule dependency *see* aGRD
Addition of attack branch 134
Addition of defense branch 134
Admissible set ... 34, 37, 52, 68, 94
aGRD 24
Answer 17
AR 148
ArgSemSAT 83
ArgTools 83
Argument 32
Arity 14
ARL 123, 124
Artificial intelligence 1
ASGL 83
ASPARTIX-D 83
ASPIC+ 7, 8, 48, 91
Assumption-based argumentation *see* ABA
Atom 15
Atomic head 18
Attack 32
Attack branch 134
Attack vs full defense 135
Automorphism 60
Automorphisms inheritance 60
Automorphisms transfer 60

B

Base 50, 93
Bipolar argumentation framework 114
Body 18
Boolean conjunctive query . . 17, 21, 30
Bottom facts 149
Breadth-first derivation 20
Burden vector 43
Burden-based ranking-based semantics 43

C

Cardinality precedence . . . . . *see* CP
Cardinality-MI 150
Cf2 extension 36, 37, 54, 63
Chase 20
Chase finiteness order 23
Closed atom 16
Closure 100
Coalition . . . . . *see* Extension
Coherent 52, 53
Combined complexity 26
Complete extension 35, 37
Complete problem 26
Complexity classes 26
Component-defeated 36
Conclusion 49, 50
Concrete classes 24
Conflict-based argument 55
Conflict-based complete graph . . 55
Conflict-freeness 34, 68, 94
Conflicting facts 27, 28
Conjunction 14, 15
Conjunctive query 17
Constant 14
CoQuiAAS 83
Core 116, 118
Core chase 23
Counter-transitivity 136
CP 126, 128, 133
Credulously accepted 36
Critical instance 25
Cycle 55, 98, 135

D

DAGGER 74
Data complexity 26
Data sensitive 150
Datalog 2
Deductive argument 50
Deductive argumentation . . 7, 9, 49
Defeasible logic programming . . . *see* DeLP
Defeasible reasoning 4
Defeasible rules 7
Defense branch 134
Defense precedence 133
DeLP 7, 9, 48
Density 82, 105
Derivation 18, 19
Derivation for a fact 19, 20
Derivation reducer 20
Description logics 2
Direct attacker 33, 132
Direct consistency 100
Direct defender 33, 132
Discussion count 42
Discussion-based ranking-based semantics 42
Distributed defense 133
Distributed-defense precedence . 133
DLGP 72, 74, 92, 106
DOT 106
Drastic Shapley inconsistency value 143, 146
Dummy argument 56, 57, 62

E

Entailment problem 16
Equivalence classes 118
Equivalence relation 118
Equivalent facts 117
Equivalent set of facts 117
Ex falso quodlibet 3
Exhaustive breadth-first derivation 20
Existence property 40
Existential quantifier . . . . . *see* ∃
Existential rules . . . . . *see* *Datalog*±

Extension 31, 33

F

Fact 16
Falsum ..... *see* ⊥
FES 23, 51
Filtrated argumentation framework 68
Filtrated set of arguments 67
Finite 52
Finite expansion set ..... *see* FES
Finite unification set ..... *see* FUS
First-order logic 1, 14
Flatness property 40
Flawed 149
Formula 15
Free fact 31, 57
Free property 151
Fresh variable 14
Frontier 18, 21
Frontier chase 21, 22
Frontier derivation reducer 21
Functional term 21
FUS 23

G

GAD 101
GBTS 24
Graal 72, 80
Graph of rule dependency 24
GRD *see* Graph of rule dependency
Greedy bounded tree-width set . *see* GBTS
Ground atom 15
Grounded extension ..35, 37, 94, 97
Grounded labelling 39
Group comparison 135

H

H-categoriser function 41
H-categoriser ranking-based semantics 42
Hard problem 26
Head 18
Homomorphism 16, 17
Homomorphism restriction 17
Hypergraph argumentation framework 92
Hypothesis ..... *see* Support

I

IAR 148
ICCMA 78
ICR 148
Ideal extension 35, 37
Implication ..... *see* →
Impossible set 61
In 38
Incoherence 3, 28, 29
Inconsistency 3, 28, 29
Inconsistency measure 143
Inconsistency value 143, 144
Inconsistency-tolerant inference 143, 144
Inconsistent knowledge base 30
Increase of attack branch 134
Increase of defense branch 135
Independence 132
Indirect consistency 99
Infinite derivation 19
Infinite frontier chase 22
Interpretation 15
Interpretation domain ..... *see* $\mathcal{D}$
Interpretation function ..... *see* $\mathcal{I}$
Isomorphism 128, 132

J

Joint acyclicity 25

K

K-copy graph 57
Killing property 40
Knowledge representation 1

L

Labelling 33, 38
LabSATSolver 83
Language 7, 14
Lexicographical order 41
Leximax ..... *see* $\mathrm{L}^{\geq}_{leximax}$
Lifting function 143, 144

Logic-based argumentation 4
Logical consequence 16
Logical equivalence 16

M

Mapping 16
MCS . . . . . *see* Repair
MFA 25
MI Shapley inconsistency value 143, 146
Minimal inconsistent set 31
Minimality property 149
Model 16
Model faithful acyclicity . . *see* MFA
More restrictive equivalence relation 122
Multiple derivations 49

