HAL
open science

Graphes et décompositions

Tom Bouvier

To cite this version:

Tom Bouvier. Graphes et décompositions. Informatique. Université de Bordeaux, 2014. Français. NNT : 2014BORD0341 . tel-01110277

HAL Id: tel-01110277

https://theses.hal.science/tel-01110277v1

Submitted on 27 Jan 2015

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.

N° d'ordre : 2014 BORD 0341

THÈSE

présentée à

L'UNIVERSITÉ DE BORDEAUX

École Doctorale de Mathématiques et Informatique

par

Tom BOUVIER

pour obtenir le grade de

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Graphes et décompositions

Soutenue le 15 décembre 2014 au Laboratoire Bordelais de Recherche en Informatique (LaBRI)

Après avis des rapporteurs :

Christophe Paul	Directeur de recherche
Ioan Todinca	Professeur

Devant la commission d'examen composée de :

Examinateurs

Bruno Courcelle	Professeur
Arnaud Labourel	Maître de conférences

Rapporteurs

Christophe Paul	Directeur de recherche
Ioan Todinca	Professeur

Directeurs de thèse

Cyril Gavoille	Professeur
Frédéric Mazoit	Maître de conférences

N° d'ordre : 2014 BORD 0341

THÈSE

présentée à

L'UNIVERSITÉ DE BORDEAUX

École Doctorale de Mathématiques et Informatique

par

Tom BOUVIER

pour obtenir le grade de

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Graphes et décompositions

Soutenue le 15 décembre 2014 au Laboratoire Bordelais de Recherche en Informatique (LaBRI)

Après avis des rapporteurs :

Christophe Paul	Directeur de recherche
Ioan Todinca	Professeur

Devant la commission d'examen composée de :

Examinateurs

Bruno Courcelle	Professeur
Arnaud Labourel	Maître de conférences

Rapporteurs

Christophe Paul	Directeur de recherche
Ioan Todinca	Professeur

Directeurs de thèse

Cyril Gavoille	Professeur
Frédéric Mazoit	Maître de conférences

Je tiens en premier lieu à remercier les membres de mon jury : Bruno Courcelle, Arnaud Labourel, Ioan Todinca et Christophe Paul, d'avoir accepté d'assister à ma soutenance et de tout l'interet que cela engendre concernant mes travaux. Je remercie tout particulièrement les deux rapporteurs qui ont assuré la lecture de ma thèse.

Je remercie aussi Bruno Courcelle, de m'avoir fait découvrir l'existance des décompositions de graphes, lors de ses cours de Master, ainsi que de l'interet qu'il a porté à mes travaux de recherche et des discussions que nous avons eues à ce sujet.

Je remercie Frédéric Mazoit qui m'a permis de me familiariser un peu plus avec l'univers étendu des décompositions arborescentes lors mon stage de fin de Master, puis de me perfectionner grâce à la thèse qu'il a proposée et durant laquel il m'a accompagné. J'ajoute à ces remerciement Cyril Gavoille, qui a soutenu l'initiative et a participé à l'encadrement de cette thèse, avec tout ce que cela représente.

Je remercie toutes celles et ceux qui m'ont accompagné durant toutes les années passées au sein de l'université. Un grand merci à Vincent, avec qui j'ai partagé l'ensemble de ces huit années (c'est pas évident de dormir en cours à côté d'un batteur fou). Merci à tous mes compagnons de labeur pour les distractions qu'ils m'ont apportées, aussi bien au bureau qu'à l'exterieur, je pense à Gabriel, Jérôme, Noémie-Fleur, Omar, Pierre, Simon et tous les autres habitants du bureau 123. Je remercie également L'AFoDIB ainsi que le LaBRUIT pour l'organisation des divers événements qui font vivre le laboratoire. J'ai également une pensée pour les habitués de la Soucoupe et les innombrables parties de Tarot qui ont accopagnées la pause déjeuner.

Je souhaite aussi remercier l'ensemble du personnel de l'UGC Ciné Cité de Bordeaux, de m'avoir acceuilli, toujours avec le sourir, et de m'avoir fait découvrir pas loin de 400 films durant ces trois années de thèse, chacun apportant son lot d'émotions.

Je pense aussi très fort à toutes les personnes avec lesquelles j'ai passé du temps en dehors du laboratoire et qui occupent une place importante dans mon cœur.

Mes derniers remerciements, je les adresse à mes parents et à mon frère, qui m'accompagnent et me soutiennent depuis toujours, dans toutes les étapes de ma vie, des moments qui parraissent les plus insignifiants à ceux qui sont les plus importants.

Graphes et décompositions

Dans cette thèse, nous étudions diverses largeurs de graphes autour de la largeur arborescente ainsi que de la largeur de clique. Nous commençons avec une étude comparative entre la largeur arborescente d'un graphe et la largeur de clique du graphe d'incidence associé, de laquelle nous extrayons des résultats algorithmiques encourageants. Puis nous présentons quelques propriétés structurelles liées à la largeur arborescente spéciale, largeur relativement récente qui est à mi-chemin entre les deux largeurs précédentes. Enfin nous nous intéressons à une notion plus générale connue sous le nom de fonction de partition sous-modulaire qui englobe, entre autres, les largeurs arborescentes « classique » et spéciale, la largeur de chemin ainsi que la largeur linéaire et les largeurs de branches de coupe et de découpe. Nous présentons alors un algorithme linéaire à paramètre fixé pour le calcul de ces différentes largeurs, lequel généralise un certain nombre de résultats propres à chacunes de ces largeurs.

Mots clés : graphe, décomposition de graphe, grammaire de graphe, largeur de graphe, logique monadique du second ordre, complexité paramétrée

Laboratoire Bordelais de Recherche en Informatique (LaBRI)
Université de Bordeaux
351, cours de la Libération, 33405 Talence cedex, France

Graphs and decompositions

In this thesis, we study some width parameters on graphs, beyond tree-width and clique-width. Our first investigation is a comparative study between the tree-width of a graph and the clique-width of the associated incidence graph, from which we extract some strong algorithmic results. Then we present a few structural properties over a recently defined width called special tree-width and which takes its definition through both tree-width and clique-width. Finally, we end our journey with a more general notion named sub-modular partition fonction and which encompass both "classic" and special tree-widths, path-width, branch-width, linear-width, cut-width and carving-width among others. So, we introduce a fixed parameter tractable algorithm computing those widths parameters and thus we generalize a number of results specific to each of them.

Keywords: graph, graph decomposition, graph grammar, width parameter, monadic second-order logic, fixed parameter tractability

Laboratoire Bordelais de Recherche en Informatique (LaBRI)
Université de Bordeaux
351, cours de la Libération, 33405 Talence cedex, France

TABLE DES MATIÈRES

1. Introduction **1**
1.1. Organisation de la thèse 2

2. Premiers contacts **5**
2.1. Définitions générales 5
2.1.1. Graphes courants 7
2.2. Largeurs de graphes 7
2.2.1. Quelques résultats concernant ces largeurs 12
2.3. Problèmes associés aux largeurs de graphes 14

3. Graphes d'incidence et graphes scindés **17**
3.1. Définitions 18
3.1.1. Arbre de super-groupes 19
3.2. Les graphes scindés 21
3.2.1. La scission d'un multi-hyper-graphe 23
3.2.2. La scission d'un graphe 27
3.2.3. La scission de K_n 31
3.3. Les graphes d'incidence 35

4. La largeur arborescente spéciale **39**
4.1. Définitions 40
4.1.1. Comparaison avec les largeurs existantes 43
4.2. Quelques résultats de complexité 44
4.2.1. Graphes de largeurs égales 44
4.2.2. Graphes récalcitrants 47

4.2.3. Un résultat polynomial intéressant 50
4.2.4. Graphes de « petite » largeur arborescente 51
4.3. Largeur arborescente spéciale et mineurs de graphes 53
4.4. Récapitulatif 57

5. Méta-algorithme pour largeurs d'arborescences 59
5.1. Définitions 61
5.1.1. Largeurs de graphes 61
5.1.2. Fonction de partition et arbre de partitions 62
5.1.3. Largeurs de graphes et arbres de partitions 64
5.2. Problème(s) de reconnaissance 67
5.2.1. Le théorème et ses corollaires 67
5.2.2. Un algorithme naïf 73
5.3. Outils de compression 78
5.3.1. Chemins étiquetés 78
5.3.2. Caractéristique 82
5.3.3. Une relation d'ordre sur les arbres de partitions 85
5.4. Algorithme utilisant la compression 86
5.4.1. Les nœuds de départ 86
5.4.2. Les nœuds d'oubli 86
5.4.3. Les nœuds d'introduction 93
5.4.4. Les nœuds de jonction 106
5.4.5. Retour sur les largeurs de graphes 119

6. Conclusion 123

A. Divisions de sommets 127

CHAPITRE

1

INTRODUCTION

Un graphe est un objet mathématique peu connu du grand publique bien que son utilisation soit omniprésente dans la vie quotidienne. Quelques exemples simples de graphes sont les réseaux routiers, les réseaux informatiques ou encore les circuits électriques, et la liste ne s'arrête pas là.

D'une manière générale, un graphe sert à modéliser puis à résoudre un problème donné. Bien qu'un graphe soit un objet très simple, il existe une multitude de problèmes qui sont en général NP-complets si nous essayons de les résoudre sur un graphe quelconque. Parmis ces problèmes nous retrouvons divers problèmes de colorations de graphes, la recherche d'un chemin ou d'un cycle hamiltoninen, la détermination d'un plus grand sous-graphe complet... Cependant, en étudiant des cas particuliers de ces problèmes ou en se restreignant à des ensembles de graphes respectant certaines proriétés, il est possible de résoudre ces problèmes de façon efficace.

Une de ces techniques de simplification consiste à décomposer un graphe donné en graphes plus petits et donc plus facilement analysables ; le but étant souvent de majorer le nombre de sommets de ces petits graphes. Ces décompositions permettent, entre autre, une étude structurelle des graphes, notamment sur la base de configurations interdites, lesquelles jouent un rôle essentiel dans la preuve du célèbre théorème des mineurs de graphes de Robertson et Seymour [44]. La caractérisation des graphes à l'aide de ces configurations interdites a également permi une caractérisation des graphes parfaits [12].

Un autre aspect important des décompositions de graphes est qu'elles donnent souvent naissance à des paramètres relatifs aux graphes qu'elles décomposent. Un tel paramètre peut être vu comme une « mesure de complexité » d'un graphe. Par exemple, si nous considérons la décomposition arborescente et la largeur arborescente associée, le théo-

rème de Courcelle [15] nous indique qu'un certain nombre de problèmes NP-complets (et que l'on peut exprimer à l'aide de la logique monadique du second ordre) deviennent polynomiaux si nous considérons des graphes dont la largeur arborescente n'est pas trop élevée. L'utilisation de ces paramètres a entrainé l'apparition de nouvelles classes de complexités regroupées dans l'appelation complexité paramétrée (voir [26]) et plus en particulier, la complexité à paramètre fixé.

Cependant, tous les graphes ne peuvent pas être caractérisés par un paramètre de faible valeur et, de manière générale, décomposer un graphe est souvent une tâche difficile, comme c'est par exemple le cas pour les décompositions arborescentes [2]. Cependant cette paramétrisation donne accès à des algorithmes efficaces sur un certain nombre de classes de graphes, c'est pourquoi il est important de connaître les classes concernées afin de tirer parti de ces algorithmes. Il est aussi intéressant de comparer les différentes décompositions liées à ces paramètres pour savoir lesquelles sont les plus adaptées, en considérant une classe de graphes donnée.

Le théorème de Courcelle concernant la largeur arborescente ainsi que son extension à la largeur de clique [22] sont basés sur l'utilisation de grammaires de graphes, décrivant des termes représentant ces graphes. Le traitement d'un tel terme se fait à l'aide d'un automate dont la taille dépend de la largeur du graphe associé. Ici encore, le fait d'avoir des graphes de faible largeur est important car sans cela les automates seraient initilisables, car trop volumineux. Afin de ne pas se cantoner à l'étude de trop petites largeurs, Courcelle et Durand [20] ont introduit la notion d'automates programmés qui, au lieu de construire entièrement la table des transitions d'un automate donné, se contente calculer les transitions nécessaires au fur et à mesure. Il reste tout de même interessant d'avoir à faire à de petites largeurs, même pour ces automates programmés.