N

N-append 149
NaE 126, 130, 133
Naive semantics 34, 62
NAKED 90, 107
Negative constraint 27, 28
Nh-categoriser 136
Nh-categoriser function 136
Nh-categoriser ranking-based semantics 140
Non entailment ejection 151
Non-attacked equivalence . *see* NaE
Non-trivial 52
Null . . . . . *see* Fresh variable

O

OBDA 30, 142
Oblivious-FES 23
Ontology-based data access . . . . . *see* OBDA
Orbit 59
Orbit cycle 59
Out 38

P

Path 33, 131
Permutation 59
Positive rule . . . . . *see* Rule
Predicate 14
Preferred extension . . 34, 37, 54, 68, 94, 97
Productivity 151

Q

Quality precedence 135
Query entailment 13, 21

R

R-append 149
Range restricted 24
Ranking-based semantics . . . 39, 40, 114
Rationality postulates 99
Reducer order relation 23
Reinstatement labelling 38
Rejected argument 36, 53
Relational databases 1
Relatively grounded 52, 54
Repair 4, 31, 95
Repair semantics 4
Restricted chase 23
Restricted inference 148
Restricted-FES 23
RIF 143, 144
Rule 17
Rule application 18
Rule-based logics 7

S

Saturated set of attacks . . . . *see* $\mathscr{R}^\circ$
Saturated set of facts 21
Sceptically accepted 36
Scoring function 40, 41
SCT 126, 130, 136
Self-attacking argument 54, 98
Self-contradiction 133
Semi-stable extension 35, 37, 54
Sentinel 52
Set attack 33, 68, 94
Set defense 33, 68, 94
Set of attacking argument . . . 67, 91
Shapley inconsistency value 146

Simple defense 133
Skolem chase 21
Skolem term 16, 21
Skolem-FES 23–25
Skolemisation 21
Stable extension 34, 37, 69, 94
Stage semnatics 36
Strict addition of defense branch 134
Strict counter-transitivity . *see* SCT
Strongly connected 35
Strongly connected component . 36, 61, 83
Subgraph 58
Substitution 16
Support 49, 50
Support argumentation framework 114
Supremacy 151
Symmetry 59, 60

T

Tarskian logics 7
Term 15
TGD . . . . . *see* Rule
Total 133

U

Undec 38
Unique name assumption 15
Universal model 21
Universal quantifier . . . . . *see* ∀

V

Variable 14
Vocabulary 14
Void precedence 132

W

Weak acyclicity 25
Weighted argumentation framework 114
Well-founded 52, 54

Bibliography

Abiteboul, S., Hull, R., and Vianu, V. (1995). *Foundations of Databases.* Addison-Wesley. (p.)

Amgoud, L. (2014). Postulates for logic-based argumentation systems. *Int. J. Approx. Reasoning*, 55(9):2028–2048. (p.)

Amgoud, L. and Ben-Naim, J. (2013). Ranking-Based Semantics for Argumentation Frameworks. In *Scalable Uncertainty Management - 7th International Conference, SUM 2013, Washington, DC, USA, September 16-18, 2013. Proceedings*, pages 134–147. (p.)

Amgoud, L. and Ben-Naim, J. (2015). Argumentation-based Ranking Logics. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 1511–1519. (p.)

Amgoud, L. and Ben-Naim, J. (2016). Evaluation of Arguments from Support Relations: Axioms and Semantics. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 900–906. (p.)

Amgoud, L., Ben-Naim, J., Doder, D., and Vesic, S. (2016). Ranking Arguments With Compensation-Based Semantics. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016.*, pages 12–21. (p.)

Amgoud, L., Ben-Naim, J., Doder, D., and Vesic, S. (2017). Acceptability Semantics for Weighted Argumentation Frameworks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 56–62. (p.)

Amgoud, L. and Besnard, P. (2009). Bridging the Gap between Abstract Argumentation Systems and Logic. In *Scalable Uncertainty Management, Third International Conference, SUM 2009, Washington, DC, USA, September 28-30, 2009. Proceedings*, pages 12–27. (p.)

Amgoud, L. and Besnard, P. (2010). A Formal Analysis of Logic-Based Argumentation Systems. In *Scalable Uncertainty Management - 4th International Conference, SUM 2010, Toulouse, France, September 27-29, 2010. Proceedings*, pages 42–55. (p.)

Amgoud, L., Besnard, P., and Vesic, S. (2014). Equivalence in logic-based argumentation. *Journal of Applied Non-Classical Logics*, 24(3):181–208. (p.)

Amgoud, L., Caminada, M., Cayrol, C., Lagasquie-Schiex, M.-C., and Prakken, H. (2004). Towards a consensual formal model: inference part. (p.)

Amgoud, L. and Cayrol, C. (1998). On the Acceptability of Arguments in Preference-based Argumentation. In *UAI '98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, University of Wisconsin Business School, Madison, Wisconsin, USA, July 24-26, 1998*, pages 1–7. (p.)

Amgoud, L., Cayrol, C., Lagasquie-Schiex, M.-C., and Livet, P. (2008). On bipolarity in argumentation frameworks. *Int. J. Intell. Syst.*, 23(10):1062–1093. (p.)

Amgoud, L., Maudet, N., and Parsons, S. (2000). Modelling dialogues using argumentation. In *Proceedings Fourth International Conference on MultiAgent Systems*, pages 31–38, Boston, MA, USA. IEEE Comput. Soc. (p.)

Antoniou, G., Billington, D., Governatori, G., Maher, M. J., and Rock, A. (2000). A Family of Defeasible Reasoning Logics and its Implementation. In *ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, Germany, August 20-25, 2000*, pages 459–463. (p.)