1.1. Organisation de la thèse

Dans le Chapitre 2, nous commençons par introduire quelques notions élémentaires sur les graphes ainsi que certaines classes de graphes simples dont nous ferons usage tout au long de cette thèse. Puis nous présentons les deux principaux invariants de graphes autour desquels nous articulons notre étude, et qui sont la largeur arborescente et la largeur de clique (correspondant aux termes anglo-saxons *tree-width* et *clique-width*). Nous poursivons ensuite par une présentation de certains résultats de complexité liés aux calculs de ces deux largeurs.

Le Chapitre 3 est consacré à une étude comportementale de la largeur de clique des graphes scindés (ou *split graphs*), en fonction de la largeur arborescente des hyper-graphes, pour lesquels nous établissons une relation bijective avec la classe des graphes scindés. Nous montrons alors que la relation entre la largeur de clique des graphes scindés et la largeur arborescente des hyper-graphes se rapproche fortement de la relation entre la largeur de clique et la largeur arborescente d'un même graphe, cette dernière nous étant donnée par Corneil et Rotics [14]. Nous restreignons ensuite les graphes scindés que nous considérons pour nous concentrer sur ceux qui sont en bijection avec les graphes et nous montrons alors que la largeur de clique desdits graphes scindés est, à

quelques optimisations près, égale à la largeur arborescente des graphes associés. Nous étendons ensuite une partie de ces résultats aux graphes d'incidence (avec une extension possible vers une plus grande portion de graphes bipartis, que nous présenterons dans le chapitre de conclusion).

L'objet d'étude du Chapitre 4 est la largeur arborescente spéciale (ou *special tree-width*), introduite il y a peu par Courcelle [17, 18], et qui est une largeur de graphes issue de restrictions portées à la fois sur la largeur arborescente et sur la largeur de clique. Après une présentation de cette nouvelle largeur ainsi que de certaines relations qu'il existe entre la largeur arborescente spéciale et les deux largeurs qui sont à l'origine de sa création, nous proposons des résultats de complexité liés au calcul de la largeur arborescente spéciale en fonction d'un certain nombre de classes de graphes bien connues, et que nous mettons en parallèle avec les résultats déjà existants pour les largeurs arborescente, de chemin (ou *path-width*) et de clique. Nous établissons également une caractérisation explicite par mineurs interdits des graphes de largeur arborescente spéciale au plus 2, et nous montrons également que les classes de graphes de largeur arborescente spéciale au plus $k \geq 3$ ne sont pas stables par minoration.

Le Chapitre 5 présente un méta-algorithme de complexité à paramètre fixé pour le calcul de diverses largeurs de graphes, dont font partie les largeurs arborescentes (spéciale ou non) et de chemin, ainsi que la largeur linéaire (*linear-width*), et les largeurs de branches (*branch-width*), de coupe (*cut-widht*) et de découpe (*carving-width*). Le méta-algorithme que nous présentons est une adaptation des travaux de Bodlaender et Kloks [5] concernant les largeurs arborescente et de chemin, afin que nous puissions nous en servir dans le cadre des fonctions de partition sous-modulaires et des arbres de partitions associés, définis par Amini *et al.* [1]. Ce méta-algorithme généralise ainsi un certain nombre de résultats concernant les largeurs de graphes précitées [5, 7, 9, 10, 49]. La preuve du bon fonctionnement du méta-algorithme est compliquée et elle constitue l'objet essentiel du chapitre.

Finalement, le Chapitre 6 offre un récapitulatif des résultats que nous avons obtenus ainsi que quelques unes des extensions possibles de ces résultats.

CHAPITRE

2

PREMIERS CONTACTS

Sommaire

2.1. Définitions générales **5**
2.1.1. Graphes courants 7
2.2. Largeurs de graphes **7**
2.2.1. Quelques résultats concernant ces largeurs 12
2.3. Problèmes associés aux largeurs de graphes **14**

2.1. Définitions générales

Définition 2.1 (Graphe)**.** Un *graphe* [1] est un couple, le plus souvent noté $G = (V, E)$, avec V un ensemble fini de *sommets* et E un ensemble de paires de sommets, appelé ensemble d'*arêtes*.

Nous disons que deux sommets u et v sont *voisins* ou *adjacents* dans G, si E contient l'arête $\{u, v\}$, que nous écrivons également uv. Les sommets u et v sont alors appelés les sommets *incidents* à uv. L'ensemble des voisins, où *voisinage*, d'un sommet v est noté $N(v)$, et le *degré* de v, généralement noté $d(v)$, désigne la taille de $N(v)$.

Définition 2.2 (Graphe orienté)**.** Un *graphe orienté* est un couple $G = (V, A)$, où V désigne toujours un ensemble fini de *sommets* et où A est un ensemble de couples de sommets, appelé ensemble d'*arcs*.

1. Sauf mention contraire, tous les graphes que nous considérons sont finis, simples, sans boucles et non-orientés.

FIGURE 2.1. : Construction du graphe K_5 (à droite) à partir du graphe de Petersen (à gauche) par minoration (contraction des arêtes en gras).

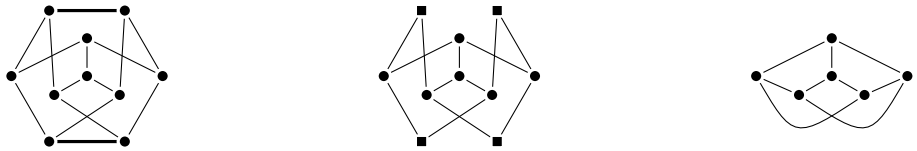


FIGURE 2.2. : Construction du graphe $K_{3,3}$ (à droite) à partir du graphe de Petersen (à gauche) par minoration topologique (suppression des arêtes en gras, puis lissage des sommets carrés).

La différence entre un graphe et un graphe orienté est que l'ordre des éléments d'une paire (arête d'un graphe) n'est pas significatif, contrairement à l'ordre des éléments d'un couple (arc d'un graphe orienté). Ainsi, dans un graphe, les arêtes $\{u,v\}$ et $\{v,u\}$ désignent le même élément, alors que, dans un graphe orienté, les arcs (u,v) et (v,u) désignent deux éléments différents.

Étant donné un arc (u,v), nous disons que u est un *voisin entrant* de v et que v est un *voisin sortant* de u. Le *voisinage entrant* d'un sommet v est noté $N^-(v)$, et le *degré entrant* de v, généralement noté $d^-(v)$, désigne la taille de $N^-(v)$. De façon analogue, le *voisinage sortant* d'un sommet v est noté $N^+(v)$, et le *degré sortant* de v, généralement noté $d^+(v)$, désigne la taille de $N^+(v)$.

Définition 2.3 (Sous-graphe, sous-graphe induit)**.** Soit $G = (V,E)$ un graphe. Nous appelons *sous-graphe* de G un graphe $G' = (V',E')$ tel que $V' \subseteq V$ et $E' \subseteq E$.

Si E' contient toutes les arêtes de E incidentes aux sommets de V', alors G' est le sous-graphe de G *induit* par les sommets de V' ; et nous notons $G' = G[V']$.

La *contraction* d'une arête uv consiste à remplacer dans G les sommets u et v par un nouveau sommet w auquel nous connectons les anciens voisins de u et de v. L'opération de contraction d'une arête incidente à un sommet v de degré 2 est aussi appelée *lissage* de v.

Définition 2.4 (Mineur, mineur topologique)**.** Soient G et H deux graphes. On dit que H est un *mineur* de G s'il peut être obtenu à partir d'un sous-graphe de G par une succession de contraction d'arêtes (voir FIGURE 2.1).

Si l'on restreint la contraction aux arêtes incidentes à des sommets de degré au plus 2 le graphe H ainsi obtenu est un *mineur topologique* de G (voir FIGURE 2.2).

2.1.1. Graphes courants

Soit $G = (V, E)$ un graphe. Si l'ensemble E contient toutes les arêtes possibles (c'est-à-dire toutes les sous-paires possible de V), nous disons que G est un *graphe complet*. À l'inverse si E ne contient aucune arête, alors G est un *graphe vide*. Soit G' un sous-graphe induit de G. Si G' est un graphe complet, alors nous disons que G' est une *clique* de G et, par extension, si G est un graphe complet, alors G est une clique. De même, si G' est un graphe vide, alors nous disons que G' est un *stable* de G et, si G est un graphe vide, alors G est un stable.

Soit $V = \{v_1, \ldots, v_n\}$ un ensemble de sommets. Si nous définissons l'ensemble $E = \{v_i v_{i+1} : 1 \leq i < n\}$, alors le graphe (V, E) est une *chaîne* et le graphe $(V, E \cup \{v_1 v_n\})$ est un *cycle*. Les sommets v_1 et v_n sont alors appelés les *extrémités* de la chaîne. Si nous définissons $A = \{(v_i, v_{i+1}) : 1 \leq i < n\}$, alors le graphe orienté (V, A) est un *chemin* et le graphe orienté $(V, A \cup \{(v_n, v_1)\})$ est un *circuit*. La *taille* d'une chaîne, d'un chemin, d'un cycle ou d'un circuit représente le nombre de sommets du graphe considéré. Dans le cas d'une chaîne ou d'un chemin P, nous définissons la *longueur* de P comme étant le nombre d'arêtes ou d'arcs que P contient (ce nombre est égal au nombre de sommets moins 1).

Un graphe est dit *connexe* si, pour toute sous-paire de sommets $\{u, v\}$ de V il existe une sous-chaîne de G dont u et v sont les extrémités.

Un *arbre* T est un graphe connexe sans cycle, dont les sommets de degré 1 sont appelés les *feuilles* de T et les autre sommets sont appelés les *sommets internes* de T. Un *arbre enraciné* T est un arbre avec un sommet r que nous identifions comme étant la *racine* de T, laquelle nous permet de définir une orientation de T telle qu'il existe un chemin reliant r à chacune des feuilles de T. Étant donné v, un sommet d'un arbre enraciné T, les voisins sortants (s'ils existent) de v sont appelés les *fils* de v et l'unique voisin entrant (s'il existe) de v est appelé le *père* de v. Tous les sommets atteignables depuis v (en suivant un chemin) sont les *descendants* de v et tous les sommets sur le chemin reliant la racine de T à v sont les *ancêtres* de v.

2.2. Largeurs de graphes

Dans cette section nous présentons deux des largeurs de graphes que nous étudions dans cette thèse.

Définition 2.5 (Expression en clique)**.** Soit G un graphe. Une *expression en clique* est un terme Γ représentant un étiquetage de G, noté $\mathrm{val}(\Gamma)$, et composé des quatre opérations suivantes :

- λ : création d'un sommet étiqueté λ ;
- $\rho_{\lambda_i \to \lambda_j}(\Gamma)$: ré-étiquetage des sommets d'étiquette λ_i avec l'étiquette λ_j dans $\mathrm{val}(\Gamma)$;
- $\eta_{\lambda_i, \lambda_j}(\Gamma)$: ajout de toutes les arêtes entre les sommets d'étiquette λ_i et ceux d'étiquette λ_j dans $\mathrm{val}(\Gamma)$, avec $\lambda_i \neq \lambda_j$;
- $\Gamma_1 \oplus \Gamma_2$: union disjointe des graphes $\mathrm{val}(\Gamma_1)$ et $\mathrm{val}(\Gamma_2)$.

La *largeur* de Γ représente le nombre d'étiquettes utilisées par Γ. La *largeur de clique* de G, notée $\mathrm{cw}(G)$, est la largeur minimale parmi toutes les expressions en clique qui représentent un étiquetage de G.

Les notions d'expression en clique et de largeur de clique peuvent également être définies dans le cadre des graphes orientés. Dans ce cas l'opération $\overrightarrow{\eta}_{\lambda_i,\lambda_j}(\Gamma)$ ajoute tous les arcs sortant des sommets étiquetés λ_i et entrant dans ceux étiquetés λ_j. Il est également nécessaire de définir l'opération $\overrightarrow{\eta}_{\lambda_j,\lambda_i}(\Gamma)$, alors que l'opération $\eta_{\lambda_j,\lambda_i}$ n'est pas utile, voir même redondante, dans le cas non-orienté.

Comme les expressions en clique sont assez volumineuses, il peut vite devenir difficile d'écrire ou de lire lesdites expressions sans faire d'erreur. C'est pour cela que nous définissons les méta-opérations suivantes :

- $op_1 \circ \cdots \circ op_n(\Gamma)$ désigne la suite d'opérations $op_1(\ldots(op_n(\Gamma))\ldots)$, avec op_i qui représente soit une opération d'ajout d'arêtes, soit une opération de renommage ;
- ρ_M désigne la suite d'opérations $\rho_{\mu_1\to\mu'_1} \circ \cdots \circ \rho_{\mu_n\to\mu'_n}(\Gamma)$, avec $M = \{(\mu_i,\mu'_i) : 1 \le i \le n\}$ et $\{\mu_i : 1 \le i \le n\} \cap \{\mu'_i : 1 \le i \le n\} = \emptyset$;
- $\eta_{\lambda,\Lambda}(\Gamma)$ désigne la suite d'opérations $\eta_{\lambda,\lambda_1} \circ \cdots \circ \eta_{\lambda,\lambda_n}(\Gamma)$, avec $\Lambda = \{\lambda_1,\ldots,\lambda_n\}$.

Par la suite, nous utilisons la notation *k-expression* pour se référer à une expression en clique de largeur k. De plus, nous notons $\ell(v)$ l'étiquette associée à un sommet v donné.

La Figure 2.3 illustre différentes étapes de la construction du graphe de Petersen à l'aide d'une expression en clique manipulant 5 étiquettes.

Définition 2.6 (Décomposition arborescente)**.** Soit $G = (V,E)$ un graphe. Une *décomposition arborescente* de G est un couple $(T,\mathcal{X})$, avec $\mathcal{X} = \{X_1,\ldots,X_p\}$ une famille de sous-ensembles de V et T un arbre dont les *nœuds* [2] sont étiquetés par les éléments de $\mathcal{X}$, tel que :

- l'union de tous les éléments de $\mathcal{X}$ est égale à V ;
- pour chaque arête $e \in E$, il existe un $X_i \in \mathcal{X}$ qui contient tous les sommets de e ;
- pour tout sommet $v \in V$, l'ensemble des éléments de $\mathcal{X}$ contenant v induit un sous-arbre dans T.

La *largeur* de $(T,\mathcal{X})$ est définie par le maximum sur la taille des éléments de $\mathcal{X}$ moins 1. La largeur minimale, parmi toutes les décompositions arborescentes de G, représente la *largeur arborescente* de G et est notée $\mathrm{tw}(G)$.

Un regard rapide sur la définition précédente nous indique qu'il existe un nombre infinit de décompositions arborescentes pour un graphe et qu'elles risquent de ne pas toutes être très élégantes à manipuler. C'est pour de remédier à ce désagrément que nous définissons la notion de « jolie décomposition arborescente » ci-dessous.

Une *décomposition arborescente enracinée* est une décomposition arborescente dans laquelle l'arbre de décomposition est enraciné.

Définition 2.7 (Jolie décomposition arborescente)**.** Une décomposition arborescente enracinée $(T,\mathcal{X})$ d'un graphe G est dite *jolie* si et seulement si elle est composée des quatre types de nœuds suivants :

2. Les sommets de l'arbre de décomposition sont communément appelés « nœuds » afin d'éviter toute confusion avec les sommets du graphe décomposé.

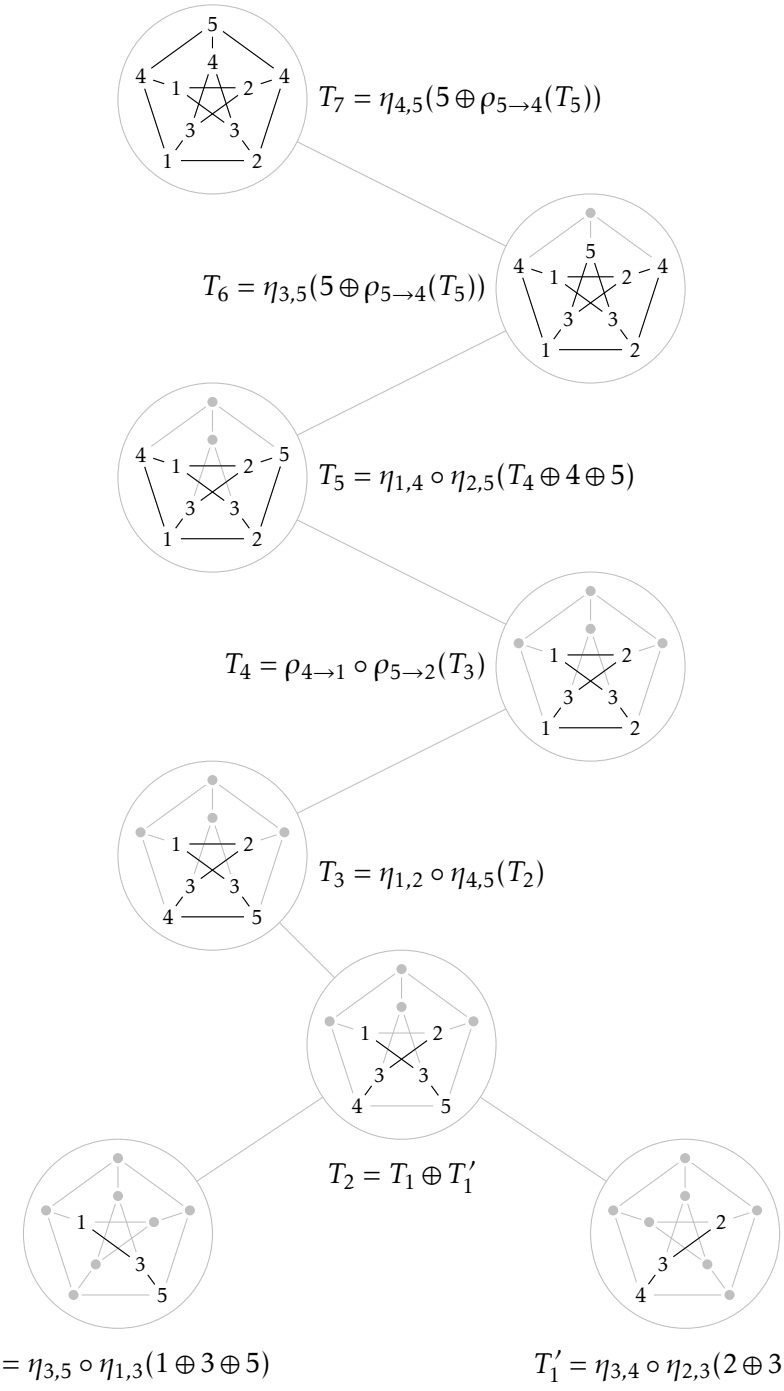


FIGURE 2.3. : Construction d'une 5-expression pour le graphe de Petersen.

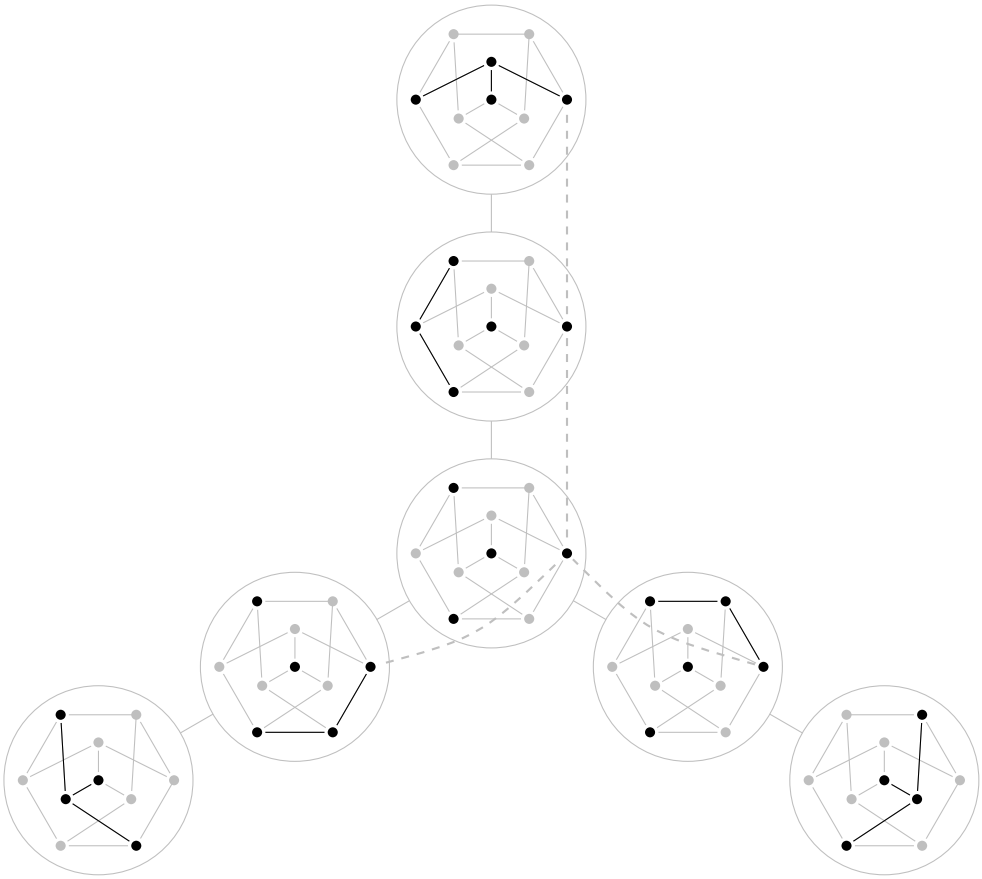


Figure 2.4. : Une décomposition arborescente du graphe de Petersen de largeur 4.

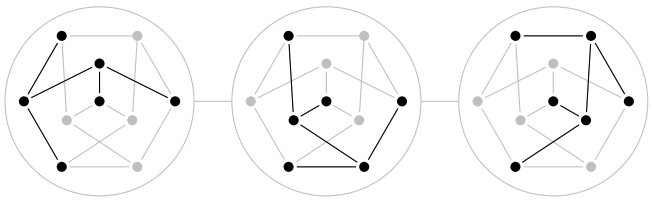


Figure 2.5. : Une décomposition en chemin du graphe de Petersen de largeur 5.

- un *nœud de départ* est une feuille de T et contient un unique sommet de G ;
- un *nœud d'oubli* α a un unique fils β tel que $X_\alpha = X_\beta \setminus \{v\}$ et $v \in X_\alpha$;
- un *nœud d'introduction* α a un unique fils β tel que $X_\alpha = X_\beta \cup \{v\}$ et $v \notin X_\alpha$;
- un *nœud de jonction* α a exactement deux fils β et γ tels que $X_\alpha = X_\beta = X_\gamma$.

Dans le cas d'un nœud d'oubli, le sommet v est appelé le *sommet oublié* et, dans le cas d'un nœud d'introduction, v est le *sommet introduit*.

Lemme 2.8. *Pour toute décomposition arborescente de G il existe une jolie décomposition arborescente de même largeur qui représente G.*

Définition 2.9 (Décomposition en chemin)**.** Une *décomposition en chemin* d'un graphe $G = (V, E)$, que nous notons $(P, \mathcal{X})$, est une décomposition arborescente de G, où P est un chemin. La *largeur* de $(P, \mathcal{X})$ est le maximum sur la taille des éléments de $\mathcal{X}$ moins 1. La *largeur de chemin* de G, notée $\mathrm{pw}(G)$, correspond à la largeur minimale parmi toutes les décompositions en chemin de G.