Arenas, M., Bertossi, L. E., and Chomicki, J. (1999). Consistent Query Answers in Inconsistent Databases. In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 31 - June 2, 1999, Philadelphia, Pennsylvania, USA*, pages 68–79. (p.)

Arioua, A. (2016). *Formalizing and Studying Dialectical Explanations in Inconsistent Knowledge Bases.* PhD thesis. (p.)

Arioua, A., Croitoru, M., and Buche, P. (2016). DALEK: A Tool for Dialectical Explanations in Inconsistent Knowledge Bases. In *Computational Models of Argument - Proceedings of COMMA 2016, Potsdam, Germany, 12-16 September, 2016.*, pages 461–462. (p.)

Arioua, A., Croitoru, M., and Vesic, S. (2017). Logic-based argumentation with existential rules. *Int. J. Approx. Reasoning*, 90:76–106. (p.)

Baader, F., Horrocks, I., and Sattler, U. (2005). Description Logics as Ontology Languages for the Semantic Web. In *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, pages 228–248. (p.)

Baget, J.-F., Benferhat, S., Bouraoui, Z., Croitoru, M., Mugnier, M.-L., Papini, O., Rocher, S., and Tabia, K. (2016a). A General Modifier-Based Framework for Inconsistency-Tolerant Query Answering. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016.*, pages 513–516. (p.)

Baget, J.-F., Benferhat, S., Bouraoui, Z., Croitoru, M., Mugnier, M.-L., Papini, O., Rocher, S., and Tabia, K. (2016b). Inconsistency-Tolerant Query Answering: Rationality Properties and Computational Complexity Analysis. In *Logics in Artificial Intelligence - 15th European Conference, JELIA 2016, Larnaca, Cyprus, November 9-11, 2016, Proceedings*, pages 64–80. (p.)

Baget, J.-F., Garreau, F., Mugnier, M.-L., and Rocher, S. (2014a). Extending Acyclicity Notions for Existential Rules. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, pages 39–44. (p.)

Baget, J.-F., Garreau, F., Mugnier, M.-L., and Rocher, S. (2014b). Revisiting Chase Termination for Existential Rules and their Extension to Nonmonotonic Negation. *CoRR*, abs/1405.1071. (p.)

Baget, J.-F., Gutierrez, A., Leclère, M., Mugnier, M.-L., Rocher, S., and Sipieter, C. (2015a). Datalog+, RuleML and OWL 2: Formats and Translations for Existential Rules. In *Proceedings of the RuleML 2015 Challenge, the Special Track on Rule-based Recommender Systems for the Web of Data, the Special Industry Track and the RuleML 2015 Doctoral Consortium hosted by the 9th International Web Rule Symposium (RuleML 2015), Berlin, Germany, August 2-5, 2015.* (p.)

Baget, J.-F., Gutierrez, A., Leclère, M., Mugnier, M.-L., Rocher, S., and Sipieter, C. (2015b). DLGP: An extended Datalog Syntax for Existential Rules and Datalog+/- Version 2.0. (p.)

Baget, J.-F., Leclère, M., Mugnier, M.-L., Rocher, S., and Sipieter, C. (2015c). Graal: A Toolkit for Query Answering with Existential Rules. In *Rule Technologies: Foundations, Tools, and Applications - 9th International Symposium, RuleML 2015, Berlin, Germany, August 2-5, 2015, Proceedings*, pages 328–344. (p.)

Baget, J.-F., Leclère, M., Mugnier, M.-L., and Salvat, E. (2011). On rules with existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10):1620–1654. (p.)

Bao, Z., Cyras, K., and Toni, F. (2017). ABAplus: Attack Reversal in Abstract and Structured Argumentation with Preferences. In *PRIMA 2017: Principles and Practice of Multi-Agent Systems - 20th International Conference, Nice, France, October 30 - November 3, 2017, Proceedings*, pages 420–437. (p.)

Baroni, P., Caminada, M., and Giacomin, M. (2011). An introduction to argumentation semantics. *Knowledge Eng. Review*, 26(4):365–410. (p.)

Baroni, P., Romano, M., Toni, F., Aurisicchio, M., and Bertanza, G. (2015). Automatic evaluation of design alternatives with quantitative argumentation. *Argument & Computation*, 6(1):24–49. (p.)

Beeri, C. and Vardi, M. Y. (1981). The Implication Problem for Data Dependencies. In *Automata, Languages and Programming, 8th Colloquium, Acre (Akko), Israel, July 13-17, 1981, Proceedings*, pages 73–85. (p.)

Benferhat, S., Bouraoui, Z., Croitoru, M., Papini, O., and Tabia, K. (2016). Non-Objection Inference for Inconsistency-Tolerant Query Answering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3684–3690. (p.)

Benferhat, S., Dubois, D., and Prade, H. (1997). Some Syntactic Approaches to the Handling of Inconsistent Knowledge Bases: A Comparative Study Part 1: The Flat Case. *Studia Logica*, 58(1):17–45. (p.)

Bertossi, L. E., Hunter, A., and Schaub, T. (2005). Introduction to Inconsistency Tolerance. In *Inconsistency Tolerance [result from a Dagstuhl seminar]*, pages 1–14. (p.)

Besnard, P., García, A. J., Hunter, A., Modgil, S., Prakken, H., Simari, G. R., and Toni, F. (2014). Introduction to structured argumentation. *Argument & Computation*, 5(1):1–4. (p.)

Besnard, P. and Hunter, A. (2001). A logic-based theory of deductive arguments. *Artif. Intell.*, 128(1-2):203–235. (p.)

Besnard, P. and Hunter, A. (2008). *Elements of Argumentation.* MIT Press. (p.)

Besnard, P. and Hunter, A. (2014). Constructing argument graphs with deductive arguments: a tutorial. *Argument & Computation*, 5(1):5–30. (p.)

Bienvenu, M. (2012). On the Complexity of Consistent Query Answering in the Presence of Simple Ontologies. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.* (p.)

Birnbaum, L. (1982). Argument Molecules: A Functional Representation of Argument Structure. In *Proceedings of the National Conference on Artificial Intelligence, Pittsburgh, PA, USA, August 18-20, 1982.*, pages 63–65. (p.)

Birnbaum, L., Flowers, M., and McGuire, R. (1980). Towards an AI Model of Argumentation. In *Proceedings of the 1st Annual National Conference on Artificial Intelligence, Stanford University, CA, USA, August 18-21, 1980.*, pages 313–315. (p.)

Bondarenko, A., Toni, F., and Kowalski, R. A. (1993). An Assumption-Based Framework for Non-Monotonic Reasoning. In *LPNMR*, pages 171–189. (p.)

Bonet, B. and Geffner, H. (1996). Arguing for Decisions: A Qualitative Model of Decision Making. In *UAI '96: Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence, Reed College, Portland, Oregon, USA, August 1-4, 1996*, pages 98–105. (p.)

Bonzon, E., Delobelle, J., Konieczny, S., and Maudet, N. (2016). A Comparative Study of Ranking-Based Semantics for Abstract Argumentation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 914–920. (p.)

Bonzon, E., Delobelle, J., Konieczny, S., and Maudet, N. (2018). Combining Extension-Based Semantics and Ranking-Based Semantics for Abstract Argumentation. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018.*, pages 118–127. (p.)

Bourgaux, C. (2016). *Inconsistency Handling in Ontology-Mediated Query Answering.* PhD thesis, Université Paris-Saclay, Paris. (p.)

Brewka, G. (1989). Preferred Subtheories: An Extended Logical Framework for Default Reasoning. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence. Detroit, MI, USA, August 1989*, pages 1043–1048. (p.)

Calì, A., Gottlob, G., and Kifer, M. (2013). Taming the Infinite Chase: Query Answering under Expressive Relational Constraints. *J. Artif. Intell. Res.*, 48:115–174. (p.)

Calì, A., Gottlob, G., and Lukasiewicz, T. (2009). A general datalog-based framework for tractable query answering over ontologies. In *Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009, June 19 - July 1, 2009, Providence, Rhode Island, USA*, pages 77–86. (p.)

Calì, A., Gottlob, G., and Lukasiewicz, T. (2012). A general Datalog-based framework for tractable query answering over ontologies. *J. Web Sem.*, 14:57–83. (p.)

Calì, A., Gottlob, G., Lukasiewicz, T., Marnette, B., and Pieris, A. (2010a). Datalog+/-: A Family of Logical Knowledge Representation and Query Languages for New Applications. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 228–242. (p.)

Calì, A., Gottlob, G., and Pieris, A. (2010b). Advanced Processing for Ontological Queries. *PVLDB*, 3(1):554–565. (p.)

Calvanese, D., Giacomo, G. D., Lembo, D., Lenzerini, M., and Rosati, R. (2007). Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. *J. Autom. Reasoning*, 39(3):385–429. (p.)

Caminada, M. (2006). On the Issue of Reinstatement in Argumentation. In *Logics in Artificial Intelligence, 10th European Conference, JELIA 2006, Liverpool, UK, September 13-15, 2006, Proceedings*, pages 111–123. (p.)

Caminada, M. (2007). Comparing two unique extension semantics for formal argumentation: ideal and eager. In *Proceedings of the 19th Belgian-Dutch conference on artificial intelligence (BNAIC 2007)*, pages 81–87. Utrecht University Press. (p.)

Caminada, M. and Amgoud, L. (2007). On the evaluation of argumentation formalisms. *Artif. Intell.*, 171(5-6):286–310. (p.)

Caminada, M. and Pigozzi, G. (2011). On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102. (p.)

Caminada, M. W. A., Carnielli, W. A., and Dunne, P. E. (2012). Semi-stable semantics. *J. Log. Comput.*, 22(5):1207–1254. (p.)

Cayrol, C. and Lagasquie-Schiex, M.-C. (2005). Graduality in Argumentation. *J. Artif. Intell. Res. (JAIR)*, 23:245–297. (p.)

Ceri, S., Gottlob, G., and Tanca, L. (1989). What you Always Wanted to Know About Datalog (And Never Dared to Ask). *IEEE Trans. Knowl. Data Eng.*, 1(1):146–166. (p.)

Cerutti, F., Dunne, P. E., Giacomin, M., and Vallati, M. (2013). Computing Preferred Extensions in Abstract Argumentation: A SAT-Based Approach. In *Theory and Applications of Formal Argumentation - Second International Workshop, TAFA 2013, Beijing, China, August 3-5, 2013, Revised Selected papers*, pages 176–193. (p.)

Chandra, A. K., Lewis, H. R., and Makowsky, J. A. (1981). Embedded Implicational Dependencies and their Inference Problem. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing, May 11-13, 1981, Milwaukee, Wisconsin, USA*, pages 342–354. (p.)