La Figure 2.4 illustre une décomposition arborescente du graphe de Petersen de largeur 4 et la Figure 2.5 représente une décomposition en chemin, toujours pour le graphe de Petersen, et de largeur 5.

La définition de la largeur arborescente que nous venons de donner n'est pas unique et nous serons amener à considérer d'autres définitions, que nous présentons de ce pas.

Définition 2.10 (k-arbre, k-arbre partiel)**.** Le graphe complet à k sommets est un *k-arbre* et, si G est un k-arbre et X une clique de taille k dans G, alors le graphe obtenu en ajoutant un sommet dans G puis en le rendant adjacent à tous les sommets de C est un k-arbre. Un *k-arbre partiel* est un sous-graphe d'un k-arbre.

Théorème 2.11 (Scheffer [45], van Leeuwen [50])**.** *Un graphe G est un k-arbre partiel si et seulement si la largeur arborescente de G est au plus k.*

Comme pour la largeur de clique, il est possible d'exprimer la largeur arborescente grâce à un ensemble d'opérations définissant des termes.

Définition 2.12 (Expression arborescente)**.** Soit G un graphe. Une *expression arborescente* est un terme T représentant un étiquetage de G, noté $\mathrm{val}(T)$. Soit S un ensemble d'étiquettes représentant les *sources* de $\mathrm{val}(T)$ et soit $\bullet$ l'étiquette représentant les autres sommets de $\mathrm{val}(T)$. Le terme T est alors composé des quatre opérations suivantes :

- λ : création d'une source étiquetée λ ;
- $\overline{\lambda_i \lambda_j}$: création de deux sources étiquetées λ_i et λ_j reliées par une arête ;
- $T_1 /\!\!/ T_2$: union disjointe de $\mathrm{val}(T_1)$ et $\mathrm{val}(T_2)$, puis identification des sources partageant une même étiquette ;
- $fg_{\lambda_i}(T)$: oubli de la source s d'étiquette λ_i (ré-étiquetage de s avec l'étiquette $\bullet$).

La *largeur* de T représente le nombre d'étiquettes utilisées pour identifier les sources dans T.

Si nous nous plaçons dans le cadre des graphes orientés, alors nous remplaçons l'opération $\overline{\lambda_i \lambda_j}$ par les opérations $\overrightarrow{\lambda_i \lambda_j}$ et $\overrightarrow{\lambda_j \lambda_i}$. Notons que les expressions arborescentes permettent de définir des multi-graphes (graphes dans lesquels deux sommets peuvent être reliés par plusieurs arêtes), ainsi que des boucles (arêtes dont les deux incidences sont un unique sommet) en ajoutant l'opération λ^ℓ.

Théorème 2.13 (Courcelle [16])**.** *Il existe une décomposition arborescente de G de largeur k si et seulement si il existe une expression arborescente de largeur $k+1$ représentant G. De plus, il existe des algorithmes linéaires permettant de passer d'une décomposition arborescente à une expression arborescente de largeur équivalente et inversement.*

La Figure 2.6 représente la construction du graphe de Petersen à l'aide d'une expression arborescente utilisant 5 sources. Il est interressant de noter la ressemblance entre cette dernière construction et la décomposition arborescente de la Figure 2.4.

2.2.1. Quelques résultats concernant ces largeurs

La largeur arborescente est au cœur de certains des plus importants résultats de la théorie des graphes. Elle joue en effet un rôle clé dans la preuve du célèbre Théorème de Robertson et Seymour [44], statuant que toute famille de graphes close par minoration peut être caractérisée par un ensemble fini de mineur interdits. Ce théorème implique entre autre que, pour tout graphe appartenant à une famille stable par minoration, il existe un algorithme de reconnaissance polynomial pour ce graphe (voir [43]).

Un autre résultat célèbre concernant la largeur arborescente est le Théorème de Courcelle [15], lequel nous dit que, étant donné un graphe G et une décomposition arborescente de G de largeur majorée par une constante, toute propriété exprimable en logique monadique du second ordre, avec une quantification sur les sommets ainsi que sur les arêtes de G, est calculable en temps linéaire sur la taille de G. Il en découle qu'un grand nombre de problèmes, qui sont NP-complets dans le cas général, deviennent polynomiaux si nous nous restreignons à l'étude des graphes de largeur arborescente majorée par une constante. Parmis ces problèmes nous retrouvons le problème de la 3-coloration d'un graphe, par exemple.

En combinant un résultat de Bodlaender et Kloks [5] avec un autre résultat de Bodlaender [4] (rappelés dans les Théorèmes 2.17 et 2.18 ci-dessous), il est possible d'améliorer le résultat de Courcelle en supprimant la décomposition arborescente de l'entrée de l'algorithme. En effet, quand la largeur arborescente de G est majorée, il existe un algorithme linéaire donnant une décomposition arborescente de G respectant cette majoration.

La largeur de clique a elle aussi des conséquences notables quand nous nous intéressons aux graphes pour lesquelles cette largeur peut être majorée. Comme l'ont montré Courcelle, Makowsky et Rotics [22], étant donné un graphe G et une expression en clique de G de largeur majorée, toute propriété exprimable en logique monadique du second ordre, avec une quantification sur les sommets de G (mais pas sur ses arêtes), est calculable en temps linéaire sur la taille de G. Les problèmes que nous pouvons traiter grâce à cet algorithme sont alors moins nombreux que si nous utilisons la largeur

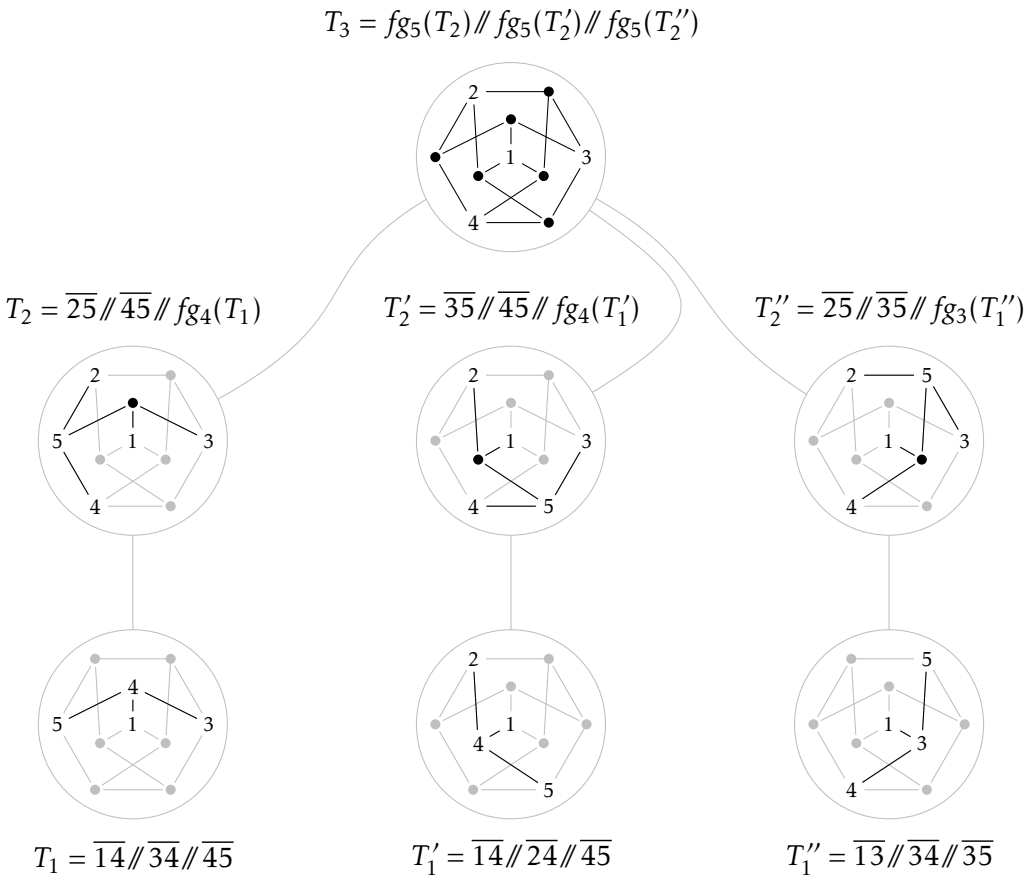


FIGURE 2.6. : Construction d'une expression arborescente pour le graphe de Petersen qui manipule 5 sources.

arborescente. Cependant, en considérant la classe des cographes ainsi que le théorème suivant, nous pouvons remarquer qu'il existe strictement plus de classes de graphes dont la largeur de clique est majorée par une constante que de classes de graphes dont la largeur arborescente est majorée par une constante.

Théorème 2.14 (Corneil et Rotics [14]). *Pour tout graphe G, nous avons* $\mathrm{cw}(G) \leq 3 \cdot 2^{\mathrm{tw}(G)-1}$.

Ce résultat nous indique que tous les graphes dont la largeur arborescente est majorée par une constante ont une largeur de clique qui est elle aussi majorée par une constante. L'écart entre ces deux constantes est cependant exponentiel et nous ne pouvons pas espérer faire beaucoup mieux comme en témoigne le résultat suivant.

Théorème 2.15 (Corneil et Rotics [14]). *Pour tout entier k, il existe un graphe G tel que* $\mathrm{tw}(G) = k$ *et* $2^{\lfloor k/2 \rfloor - 1} \leq \mathrm{cw}(G)$.

Pour avoir un bon apperçu des autres résultats concernant ces deux largeurs de graphes, ainsi qu'un certain nombre d'autres largeurs, nous vous proposons de lire [33].

2.3. Problèmes associés aux largeurs de graphes

Quand nous avons à faire à des largeurs de graphes, il existe trois grandes familles de problèmes. Le *problème d'optimisation* qui consiste, étant donné un graphe G, à déterminer quelle est la largeur optimale de G. D'une première restriction de ce problème découle le *problème de décision* qui, étant donnés un graphe G et un entier k, permet de décider si la largeur de G est au plus k. Si nous fixon à l'avance l'entier k du problème précédent, nous obtenons le *problème de k-reconnaissance* qui, étant donné un graphe G, détermine si G appartient à la classe des graphes de largeur au plus k. Nous définissons également le *problème de reconnaissance* qui englobe les problèmes de k-reconnaissance, pour tout k fixé.

Étant donné que nous travaillons sur des problèmes algorithmiques, nous allons être amenés à parler de complexité. Certains des algorithmes que nous évoquons ont une complexité dite *paramétrée*, c'est à dire que la mesure de complexité de l'algorithme choisi est fonction de plusieurs paramètres, en plus de la taille totale des données en entrée. Et nous traiterons plus précisément d'algorithmes de complexité *à paramètre fixé*, dont la mesure de complexité s'écrit $\mathcal{O}(f(k) \cdot P(n))$, où n représente la taille des données en entrée, k est le paramètre, P est une fonction polynomiale et f est une fonction calculable (et généralement non-polynomiale).

Un algorithme est dit *constructif*, si dans le cas où le problème de décision, de k-reconnaissance ou de reconnaissance considéré retourne une réponse positive, alors l'algorithme est également capable de construire une décomposition ou une expression dont la largeur respecte le majorant défini dans l'énoncé du problème. Dans le cas du problème d'optimisation, l'algorithme est dit constructif, s'il retourne une décomposition ou une expression de largeur optimale.

En ce qui concerne la largeur arborescente, nous avons les résultats de complexité suivants.

Théorème 2.16 (Arnborg *et al.* [2]). *Les problèmes de décision liés à la largeur arborescente ainsi qu'à la largeur de chemin sont NP-complets.*

Théorème 2.17 (Bodlaender et Kloks [5]). *Pour tout graphe, il existe des algorithmes constructifs linéaires résolvant les problèmes de reconnaissance liés à la largeur arborescente ainsi qu'à la largeur de chemin.*

Théorème 2.18 (Bodlaender [4]). *Pour tout graphe de largeur arborescente majorée, il existe un algorithme constructif linéaire résolvant le problème d'optimisation lié à la largeur arborescente.*

Théorème 2.19 (Bodlaender et Kloks [5]). *Pour tout graphe de largeur arborescente majorée, il existe un algorithme constructif polynomial résolvant le problème d'optimisation lié à la largeur de chemin.*

Les mêmes questions se posent naturellement pour la largeur de clique, mais avec moins de résultats.

Théorème 2.20 (Fellows *et al.* [28]). *Le problème de décision lié à la largeur de clique est NP-complet.*

Théorème 2.21 (Corneil *et al.* [13]). *Pour tout graphe, il existe un algorithme constructif polynomial résolvant le problème de* 3*-reconnaissance lié à la largeur de clique.*

Théorème 2.22 (Espelage *et al.* [27]). *Pour tout graphe de largeur arborescente majorée, il existe un algorithme constructif linéaire résolvant le problème d'optimisation lié à la largeur de clique.*

Les problèmes suivants sont, à ce jour, toujours ouverts.

Question 2.23. Existe-t-il un algorithme polynomial résolvant le problème de 4-reconnaissance lié à la largeur de clique ?

Question 2.24. Existe-t-il un algorithme polynomial résolvant le problème de reconnaissance lié à la largeur de clique ?

Question 2.25. Existe-t-il un algorithme de complexité à paramètre fixé résolvant le problème de reconnaissance lié à la largeur de clique ?

CHAPITRE

3

GRAPHES D'INCIDENCE ET GRAPHES SCINDÉS

Sommaire

3.1. Définitions **18**
3.1.1. Arbre de super-groupes 19
3.2. Les graphes scindés **21**
3.2.1. La scission d'un multi-hyper-graphe 23
3.2.2. La scission d'un graphe 27
3.2.3. La scission de K_n 31
3.3. Les graphes d'incidence **35**

Les deux largeurs de graphes qui vont nous accompagner durant ce chapitre sont la largeur arborescente ainsi que la largeur de clique. Comme nous l'avons vu dans les deux premiers chapitres, ces deux paramètres sont loin d'être identiques, bien que fortement reliés. L'étude que nous proposons ici va nous permettre de rapprocher un peu plus ces deux invariants de graphes, et il se pourrait même que nous arrivions à les confondre.

Pour cela nous définissons une fonction bijective, que nous appelons la scission d'un graphe, et qui retourne un unique graphe scindé associé au graphe passé en argument de la fonction. Nous montrons, entre autres, que la largeur arborescente d'un graphe donné est équivalente à la largeur de clique du graphe de scission associé.

Les graphes de scission que nous définissons resemblent à s'y méprendre aux graphes dits d'incidence. Nous pouvons, de ce fait, étendre les résultats concernant les graphes

de scissions que nous avons obtenus à ces graphes d'incidence. Par la même occasion, nous répondons à une question récement posée par Courcelle [19], et mettant en scène largeur de clique et mineurs topologiques.

3.1. Définitions

Définition 3.1 (Graphe scindé)**.** Un *graphe scindé* G est un graphe dont les sommets peuvent être partitionnés en deux ensembles, l'un induisant une clique C de G et l'autre un stable P. On écrit alors $G = (C, P, E)$, où $C \neq \emptyset$ représente la clique (et ses sommets sont appelés les *sommets centraux* de G), P représente le stable (et ses sommets sont appelés les *sommets périphériques* de G) et E désigne l'ensemble des arêtes joignant les sommets centraux aux sommets périphériques.

Remarque 3.2. Deux graphes scindés différents peuvent être isomorphes. Par exemple, le graphe complet K_n correspond à deux graphes scindés différents : dans le premier cas le graphe s'écrit $(K_n, \emptyset, \emptyset)$ alors que, dans le second cas, nous avons $(K_{n-1}, \{v\}, E)$, avec $E = \{uv : u \in V(K_{n-1})\}$.

Afin de pouvoir présenter la notion de graphe de scission, nous devons préalablement introduire ce que sont les hyper-graphes.

Définition 3.3 (Hyper-graphe et multi-hyper-graphe)**.** Un *hyper-graphe* est un couple, généralement noté $H = (V, E)$, avec V un ensemble fini de sommets et E un ensemble fini d'*hyper-arêtes*. Chaque élément de E est un sous-ensemble non vide de V.

Dans un *multi-hyper-graphe*, l'ensemble E devient un multi-ensemble, c'est-à-dire que E peut contenir plusieurs hyper-arêtes incidentes à un même sous-ensemble de sommets de V.

En particulier un graphe est un hyper-graphe dont toutes les hyper-arêtes sont de taille 2. De plus, la définition de la décomposition arborescente que nous avons donnée (Définition 2.6) est compatible avec la notion de multi-hyper-graphe. Nous reprenons également la définition de k-arbre (Définition 2.10) et nous l'étendons pour obtenir un *hyper-k-arbre* en ajoutant les multi-arêtes nécessaires.

Définition 3.4 (Graphe de scission)**.** Soit $H = (V, E)$ un multi-hyper-graphe. Le *graphe de scission* de H est un graphe scindé dont les sommets centraux sont les éléments de V, les sommets périphériques sont les éléments de E et les arêtes sont tous les éléments uv et xe, avec $u, v \in V$, $e \in E$ et $x \in e$. Nous notons ce graphe $\mathcal{S}(H)$.

Les graphes de scissions ont été utilisés en autres par Makowsky et Rotics [37] afin de montrer que la largeur de clique de graphes scindés ne peut pas être majorée par une constante.

Lemme 3.5. *Il existe une bijection entre l'ensemble des graphes scindés et l'ensemble des graphes de scission de tous les multi-hyper-graphes.*

Ce résultat découle de la définition. Nous reprenons les graphes de la Remarque 3.2 pour illustrer cette bijection. Nous avons donc deux hyper-graphes H_1 et H_2 tels que $\mathcal{S}(H_1) = (K_n, \emptyset, \emptyset)$ et $\mathcal{S}(H_2) = (K_{n-1}, \{v\}, E)$, avec $E = \{uv : u \in V(K_{n-1})\}$. D'après la définition, H_1 est un multi-hyper-graphe d'ordre n sans hyper-arête et H_2 est un multi-hyper-graphe d'ordre $n-1$ avec une hyper-arête incidente à tous les sommets de H_2.

3.1.1. Arbre de super-groupes

La notion que nous nous apprêtons à définir est une notion équivalente à la largeur de clique qui a été introduite il y a peu par Heggernes, Meister et Rotics [32]. Ils se servent de cette nouvelle caractérisation de la largeur de clique pour définir la largeur de clique d'une certaine classe de graphes, tout en faisant abstraction des difficultés qu'auraient causé l'utilisation de la définition « classique » de la largeur de clique. Nous réutilisons leur travail pour les mêmes raisons. Notons que la façon dont cette nouvelle notion a été introduite est plutôt complexe et elle tend à être remplacée par une notion plus simple de partitions imbriquées [21].

Nous commençons alors par donner les définitions de super-groupe et d'arbre de super-groupes, puis nous éclaircissons ces nouvelles notions grâce à une comparaison avec la définition « classique » de la largeur de clique.

Étant donnés deux graphes G et H, avec H un sous-graphe de G, nous définissons le graphe $G - H = (V(G), E(G) \setminus E(H))$, en d'autres termes $G - H$ est le complémentaire de H dans G.

Définition 3.6 (Super-groupe, partition et compatibilité)**.** Soient G et H deux graphes tels que H est un sous-graphe de G. Un *super-groupe* de H, noté A, est un sous-ensemble des sommets de H tel que, pour toute paire (u, v) de sommets pris dans A, nous avons $N_{G-H}(u) \subseteq N_G(v)$.

Une *partition en super-groupes* de H, notée $(A_1, \ldots, A_r)$, est une partition des sommets de H en r super-groupes, où r dénote la *largeur* de la partition.

Finalement, nous disons que deux super-groupes A_i et A_j sont *compatibles* dans H si : il existe une arête uv dans $E(G - H)$, avec $u \in A_i$ et $v \in A_j$ alors, pour tout $x \in A_i$ et pour tout $y \in A_j$, il correspond une arête xy dans $E(G)$.

Nous faisons maintenant un rapprochement entre les super-groupes et les expressions en clique. Soit Γ une expression en clique et soit Γ_α une sous-expression de Γ. Dans la notion de super-groupe que nous venons de définir, le graphe noté G correspond au graphe $\mathrm{val}(\Gamma)$, et le sous-graphe de G, noté H dans la définition, correspond au graphe $\mathrm{val}(\Gamma_\alpha)$. Un super-groupe A_i de H regroupe l'ensemble des sommets de $\mathrm{val}(\Gamma_\alpha)$ étiquettés λ_i ; la notion de partition en super-groupes est alors naturelle et la largeur d'une partition correspond au nombre d'étiquettes présentes dans $\mathrm{val}(\Gamma_\alpha)$.

La propriété $N_{G-H}(u) \subseteq N_G(v)$ nous dit que si x est un voisin de u dans $\mathrm{val}(\Gamma)$ mais qu'il ne l'est pas dans $\mathrm{val}(\Gamma_\alpha)$, alors x est un voisin de v dans $\mathrm{val}(\Gamma)$. Cette propriété va de paire avec la notion de compatibilité de deux super-groupes, laquelle revient à dire que si l'on ajoute, dans $\mathrm{val}(\Gamma_\alpha)$ une arête entre deux sommets u et v, respectivement étiquetés λ_i et λ_j, alors nous devons ajouter toutes les arêtes entre les sommets d'étiquette λ_i et

les sommets d'étiquette λ_j. La combinaison de ces deux propriétés est donc équivalente à l'opération d'ajout d'arêtes. En particulier, si u et v appartiennent à un même super-groupe mais ne sont pas voisins dans H, alors ils ne sont pas non plus voisin dans G car, dans le cas contraire, nous aurions $v \in N_G(v)$, ce qui n'est pas possible étant donné que $N_G(v)$ représente le voisinage de v au sens strict ; cela revient à dire que l'on ne peut pas ajouter d'arête entre deux sommets de même étiquette.

Nous passons maintenant à la seconde partie de la définition. Étant données deux partitions $\mathcal{A}$ et $\mathcal{B}$ d'un ensemble, on dit que $\mathcal{B}$ est un *raffinement* de $\mathcal{A}$ si, pour tout $B_i \in \mathcal{B}$, il existe $A_j \in \mathcal{A}$ tel que $B_i \subseteq A_j$.

Définition 3.7 (Arbre de super-groupes)**.** Un *arbre de super-groupes* d'un graphe G est un couple $(\mathcal{J}, \mathcal{Y})$, avec $\mathcal{Y}$ un ensemble de partitions en super-groupes de sous-graphes de G et $\mathcal{J}$ un arbre binaire enraciné et dont les nœuds sont étiquetés par ces partitions, tel que :

- les feuilles de $\mathcal{J}$ sont en bijection avec les partitions $(\{v_1\}), \ldots, (\{v_n\})$, où $v_1, \ldots, v_n$ sont les sommets de G ;
- si α est un nœud de $\mathcal{J}$ avec β et γ pour fils et que ces trois nœuds sont respectivement étiquetés avec les partitions $(A_1, \ldots, A_p)$, $(B_1, \ldots, B_q)$ et $(C_1, \ldots, C_r)$, alors :
  - $(B_1, \ldots, B_q, C_1, \ldots, C_r)$ est un raffinement de $(A_1, \ldots, A_p)$;
  - $(A_1, \ldots, A_p)$ est une partition en super-groupes de $G[B_1 \cup \cdots \cup B_q] \oplus G[C_1 \cup \cdots \cup C_r]$;
  - les super-groupes $A_1, \ldots, A_p$ sont deux à deux compatibles.