Chein, M. and Mugnier, M.-L. (2009). *Graph-based Knowledge Representation - Computational Foundations of Conceptual Graphs.* Advanced Information and Knowledge Processing. Springer. (p.)

Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. *Commun. ACM*, 13(6):377–387. (p.)

Cohen, R. (1987). Analyzing the Structure of Argumentative Discourse. *Computational Linguistics*, 13(1-2):11–24. (p.)

Colmerauer, A. and Roussel, P. (1996). The birth of Prolog. In *History of programming languages—II*, pages 331–367. ACM, New York, 1 edition. (p.)

Coste-Marquis, S., Devred, C., and Marquis, P. (2005). Symmetric Argumentation Frameworks. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 8th European Conference, ECSQARU 2005, Barcelona, Spain, July 6-8, 2005, Proceedings*, pages 317–328. (p.)

Coste-Marquis, S., Konieczny, S., Marquis, P., and Ouali, M. A. (2012). Weighted Attacks in Argumentation Frameworks. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012.* (p.)

Croitoru, M., Thomopoulos, R., and Vesic, S. (2015). Introducing Preference-Based Argumentation to Inconsistent Ontological Knowledge Bases. In *PRIMA 2015: Principles and Practice of Multi-Agent Systems - 18th International Conference, Bertinoro, Italy, October 26-30, 2015, Proceedings*, pages 594–602. (p.)

Croitoru, M. and Vesic, S. (2013). What Can Argumentation Do for Inconsistent Ontology Query Answering? In *Scalable Uncertainty Management - 7th International Conference, SUM 2013, Washington, DC, USA, September 16-18, 2013. Proceedings*, pages 15–29. (p.)

Dantsin, E., Eiter, T., Gottlob, G., and Voronkov, A. (2001). Complexity and expressive power of logic programming. *ACM Comput. Surv.*, 33(3):374–425. (p.)

Delobelle, J. (2017). *Ranking-based Semantics for Abstract Argumentation. (Sémantique à base de Classement pour l'Argumentation Abstraite).* PhD thesis. (p.)

Deutsch, A., Nash, A., and Remmel, J. B. (2008). The chase revisited. In *Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, June 9-11, 2008, Vancouver, BC, Canada*, pages 149–158. (p.)

Dimopoulos, Y., Nebel, B., and Toni, F. (1999). Preferred Arguments are Harder to Compute than Stable Extension. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages*, pages 36–43. (p.)

Dung, P. M. (1995). On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artif. Intell.*, 77(2):321–358. (p.)

Dung, P. M., Kowalski, R. A., and Toni, F. (2006). Dialectic proof procedures for assumption-based, admissible argumentation. *Artif. Intell.*, 170(2):114–159. (p.)

Dung, P. M., Mancarella, P., and Toni, F. (2007). Computing ideal sceptical argumentation. *Artif. Intell.*, 171(10-15):642–674. (p.)

Dunne, P. E. and Bench-Capon, T. J. M. (2002). Coherence in finite argument systems. *Artif. Intell.*, 141(1/2):187–203. (p.)

Dunne, P. E., Hunter, A., McBurney, P., Parsons, S., and Wooldridge, M. (2011). Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artif. Intell.*, 175(2):457–486. (p.)

Dunne, P. E. and Wooldridge, M. (2009). Complexity of Abstract Argumentation. In *Argumentation in Artificial Intelligence*, pages 85–104. (p.)

Egly, U., Gaggl, S. A., and Woltran, S. (2008). ASPARTIX: Implementing Argumentation Frameworks Using Answer-Set Programming. In *Logic Programming, 24th International Conference, ICLP 2008, Udine, Italy, December 9-13 2008, Proceedings*, pages 734–738. (p.)

Fagin, R., Kolaitis, P. G., Miller, R. J., and Popa, L. (2005). Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124. (p.)

Flouris, G., Huang, Z., Pan, J. Z., Plexousakis, D., and Wache, H. (2006). Inconsistencies, Negations and Changes in Ontologies. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1295–1300. (p.)

Gabbay, D. M., Hogger, C. J., and Robinson, J. A., editors (1993). *Handbook of logic in artificial intelligence and logic programming.* Clarendon Press ; Oxford University Press, Oxford : Oxford ; New York. (p.)

Gabbay, D. M. and Rodrigues, O. (2015). Equilibrium States in Numerical Argumentation Networks. *Logica Universalis*, 9(4):411–473. (p.)

Gaggl, S. A. (2013). *A Comprehensive Analysis of the cf2 Argumentation Semantics: From Characterization to Implementation.* PhD thesis, TU Wien. (p.)

Gaggl, S. A. and Woltran, S. (2013). The cf2 argumentation semantics revisited. *J. Log. Comput.*, 23(5):925–949. (p.)

Gallaire, H. and Nicolas, J.-M. (1987). Logic Approach to Knowledge and Data Bases at ECRC. *IEEE Data Eng. Bull.*, 10(4):2–9. (p.)

García, A. J., Gollapally, D., Tarau, P., and Simari, G. R. (2000). Deliberative stock market agents using jinni and defeasible logic programming. In *esaw'00 engineering societies in the agents' world, workshop of ecai 2000.* (p.)

García, A. J. and Simari, G. R. (2004). Defeasible Logic Programming: An Argumentative Approach. *TPLP*, 4(1-2):95–138. (p.)