Finalement, la *largeur* de $(\mathcal{J}, \mathcal{Y})$ correspond à la largeur de la plus grande partition qui étiquette un nœud de $\mathcal{J}$.

Une fois encore nous exprimons les notions nouvellement introduites en fonction des expressions en clique avec lesquelles nous avons l'habitude de travailler. Forcer une feuille de $\mathcal{J}$ à être étiquetée avec la partition $(\{v_i\})$ correspond, dans l'expression Γ associée à G, à définir l'opération λ_i ; la bijection devient alors évidente. La condition de rafinement permet de simuler les différentes opérations de ré-étiquetages de Γ et les deux dernières conditions remplacent les opérations d'ajouts d'arêtes. En d'autres termes, un arbre de super-groupes ressemble à une expression en clique dans laquelle toutes les opérations de ré-étiquetages et d'ajouts d'arêtes sont implicitement définies par le graphe G ainsi que par le partitionnement (ou étiquetage) de sous-graphes de G bien et dûment choisis.

Maintenant que les définitions ont été posées, il nous faut montrer que la notion que nous venons de définir est, comme nous l'affirmons, équivalente à la notion de largeur de clique.

Théorème 3.8 (Heggernes *et al.* [32])**.** *La largeur minimale que peut atteindre un arbre de super-groupes définissant un graphe G est égale à la largeur de clique de G.*

Afin de mieux comprendre le parallèle entre expressions en clique et arbres de super-groupes nous montrons comment passer d'une k-expression Γ d'un graphe G à un arbre de super-groupes $(\mathcal{J}, \mathcal{Y})$ de largeur k définissant G. Nous appelons T l'arbre enraciné sous-jacent à l'expression Γ : l'opération λ_i définit une feuille de T, les opérations $\eta_{\lambda_i, \lambda_j}$

et $\rho_{\lambda_i \to \lambda_j}$ sont des sommets avec un unique fils et l'opération $\oplus$ définit un sommet avec deux fils. Notons que, par définition, les feuilles de T sont en bijection avec les sommets de G. La structure de $\mathcal{J}$ est obtenue à partir de T en lissant (ou en le supprimant dans le cas de la racine de T) tous les sommets avec un unique fils. Il existe donc une bijection entre les feuilles de $\mathcal{J}$ et celles de T ainsi qu'entre les sommets internes de $\mathcal{J}$ et les sommet de T correspondant à l'opération $\oplus$. Nous passons maintenant au remplissage des nœuds de $\mathcal{J}$. Soit α un nœud de $\mathcal{J}$ et soit Γ_α la sous-expression de Γ correspondante. Nous remplissons alors Y_α de la façon suivante :

- si $\Gamma_\alpha = \lambda_i$, alors Y_α est constitué d'une unique partie correspondant au singleton qui contient l'unique sommet de $\mathrm{val}(\Gamma_\alpha)$;
- si $\Gamma_\alpha = \Gamma_\beta \oplus \Gamma_\gamma$, alors nous mettons dans Y_α la partition $(P_1, \dots, P_r)$ des sommets de $\mathrm{val}(\Gamma_\alpha)$ et définie par l'étiquettage de $\mathrm{val}(\Gamma_\alpha)$, avec $r \leq k$.

3.2. Les graphes scindés

Comme nous l'avons vu dans l'introduction de ce chapitre, nous allons montrer que calculer la largeur arborescente des graphes est équivalent à calculer la largeur de clique de certains graphes scindés. Nous allons également tenter de caractériser la largeur de clique de l'ensemble des graphes scindés qui, comme nous l'avons vu, est en bijection avec l'ensemble des multi-hyper-graphes. Nous ferons aussi un tour du côté des graphes de scissions des graphes complets, afin d'améliorer le résultat de Makowsky et Rotics et, par la même occasion, d'illustrer l'utilité des arbres de super-groupes.

Voici un premier résultat sur la largeur de clique des graphes scindés qui va nous être utile, bien qu'il ne soit pas lié à la largeur arborescente des multi-hyper-graphes.

Théorème 3.9. *Soit G un graphe scindé, alors* $\mathrm{cw}(G) \leq 1 + \min\{c, p\}$, *où c est le nombre de sommets centraux et p le nombre de sommets périphériques.*

Démonstration. Pour commencer, nous considérons que les sommets centraux de G sont indexés de cv_1 à cv_c et que les sommets périphériques sont indexés de pv_1 à pv_p. Sans trop de surprise la preuve que nous proposons est scindée en deux parties, d'une part pour montrer que $\mathrm{cw}(G) \leq p+1$, et d'autre part pour établir $\mathrm{cw}(G) \leq c+1$.

Majorant périphérique. Pour ce majorant, nous fournissons un algorithme construisant une $(p+1)$-expression définissant G. Cet algorithme se décompose en deux temps. Dans le premier temps, nous construisons le graphe G' qui est le sous-graphe induit de G, privé du sommet périphérique pv_p ainsi que de tous ses voisins (centraux) dans G. Dans le second temps, nous insérons dans G' les sommets manquants pour obtenir G. Pour cela nous utlisons l'ensemble d'étiquettes $\{\lambda_1, \dots, \lambda_{p-1}\} \cup \{\star, \bullet\}$.

Afin de travailler simplement, nous supposons que les k premiers sommets centraux ne sont pas voisins de pv_p dans G, alors que les $c-k$ derniers le sont. Nous pouvons alors construire l'expression définissant G' grâce à la récursion suivante :

$$\begin{aligned}
&\Gamma_0(G') := \lambda_1 \oplus \cdots \oplus \lambda_{p-1} \\
&\Gamma_i(G') := \rho_{\star \to \bullet} \circ \eta_{\star,\bullet} \circ \eta_{\star,\Lambda_i}\big(\Gamma_{i-1}(G') \oplus \star\big), \quad 1 \leq i \leq k,
\end{aligned}$$

avec $\Lambda_i = \{\lambda_j : cv_i pv_j \in E(G)\}$. En d'autres termes, nous commençons pas considérer l'union disjointe des $p-1$ premiers sommets périphériques, chacun avec une étiquette qui lui est propre (pv_i est étiqueté avec λ_i), puis nous insérons, un à un et à l'aide de l'étiquette $\star$, les k premiers sommets centraux en les connectant aux sommets périphériques qui doivent leur être adjacents dans G ainsi qu'aux autres sommets centraux déjà présents et étiquetés $\bullet$.

Il nous reste ensuite à traiter tous les sommets jusqu'alors écartés. Pour cela nous commençons par insérer, dans G', le sommet pv_p avec l'étiquette $\bullet$, c'est-à-dire que pv_p rejoint les sommets centraux de G' (sommets auxquels il n'est pas et, par construction, ne pourra pas être adjacent dans G). Puis, nous insérons, un à un, les $c-k$ sommets centraux restant, et nous les connectons comme il se doit. Nous créons donc les expressions :

$$\begin{aligned}
&\Gamma_k(G) := \Gamma_k(G') \oplus \bullet \\
&\Gamma_i(G) := \rho_{\star\to\bullet} \circ \eta_{\star,\bullet} \circ \eta_{\star,\Lambda_i}\Big(\Gamma_{i-1}(G') \oplus \star\Big), \quad k < i \leq c,
\end{aligned}$$

avec $\Lambda_i = \{\lambda_j : j \neq p \wedge cv_i pv_j \in E(G)\}$. Notons que l'arête entre cv_i et pv_p est ajoutée par l'opération $\eta_{\star,\bullet}$. L'expression $\Gamma_c(G)$ finalement obtenue est une $(p+1)$-expression définissant G.

Majorant central. Nous passons maintenant à la deuxième partie de la preuve dans laquelle nous montrons que $\mathrm{cw}(G) \leq c+1$. L'algorithme que nous décrivons ici est très similaire à celui de la première partie. Nous commençons donc par construire un graphe G', sous-graphe induit de G, dans lequel n'apparaissent ni cv_c ni les sommets périphériques qui ne lui sont pas adjacent dans G. Puis nous ajoutons à G' les sommets exclus afin de construire G. L'ensemble d'étiquettes utilisé est $\{\lambda_1, \ldots, \lambda_{c-1}\} \cup \{\star, \bullet\}$.

Nous supposons alors que les sommets périphériques pv_1 à pv_k sont des voisins de cv_c et nous construisons ainsi l'expression définissant G' :

$$\begin{aligned}
&\Gamma_0(G') := \lambda_1 \oplus \cdots \oplus \lambda_{c-1} \\
&\Gamma_i(G') := \rho_{\star\to\bullet} \circ \eta_{\star,\Lambda_i}\Big(\Gamma_{i-1}(G') \oplus \star\Big), \quad 1 \leq i \leq k,
\end{aligned}$$

avec $\Lambda_i = \{\lambda_j : j \neq c \wedge pv_i cv_j \in E(G)\}$. C'est-à-dire que nous construisons la clique formée par les $c-1$ premiers sommets centraux, chacun étiqueté avec sa propre étiquette (cv_i est étiqueté avec λ_i), puis, avec l'étiquette $\star$, nous ajoutons un à un les sommets périphériques qui sont adjacents à cv_c dans G et nous les connectons aux autres sommets centraux auxquels ils sont également adjacents.

Le graphe G' étant définit, nous pouvons y ajouter les sommets manquants en commençant par ajouter cv_c, étiqueté en premier lieu avec l'étiquette $\star$ afin que nous le connections à l'ensemble des sommets de G' et ce avant qu'il rejoigne les sommets périphériques (ré-étiquetage vers $\bullet$). Nous ajoutons ensuite, un à un et grâce à l'étiquette $\star$, les sommets périphériques restants et nous obtenons ainsi la suite d'expressions :

$$\begin{aligned}
&\Gamma_k(G) := \rho_{\star\to\bullet} \circ \eta_{\star,\Lambda}\Big(\Gamma_k(G') \oplus \star\Big) \\
&\Gamma_i(G) := \rho_{\star\to\bullet} \circ \eta_{\star,\Lambda_i}\Big(\Gamma_{i-1}(G) \oplus \star\Big), \quad k < i \leq p,
\end{aligned}$$

avec $\Lambda = \{\lambda_j : 1 \leq j \leq c-1\}$ et $\Lambda_i = \{\lambda_j : pv_icv_j \in E(G)\}$. Avant de conclure, nous n'oublions pas de connecter deux à deux les sommets cv_1 à cv_{c-1} :

$$\Gamma(G) := \eta_{\lambda_1,\Lambda_1} \circ \cdots \circ \eta_{\lambda_{c-2},\Lambda_{c-2}}\big(\Gamma_p(G)\big),$$

avec $\Lambda_i = \{\lambda_j : i < j \leq c-1\}$. L'expression $\Gamma(G)$ ainsi définie est une expression définissant G et manipulant $c+1$ étiquettes. □

Nous laissons là ce résultat et nous passons à la caractérisation de la largeur de clique des graphes scindés en fonction de la largeur arborescente des multi-hyper-graphes.

3.2.1. La scission d'un multi-hyper-graphe

Théorème 3.10. *Soit H un multi-hyper-graphe, alors* $\mathrm{cw}(\mathcal{S}(H)) \leq 3 \cdot 2^{\mathrm{tw}(H)-1} + 3$.

La preuve que nous proposons est inspirée du résultat très ressemblant énoncé par Corneil et Rotics (voir Théorème 2.14). Nous partons donc d'une décomposition arborescente $(\mathcal{T},\mathcal{X})$ de largeur k pour un multi-hyper-graphe H et, grâce à une procédure de programmation dynamique montante sur $\mathcal{T}$, nous construisons une expression en clique de largeur au plus $3 \cdot 2^{k-1} + 3$ pour $\mathcal{S}(H)$.

Le lemme qui suit nous permet de réduire le nombre de décompositions arborescentes à prendre en compte lors de la preuve du Théorème 3.10.

Lemme 3.11. *Pour tout hyper-k-arbre H, il existe une décomposition arborescente enracinée $(\mathcal{T},\mathcal{X})$ telle que, pour tout nœud interne α de $\mathcal{T}$, il existe un sommet $v \in X_\alpha$ qui est présent dans tous les fils de α dans $\mathcal{T}$. De plus, si α n'est pas la racine de $\mathcal{T}$, nous avons que v est absent du père de α dans $\mathcal{T}$.*

Démonstration. Dans cette preuve, nous partons d'un hyper-k-arbre H et nous construisons une décomposition arborescente enracinée $(\mathcal{T},\mathcal{X})$ respectant les contraintes imposées. Soit $\mathcal{Y}$ l'ensemble des $(k+1)$-hyper-cliques de H. Soit u un sommet de degré k dans H et soit Y_u la $(k+1)$-hyper-clique contenant u dans H. Nous définissons un premier nœud τ, avec $X_\tau = Y_u$, et nous le considérons comme étant la racine de $\mathcal{T}$. Puis, pour chaque $(k+1)$-hyper-clique Y_v de H telle que $Y_u \setminus \{u\} \subset Y_v$, nous ajoutons le nœud α en temps que fils de τ, avec $X_\alpha = Y_v$; les contraintes sont alors respectées.

Nous faisons maintenant appel à la procédure récursive suivante. Soit α une feuille de $\mathcal{T}$ et soit v le sommet de X_α qui est absent du père de α dans $\mathcal{T}$. Pour chaque $(k+1)$-hyper-clique Y_w de H, qui contient v mais n'étiquette pas encore un nœud de $\mathcal{T}$, et telle que $|Y_w \cap X_\alpha| = k$, nous ajoutons le nœud β en temps que fils de α, avec $X_\beta = Y_w$. Encore une fois, nous respectons les contraintes.

Il nous reste donc à vérifier que toutes les $(k+1)$-hyper-cliques de H étiquettent un nœud de $\mathcal{T}$. Pour cela, nous considérons Y_u une $(k+1)$-hyper-clique de H qui n'a pas été insérée dans $\mathcal{T}$ quand la procédure récursive prend fin. Il se peut alors que, pour tout nœud α de $\mathcal{T}$, nous ayons $|Y_u \cap X_\alpha| < k$ mais, dans ce cas, H ne serait pas un hyper-k-arbre. Il existe donc un nœud α avec un père β dans $\mathcal{T}$ tel que $|Y_u \cap X_\alpha| = k$ et que Y_u ne contient pas le sommet $v \in X_\beta \setminus X_\alpha$. Ceci implique que $Y_u \cap X_\alpha = Y_u \cap X_\beta$ et,

par conséquent, Y_u aurait dû être à l'origine d'un des fils de β lorsque celui-ci était une feuille de $\mathcal{T}$. □

Afin de simplifier la démonstration du Théorème 3.10, nous introduisons la notion suivante.

Définition 3.12 (Graphe de scission partiel)**.** Soient H un hyper-graphe et $(\mathcal{T},\mathcal{X})$ une décomposition arborescente de H. Soit α un nœud de $\mathcal{T}$, et soit v le sommet de X_α absent du père de α. Le *graphe de scission partiel* de H_α correspond au graphe $\mathcal{S}(H_\alpha)$ privé des sommets centraux correspondants aux sommets de $X_\alpha \setminus \{v\}$ ainsi que des sommets périphériques correspondants aux arêtes de $H_\alpha[X_\alpha \setminus \{v\}]$. Nous notons $\mathcal{Z}(H_\alpha)$ le graphe obtenu.

Soit $\mathbb{P}(E)$ l'ensemble des parties d'un ensemble E et soit X un sous-ensemble de E, nous définissons alors l'opération $\mathbb{P}(E) \uplus X$, qui désigne l'ensemble $\{Y \cup X : Y \in \mathbb{P}(E)\}$, c'est-à-dire l'ensemble des parties de E contenant X.

Démonstration du Théorème 3.10. Tout d'abord, notons qu'il nous suffit de considérer le cas des hyper-graphes. En effet, si $e_1,\dots,e_r$ sont des hyper-arêtes de H adjacentes au même ensemble de sommets et que $pv_1,\dots,pv_r$ sont les sommets périphériques correspondants dans $\mathcal{S}(H)$ alors, au moment où nous insérons pv_1 avec l'étiquette $\ell(pv_1)$, il nous suffit d'insérer le stable de $\mathcal{S}(H)$ formé par les sommets pv_1 à pv_r avec l'étiquette $\ell(pv_1)$. Les opérations suivantes traiteront alors ces sommets comme une seule entité.

Nous partons donc d'une décomposition arborescente $(\mathcal{T},\mathcal{X})$ de H, respectant le Lemme 3.11, et nous commençons par traiter les feuilles de $\mathcal{T}$. La procédure montante que nous appliquons à $\mathcal{T}$ commence par calculer les expressions associées aux feuilles de $\mathcal{T}$ puis, pour chaque nœud α de $\mathcal{T}$ dont tous les fils ont été traités, nous calculons l'expression associée à α. Nous avons donc une étape d'*initialisation* pour les feuilles de $\mathcal{T}$, puis une étape d'*induction* pour les nœuds internes et enfin une étape de *finalisation* pour la racine de $\mathcal{T}$. À la fin de chaque étape d'initialisation ou d'induction le terme que nous obtenons doit respecter un certain nombre de conditions que nous définissons dans l'*hypothèse d'induction* ci-dessous.

Hypothèse d'induction. Soit α un nœud de $\mathcal{T}$ qui vient d'être traité, nous considérons que les sommets du graphe $\mathcal{Z}(H_\alpha)$ associé sont étiquetés comme indiqué ci-après. Nous rappelons que $X_\alpha = \{v_0,\dots,v_k\}$ et que $X_\beta = \{v_1,\dots,v_{k+1}\}$, où est β le père de α dans $\mathcal{T}$. Tous les sommets centraux de $\mathcal{Z}(H_\alpha)$ sont étiquetés avec $\bullet$. Chaque sommet périphérique pv_e de $\mathcal{Z}(H_\alpha)$ est étiqueté λ_e, avec $\lambda_e = \{i : v_i \in e \cap X_\beta\}$. Par définition, aucun sommet de $V(H_\alpha) \setminus X_\alpha$ n'est adjacent à v_{k+1}, de ce fait, $e \cap X_\beta \subseteq X_\alpha \cap X_\beta = \{v_1,\dots,v_k\}$. Le nombre d'étiquettes présentes dans $\mathcal{Z}(H_\alpha)$ est donc au plus $2^k + 1$ et, par hypothèse, nous supposons que $\Gamma(\mathcal{Z}(H_\alpha))$ manipule au plus $3 \cdot 2^{k-1} + 3$ étiquettes.

Nous revenons à présent sur la signification des étiquettes que nous donnons aux sommets périphériques. Un sommet pv_e, étiqueté avec une étiquette différente de $\emptyset$, représente une hyper-arête « en cours de construction », c'est-à-dire que e contient un sous-ensemble de $\{v_1,\dots,v_k\}$ alors que les sommets centraux correspondants ne sont pas présents dans $\mathcal{Z}(H_\alpha)$.

Initialisation. Pour chaque feuille α de $\mathcal{T}$, nous devons définir un terme $\Gamma(\mathcal{Z}(H_\alpha))$ qui respecte les conditions de l'hypothèse d'induction. Encore une fois nous considérons $X_\alpha = \{v_0,\dots,v_k\}$, avec v_0 le sommet absent du père de α dans $\mathcal{T}$. Comme nous l'avons vu $\mathcal{Z}(H_\alpha)$ est composé d'un sommet central cv_0 et de 2^k sommets périphériques, lesquels correspondent aux arêtes de $H[X_\alpha]$ adjacentes à v_0. Chacun de ces 2^k+1 sommets doit avoir sa propre étiquette et, pour respecter l'hypothèse d'induction, cv_0 doit être étiqueté avec $\bullet$ et chaque sommet périphérique pv_{e_i}, avec $e_i \in \mathbb{P}(X_a) \uplus \{v_0\}$ doit être étiqueté avec λ_i, où $\lambda_i = \{j : v_j \in e_i \setminus \{v_0\}\}$. Nous indexons alors les arêtes précitées telles que $\mathbb{P}(X_a) \uplus \{v_0\} = \{e_1,\dots,e_{2^k}\}$ afin de construire l'expression associée à $\mathcal{Z}(H_\alpha)$:

$$\Gamma(\mathcal{Z}(H_\alpha)) := \eta_{\bullet,\Lambda}\Big(\bullet \oplus \lambda_1 \oplus \cdots \oplus \lambda_{2^k}\Big),$$

avec $\Lambda = \mathbb{P}(\{1,\dots,k\})$. L'expression $\Gamma(\mathcal{Z}(H_\alpha))$ ainsi définie manipule 2^k+1 étiquettes et le graphe $\mathcal{Z}(H_\alpha)$ est effectivement composé de 2^k+1 étiquettes.

Induction. Soit α un nœud de $\mathcal{T}$ et soient $\alpha_1,\dots,\alpha_p$ les fils de α dans $\mathcal{T}$. La procédure d'induction se réalise en deux temps. Dans le premier temps, nous construisons le sous-graphe de $\mathcal{Z}(H_\alpha)$ induit par cv_0 ainsi que par tous les sommets périphériques correspondant aux arêtes de $H[X_\alpha]$ adjacentes à v_0. L'expression qui défini ce graphe se construit en suivant la procédure d'initialisation, à cela près que cv_0 est d'abord étiqueté $\star$, puis ré-étiqueté avec $\bullet$ à la fin de la procédure d'induction. Vient alors le deuxième temps durant lequel nous ajoutons un à un, au graphe du temps d'avant, les graphes associés aux fils de α, tout en créant les arêtes nécessaires (en particulier, nous faisons appel à l'étiquette $\diamond$ pour connecter entre eux les sommets centraux). L'expression associée à $\mathcal{Z}(H_\alpha)$ s'écrit donc :

$$\Gamma(\mathcal{Z}(H_\alpha)) := \rho_{\star\to\bullet}\Big(\Gamma_p(\mathcal{Z}(H_\alpha))\Big),$$

où $\Gamma_p(\mathcal{Z}(H_\alpha))$ est obtenue grâce à la récursion suivante :

$$\begin{aligned}
&\Gamma_0(\mathcal{Z}(H_\alpha)) := \eta_{\star,\Lambda_0}\Big(\star \oplus \lambda_1 \oplus \cdots \oplus \lambda_{2^k}\Big)\\
&\Gamma_i(\mathcal{Z}(H_\alpha)) := \rho_{M_i} \circ \eta_{\star,\Lambda_i} \circ \eta_{\star,\bullet} \circ \rho_{\diamond\to\bullet} \circ \eta_{\diamond,\bullet}\Big(\Gamma_{i-1}(\mathcal{Z}(H_\alpha)) \oplus \rho_{\bullet\to\diamond}\Big(\Gamma(\mathcal{Z}(H_{\alpha_i}))\Big)\Big), \quad 1 \le i \le p,
\end{aligned}$$

avec $\Lambda_0 = \mathbb{P}(\{1,\dots,k\})$, $M_i = \{(\lambda, \lambda \setminus \{0\}) : \lambda \in \Lambda_i\}$ et $\Lambda_i = \mathbb{P}(\{1,\dots,k\} \setminus \{\ell_i\}) \uplus \{0\}$, où ℓ_i est l'indice du sommet de X_α absent de X_{α_i}.

Le graphe $\mathcal{Z}(H_\alpha)$ alors défini par $\Gamma(\mathcal{Z}(H_\alpha))$ est composé des étiquettes de $\mathbb{P}(\{1,\dots,k\})$ et de l'étiquette $\bullet$, soit 2^k+1 étiquettes comme le veux l'hypothèse d'induction. Cependant, l'ensemble des étiquettes affectées aux sommets périphériques tout au long de la récursion correspond à l'ensemble $\mathbb{P}(\{0,\dots,k\}) \setminus \{\{0,\dots,k\}\}$, dont la taille est $2^{k+1}-1$, ce qui est bien plus que ce que nous nous devions de respecter. Afin de remédier à ce problème, il nous faut regarder d'un peu plus près chacune des étapes de la récursion.