Gordon, T. F. and Karacapilidis, N. I. (1997). The Zeno Argumentation Framework. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Law, ICAIL '97, Melbourne, Vicoria, Australia, June 30 - July 3, 1997*, pages 10–18. (p.)

Grant, J. and Hunter, A. (2011). Measuring the Good and the Bad in Inconsistent Information. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2632–2637. (p.)

Grau, B. C., Horrocks, I., Krötzsch, M., Kupke, C., Magka, D., Motik, B., and Wang, Z. (2013). Acyclicity Notions for Existential Rules and Their Application to Query Answering in Ontologies. *J. Artif. Intell. Res.*, 47:741–808. (p.)

Grégoire, É., Mazure, B., and Piette, C. (2007). Boosting a Complete Technique to Find MSS and MUS Thanks to a Local Search Oracle. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2300–2305. (p.)

Hadjinikolis, C., Siantos, Y., Modgil, S., Black, E., and McBurney, P. (2013). Opponent Modelling in Persuasion Dialogues. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 164–170. (p.)

Hecham, A. (2018). *Defeasible reasoning for existential rules. (Raisonnement defaisable dans les règles existentielles).* PhD thesis. (p.)

Hecham, A., Bisquert, P., and Croitoru, M. (2017a). On the Chase for All Provenance Paths with Existential Rules. In *Rules and Reasoning - International Joint Conference, RuleML+RR 2017, London, UK, July 12-15, 2017, Proceedings*, pages 135–150. (p.)

Hecham, A., Croitoru, M., and Bisquert, P. (2017b). Argumentation-Based Defeasible Reasoning For Existential Rules. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, pages 1568–1569. (p.)

Hunter, A. and Konieczny, S. (2010). On the measure of conflicts: Shapley Inconsistency Values. *Artif. Intell.*, 174(14):1007–1026. (p.)

Konieczny, S., Marquis, P., and Vesic, S. (2015). On Supported Inference and Extension Selection in Abstract Argumentation Frameworks. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 13th European Conference, ECSQARU 2015, Compiègne, France, July 15-17, 2015. Proceedings*, pages 49–59. (p.)

Krötzsch, M. and Rudolph, S. (2011). Extending Decidable Existential Rules by Joining Acyclicity and Guardedness. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 963–968. (p.)

Lagniez, J.-M., Lonca, E., and Mailly, J.-G. (2015). CoQuiAAS: A Constraint-Based Quick Abstract Argumentation Solver. In *27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2015, Vietri sul Mare, Italy, November 9-11, 2015*, pages 928–935. (p.)

Lam, H.-P., Governatori, G., and Riveret, R. (2016). On ASPIC+ and Defeasible Logic. In *Computational Models of Argument - Proceedings of COMMA 2016, Potsdam, Germany, 12-16 September, 2016.*, pages 359–370. (p.)

Leite, J. and Martins, J. (2011). Social Abstract Argumentation. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2287–2292. (p.)

Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., and Savo, D. F. (2010). Inconsistency-Tolerant Semantics for Description Logics. In *Web Reasoning and Rule Systems - Fourth International Conference, RR 2010, Bressanone/Brixen, Italy, September 22-24, 2010. Proceedings*, pages 103–117. (p.)

Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., and Savo, D. F. (2015). Inconsistency-tolerant query answering in ontology-based data access. *J. Web Sem.*, 33:3–29. (p.)

Lembo, D. and Ruzzi, M. (2007). Consistent Query Answering over Description Logic Ontologies. In *Web Reasoning and Rule Systems, First International Conference, RR 2007, Innsbruck , Austria, June 7-8, 2007, Proceedings*, pages 194–208. (p.)

Levesque, H. J. and Brachman, R. J. (1987). Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3:78–93. (p.)

Lukasiewicz, T., Martinez, M. V., Pieris, A., and Simari, G. I. (2015). From Classical to Consistent Query Answering under Existential Rules. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 1546–1552. (p.)

Marnette, B. (2009). Generalized schema-mappings: from termination to tractability. In *Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009, June 19 - July 1, 2009, Providence, Rhode Island, USA*, pages 13–22. (p.)

Martinez, M. V., Deagustini, C. A. D., Falappa, M. A., and Simari, G. R. (2014). Inconsistency-Tolerant Reasoning in Datalog +/- Ontologies via an Argumentative Semantics. In *Advances in Artificial Intelligence - IBERAMIA 2014 - 14th Ibero-American Conference on AI, Santiago de Chile, Chile, November 24-27, 2014, Proceedings*, pages 15–27. (p.)

Modgil, S. and Prakken, H. (2013). A general account of argumentation with preferences. *Artif. Intell.*, 195:361–397. (p.)

Modgil, S. and Prakken, H. (2014). The ASPIC+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62. (p.)

Nielsen, S. H. and Parsons, S. (2006). Computing Preferred Extensions for Argumentation Systems with Sets of Attacking Arguments. In *Computational Models of Argument: Proceedings of COMMA 2006, September 11-12, 2006, Liverpool, UK*, pages 97–108. (p.)

Nielsen, S. H. and Parsons, S. (2007). A Generalization of Dung's Abstract Framework for Argumentation: Arguing with Sets of Attacking Arguments. In Maudet, N., Parsons, S., and Rahwan, I., editors, *Argumentation in Multi-Agent Systems*, pages 54–73. Springer Berlin Heidelberg. (p.)

Nofal, S., Atkinson, K., and Dunne, P. E. (2014). Algorithms for decision problems in argument systems under preferred semantics. *Artif. Intell.*, 207:23–51. (p.)

Onet, A. (2013). The Chase Procedure and its Applications in Data Exchange. In *Data Exchange, Integration, and Streams*, pages 1–37. (p.)

Ostrowski, M. and Schaub, T. (2012). ASP modulo CSP: The clingcon system. *TPLP*, 12(4-5):485–503. (p.)

Papadimitriou, C. (1994). *Computational Complexity.* Addison Wesley Pub. Co. (p.)

Pereira, C. d. C., Tettamanzi, A., and Villata, S. (2011). Changing One's Mind: Erase or Rewind? In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 164–171. (p.)

Poggi, A., Lembo, D., Calvanese, D., Giacomo, G. D., Lenzerini, M., and Rosati, R. (2008). Linking Data to Ontologies. *J. Data Semantics*, 10:133–173. (p.)

Pollock, J. L. (1987). Defeasible Reasoning. *Cognitive Science*, 11(4):481–518. (p.)

Pollock, J. L. (2001). Defeasible reasoning with variable degrees of justification. *Artif. Intell.*, 133(1-2):233–282. (p.)

Poole, D. (1985). On the Comparison of Theories: Preferring the Most Specific Explanation. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence. Los Angeles, CA, USA, August 1985*, pages 144–147. (p.)

Prakken, H. (2006). Formal systems for persuasion dialogue. *Knowledge Eng. Review*, 21(2):163–188. (p.)

Pu, F., Luo, J., Zhang, Y., and Luo, G. (2014). Argument Ranking with Categoriser Function. *CoRR*, abs/1406.3877. (p.)

Rago, A., Toni, F., Aurisicchio, M., and Baroni, P. (2016). Discontinuity-Free Decision Support with Quantitative Argumentation Debates. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016.*, pages 63–73. (p.)

Rescher, N. and Manor, R. (1970). *On inference from inconsistent premisses*, volume 1. (p.)

Rocher, S. (2013). Interrogation tolérante aux incohérences. Technical report, Université de Montpellier. (p.)

Rocher, S. (2016). *Querying Existential Rule Knowledge Bases: Decidability and Complexity*. PhD thesis, Université Montpellier II - Sciences et Techniques du Languedoc, Montpellier. (p.)

Salvat, E. and Mugnier, M.-L. (1996). Sound and Complete Forward and backward Chainingd of Graph Rules. In *Conceptual Structures: Knowledge Representation as Interlingua, 4th International Conference on Conceptual Structures, ICCS '96, Sydney, Australia, August 19-22, 1996, Proceedings*, pages 248–262. (p.)

Sowa, J. F. (1976). Conceptual Graphs for a Data Base Interface. *IBM Journal of Research and Development*, 20(4):336–357. (p.)

Staworko, S., Chomicki, J., and Marcinkowski, J. (2012). Prioritized repairing and consistent query answering in relational databases. *Ann. Math. Artif. Intell.*, 64(2-3):209–246. (p.)

Tamani, N., Mosse, P., Croitoru, M., Buche, P., and Guillard, V. (2014a). A Food Packaging Use Case for Argumentation. In Closs, S., Studer, R., Garoufallou, E., and Sicilia, M.-A., editors, *Metadata and Semantics Research: 8th Research Conference, MTSR 2014, Karlsruhe, Germany, November 27-29, 2014. Proceedings*, pages 344–358. Springer International Publishing, Cham. (p.)

Tamani, N., Mosse, P., Croitoru, M., Buche, P., Guillard, V., Guillaume, C., and Gontard, N. (2014b). Eco-Efficient Packaging Material Selection for Fresh Produce: Industrial Session. In Hernandez, N., Jäschke, R., and Croitoru, M., editors, *Graph-Based Representation and Reasoning: 21st International Conference on Conceptual Structures, ICCS 2014, Iaşi, Romania, July 27-30, 2014, Proceedings*, pages 305–310. Springer International Publishing, Cham. (p.)

Tarski, A. (1936). On Some Fundamental Concepts of Metamathematics. In *Logic, Semantics, Methamathematics*. Oxford University Press. (p.)

Thimm, M. (2017). The Tweety Library Collection for Logical Aspects of Artificial Intelligence and Knowledge Representation. *KI*, 31(1):93–97. (p.)

Toni, F. (2012). Reasoning on the Web with Assumption-Based Argumentation. In *Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School 2012, Vienna, Austria, September 3-8, 2012. Proceedings*, pages 370–386. (p.)

Toni, F. (2013). A generalised framework for dispute derivations in assumption-based argumentation. *Artif. Intell.*, 195:1–43. (p.)

Toni, F. (2014). A tutorial on assumption-based argumentation. *Argument & Computation*, 5(1):89–117. (p.)

Verheij, B. (1999). *Two Approaches to Dialectical Argumentation: Admissible Sets and Argumentation Stages.* (p.)

Wolfgang, D. (2017). Technical Note: On the Complexity of the Uniqueness Problem in Abstract Argumentation. (p.)

Yun, B., Bisquert, P., Buche, P., and Croitoru, M. (2016). Arguing About End-of-Life of Packagings: Preferences to the Rescue. In *Metadata and Semantics Research - 10th International Conference, MTSR 2016, Göttingen, Germany, November 22-25, 2016, Proceedings*, pages 119–131. (p.)

Yun, B., Bisquert, P., Buche, P., Croitoru, M., Guillard, V., and Thomopoulos, R. (2018a). Choice of environment-friendly food packagings through argumentation systems and preferences. *Ecological Informatics*, 48:24–36. (p.)

Yun, B. and Croitoru, M. (2016). An Argumentation Workflow for Reasoning in Ontology Based Data Access. In *Computational Models of Argument - Proceedings of COMMA 2016, Potsdam, Germany, 12-16 September, 2016.*, pages 61–68. (p.)

Yun, B., Croitoru, M., and Bisquert, P. (2017a). Are Ranking Semantics Sensitive to the Notion of Core? In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, pages 943–951. (p.)

Yun, B., Croitoru, M., Bisquert, P., and Vesic, S. (2018b). Graph Theoretical Properties of Logic Based Argumentation Frameworks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 2148–2149. (p.)

Yun, B., Croitoru, M., and Vesic, S. (2019). NAKED: N-Ary graphs from Knowledge bases Expressed in Datalog+/-. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2019, Montreal, Canada, May 13-17, 2019.* (p.)

Yun, B., Croitoru, M., Vesic, S., and Bisquert, P. (2018c). DAGGER: Datalog+/- Argumentation Graph GEneRator. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 1841–1843. (p.)

Yun, B., Croitoru, M., Vesic, S., and Bisquert, P. (2018d). Graph Theoretical Properties of Logic Based Argumentation Frameworks: Proofs and General Results. In *Graph Structures for Knowledge Representation and Reasoning - 5th International Workshop, GKR 2017, Melbourne, VIC, Australia, August 21, 2017, Revised Selected Papers*, pages 118–138. (p.)

Yun, B., Thomopoulos, R., Bisquert, P., and Croitoru, M. (2018e). Defining Argumentation Attacks in Practice: An Experiment in Food Packaging Consumer Expectations. In *Graph-Based Representation and Reasoning - 23rd International Conference on Conceptual Structures, ICCS 2018, Edinburgh, UK, June 20-22, 2018, Proceedings*, pages 73–87. (p.)

Yun, B., Vesic, S., and Croitoru, M. (2018f). Toward a More Efficient Generation of Structured Argumentation Graphs. In *Proceedings of the 7th International Conference on Computational Models of Argument, COMMA 2018, 11th - 14th September, 2018, Warsaw, Poland.* (p.)

Yun, B., Vesic, S., Croitoru, M., and Bisquert, P. (2018g). Inconsistency Measures for Repair Semantics in OBDA. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 1977–1983. (p.)

Yun, B., Vesic, S., Croitoru, M., and Bisquert, P. (2018h). Viewpoints using ranking-based argumentation semantics. In *Proceedings of the 7th International Conference on Computational Models of Argument, COMMA 2018, 11th - 14th September, 2018, Warsaw, Poland.* (p.)

Yun, B., Vesic, S., Croitoru, M., Bisquert, P., and Thomopoulos, R. (2017b). A Structural Benchmark for Logical Argumentation Frameworks. In *Advances in Intelligent Data Analysis XVI - 16th International Symposium, IDA 2017, London, UK, October 26-28, 2017, Proceedings*, pages 334–346. (p.)

Zhang, H., Zhang, Y., and You, J.-H. (2015). Existential Rule Languages with Finite Chase: Complexity and Expressiveness. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 1678–1685. (p.)

Abstract

In this thesis, we investigate reasoning techniques with argumentation graphs generated from inconsistent knowledge bases expressed in the existential rules language. The existential rules language is a subset of first-order logic in which a knowledge base is composed of two layers: a fact layer and an ontology layer. The fact layer consists of factual knowledge (usually stored in relational databases) whereas the ontology layer consists of reasoning rules of deduction and negative constraints. Since the classical query answering approaches fail in the presence of inconsistencies, we chose to work with an conflict-tolerant reasoning approach that is based on building graphs with structured arguments and attacks from the deductions of the underlying logical knowledge base.

The three main results are the following. First, we study how argumentation graphs are obtained from knowledge bases expressed in existential rules, the structural properties of such graphs and show several insights as to how their generation can be improved. Second, we propose a framework that generates an argumentation graph with a special feature called sets of attacking arguments instead of the regular binary attack relation and show how it improves upon the current state-of-the-art using an empirical analysis. Third, we interest ourselves to ranking-based approaches in both the context of query answering and argumentation reasoning. In the former, we introduce a framework that is based on ranking maximal consistent subsets of facts (repairs) in order to have a more productive query answering. In the latter, we set up the foundation theory for semantics that rank arguments in argumentation graphs with sets of attacking arguments.

Résumé

Dans cette thèse, nous étudions les techniques de raisonnement utilisant des graphes d'argumentation générés à partir de bases de connaissances inconsistantes exprimées dans le langage des *règles existentielles*. Les trois principaux résultats sont les suivants. Tout d'abord, nous étudions les propriétés structurelles des graphes obtenus à partir de bases de connaissances exprimées avec des règles existentielles et nous donnons plusieurs indications sur la manière dont leur génération peut être améliorée. Deuxièmement, nous proposons une technique pour générer un graphe d'argumentation où plusieurs arguments peuvent attaquer collectivement, remplaçant ainsi la relation d'attaque binaire classique et montrons expérimentalement les avantages de cette technique. Troisièmement, nous nous intéressons aux approches fondées sur les classements pour le raisonnement en logique et en argumentation.