Soit $Z_{i-1} = \mathrm{val}(\Gamma_{i-1}(\mathcal{Z}(H_\alpha)))$ le graphe de la $(i-1)$-ième étape de la récursion. Nous mettons de côté les trois étiquettes utilisées par les sommets centraux ($\diamond$, $\bullet$ et $\star$) pour nous concentrer sur les étiquettes des sommets périphériques, que nous appelons *étiquettes périphériques*. Les étiquettes périphériques présentes dans Z_{i-1} sont alors définies par

l'ensemble $\mathbb{P}(\{1,\dots,k\})$. Pour construire Z_i, nous unissons Z_{i-1} à $\mathcal{Z}(H_{\alpha_i})$. Les étiquettes périphériques de $\mathcal{Z}(H_{\alpha_i})$ sont définies par $\mathbb{P}(\{0,\dots,k\} \setminus \{\ell_i\})$, où ℓ_i est l'indice du sommet de X_α absent de X_{α_i}. Les étiquettes périphériques qui sont alors manipulées par les opérations menant de $\Gamma_{i-1}(\mathcal{Z}(H_\alpha))$ à $\Gamma_i(\mathcal{Z}(H_\alpha))$, correspondent à l'ensemble :

$$\Big(\mathbb{P}(\{1,\dots,k\} \setminus \{\ell_i\}) \uplus \{0\}\Big) \cup \Big(\mathbb{P}(\{1,\dots,k\} \setminus \{\ell_i\})\Big) \cup \Big(\mathbb{P}(\{1,\dots,k\}) \uplus \{\ell_i\}\Big),$$

dont la taille est $3 \cdot 2^{k-1}$. En utilisant une astucieuse suite de renommages, et en changeant les opérations d'ajouts d'arêtes de façon associée, il est possible de définir une expression $\Gamma(\mathcal{Z}(H_\alpha))$ manipulant un maximum de $3 \cdot 2^{k-1} + 3$ étiquettes.

Finalisation. Il arrive un moment où, dans notre induction, le nœud α que nous considérons n'a pas de père : c'est la racine de notre arbre de décomposition $\mathcal{T}$. Nous accordons donc au vénérable un traitement de faveur.

Dans le graphe $\mathcal{Z}(H)$ qui nous est retourné par la procédure d'induction, il n'apparait aucun des sommets centraux correspondant aux sommets de $X_\alpha \setminus \{v_0\}$, il manque aussi tous les sommets périphériques associés aux hyper-arêtes induites par les sommets de $X_\alpha \setminus \{v_0\}$ dans H ; il nous faut donc les ajouter à $\mathcal{Z}(H)$.

Nous commençons donc par ajouter à $\mathcal{Z}(H)$ tous les sommets périphériques correspondant aux 2^k hyper-arêtes manquantes, tels que le sommet pv_{e_i} correspondant à l'hyper-arête $e_i \in \mathbb{P}(X_\alpha \setminus \{v_0\})$ est étiquetée avec $\lambda_i = \{j : v_j \in e_i\}$. Puis nous insérons un par un les sommets centraux manquants, à l'aide de l'étiquette $\star$. L'expression finale s'écrit alors :

$$\Gamma(\mathcal{S}(H)) := \Gamma_k(\mathcal{S}(H)),$$

où $\Gamma_k(\mathcal{S}(H))$ est obtenue grâce à la récursion suivante :

$$\begin{aligned}
\Gamma_0(\mathcal{S}(H)) &:= \Gamma(\mathcal{Z}(H)) \oplus \lambda_1 \oplus \cdots \oplus \lambda_{2^k} \\
\Gamma_i(\mathcal{S}(H)) &:= \rho_{\star\to\bullet} \circ \eta_{\star,\bullet} \circ \eta_{\star,\Lambda_i}\Big(\Gamma_{i-1}(\mathcal{S}(H)) \oplus \star\Big), \quad 1 \le i \le k,
\end{aligned}$$

avec $\Lambda_i = \mathbb{P}(\{1,\dots,k\}) \uplus \{i\}$. La procédure de finalisation réutilise donc un sous-ensemble des étiquettes manipulées lors de la procédure d'induction, sans nul besoin d'en introduire de nouvelles.

Nous venons donc de définir un algorithme linéaire qui, à partir d'un hyper-k-arbre H, construit une expression en clique de largeur au plus $3 \cdot 2^{k-1} + 3$ pour le graphe de scission de H. Comme nous l'avons vu plus haut, la largeur de clique du graphe de scission d'un multi-hyper-graphe H est stable par suppressions de sommets ou d'arêtes dans H, car ces opérations sont équivalentes à des suppressions de sommets dans $\mathcal{S}(H)$. Par conséquent, pour construire une expression en clique de largeur au plus $3 \cdot 2^{k-1} + 3$ pour un multi-hyper-graphe H de largeur arborescente au plus k, il nous suffit de construire un hyper-k-arbre H' contenant H, puis de construire l'expression $\Gamma(\mathcal{S}(H'))$ avant d'en supprimer les opérations qui concernent les sommets et hyper-arêtes de H' qui ne sont pas dans H. □

La largeur de clique d'un graphe scindé peut donc être majorée en fonction de la largeur arborescente du multi-hyper-graphe associé. Nous ne pouvons cependant pas minorer la largeur de clique d'un graphe scindé en fonction de la largeur arborescente du multi-hyper-graphe associé. En effet, si nous considérons un hyper-graphe H d'ordre n et avec une unique arête reliant tous ses sommets, alors le graphe de scission de H est un graphe complet d'ordre $n+1$, par conséquent la largeur de clique de $\mathcal{S}(H)$ vaut 2 quand la largeur arborescente de H est égale à $n-1$.

De plus, le majorant sur la largeur de clique d'un graphe scindé qui a été mis en lumière est exponentiel en fonction de la largeur arborescente du multi-hyper-graphe associé. Une question se pose alors : existe-t-il un majorant polynomial ? Dans l'article sur lequel nous nous sommes appuyés pour énoncer la preuve du Théorème 3.10, Corneil et Rotics ont également montré l'existence d'un minorant de taille exponentielle relatif à leur problématique. Il serait intéressant de regarder si la preuve qu'ils apportent peut, elle aussi, être adaptée à notre problème, apportant alors une réponse négative à la question précédente.

Majoration de la taille des hyper-arêtes

Si nous définissons un majorant sur la taille des hyper-arêtes des hyper-graphes considérés, alors nous obtenons le résultat ci-dessous, lequel propose un majorant polynomial (et non plus exponentiel) pour la largeur de clique de $\mathcal{S}(H)$ en fonction de la largeur arborescente de H.

Théorème 3.13. *Soit H un multi-hyper-graphe et soit $d \geq 1$ la taille d'une plus grosse hyper-arête de H, alors* $\mathrm{cw}(\mathcal{S}(H)) \leq (d-1) \cdot \mathrm{tw}(H)^{d-1} + 4$.

Démonstration. Nous reprenons ici une décomposition arborescente $(\mathcal{T}, \mathcal{X})$ de H respectant le Lemme 3.11. Soit α un nœud de $\mathcal{T}$ et soit $X_\alpha = \{v_0, \ldots, v_k\}$, avec v_0 le sommet absent du père de α dans $\mathcal{T}$. Chaque sommet périphérique de $\mathcal{Z}(H_\alpha)$, qui correspond à une hyper-arêtes de H_α, est adjacent à au moins un sommet central dans $\mathcal{Z}(H_\alpha)$, par conséquent, il sera connecté à au plus $d-1$ des sommets centraux correspondant aux sommets de l'ensemble $X_\alpha \setminus \{v_0\}$ de taille k. Le nombre d'étiquettes périphériques apparaissant dans $\mathcal{Z}(H_\alpha)$ est donc au plus :

$$\sum_{i=0}^{d-1} \binom{k}{i} \leq \binom{k}{0} + \sum_{i=1}^{d-1} \binom{k}{i} \leq 1 + \sum_{i=1}^{d-1} \frac{k^i}{i!} \leq (d-1) \cdot k^{d-1} + 1.$$

Et, en ajoutant les 3 étiquettes nécessaires pour la manipulation des sommets centraux, nous obtenons une expression en clique de largeur au plus $(d-1) \cdot k^{d-1} + 4$ pour $\mathcal{S}(H)$. □

3.2.2. La scission d'un graphe

Si nous posons la contrainte que les hyper-arêtes du multi-hyper-graphe sur lequel nous travaillons doivent être de taille au plus 2, en d'autre termes, si nous ne considérons

que les graphes scindées qui sont issus de multi-graphes avec boucles, alors nous obtenons des bornes bien plus significatives.

En préambule de ces résultats, nous changeons un peu les notations liées aux sommets périphériques. Dorénavant, le sommet périphérique noté $pv_{i,j}$ référence l'arête reliant les sommets v_i et v_j dans un graphe donné.

Théorème 3.14. *Soit G un multi-graphe, alors* $\mathrm{tw}(G)+1 \leq \mathrm{cw}(\mathcal{S}(G)) \leq \mathrm{tw}(G)+4$.

La preuve de ce théorème est divisée en deux parties. Dans la première partie, bien que la preuve du Théorème 3.13 soit suffisante pour prouver la borne supérieure, nous préférons présenter une démonstration plus simple et utilisant le principe de ré-écriture de termes. Pour prouver la borne inférieure, nous utilisons la notion d'arbre de super-groupes que nous avons présentée dans la Section 3.1.1.

La borne supérieure

Comme nous venons de le dire, le Théorème 3.13 nous donne directement le résultat voulu. En effet, si nous considérons un multi-graphe G, alors il nous faut $\mathrm{tw}(G)$ étiquettes pour différencier les sommets périphériques « en cours de construction » dans $\Gamma(\mathcal{Z}(H_\alpha))$, ainsi qu'une étiquette pour les sommets périphériques « terminés ». Si nous ajoutons à cela les étiquettes $\diamond$, $\bullet$ et $\star$ pour la gestion des sommets centraux, nous atteignons un maximum de $\mathrm{tw}(G)+4$ étiquettes.

Bien que la discussion précédente soit une preuve suffisante de la borne supérieure, nous présentons tout de même une seconde preuve utilisant une technique de ré-écriture de terme. C'est-à-dire que nous partons d'un terme arborescent T définissant G et utilisant $k+1$ sources, puis, dans une procédure montante sur T, nous associons à chaque opération de T une série d'opérations d'expression en clique, lesquelles, mises bout à bout, définissent une expression en clique pour $\mathcal{S}(G)$ utilisant au plus $k+4$ étiquettes.

Afin de simplifier la démonstration, nous adaptons la notion de graphe de scission partiel précédement définie.

Définition 3.15 (Graphe de scission partiel)**.** Soient G un graphe et T un terme arborescent de G. Soit $G_\alpha = \mathrm{val}(T_\alpha)$, avec T_α une sous-expression de T. Le *graphe de scission partiel* de G_α correspond au graphe $\mathcal{S}(G_\alpha)$ privé des sommets centraux correspondant aux sources de G_α ainsi que des sommets périphériques correspondant aux arêtes reliant les sources de G_α entre elles. Nous notons $\mathcal{Z}(G_\alpha)$ le graphe obtenu.

Lemme 3.16. *Soit G un multi-graphe, alors* $\mathrm{cw}(\mathcal{S}(G)) \leq \mathrm{tw}(G)+4$.

Démonstration. Soit T un terme arborescent de G utilisant $k+1$ sources. Nous montrons alors comment obtenir une $(k+4)$-expression Γ de $\mathcal{S}(G)$. L'algorithme que nous décrivons utilise une procédure montante sur T, construisant, pour chaque sous-terme T_α de T, une expression en clique définissant le graphe de scission partiel de $G_\alpha = \mathrm{val}(T_\alpha)$. À chaque étape, nous avons également besoin de retenir les sources de G_α ainsi que les arêtes reliant ces sources, lesquelles sont absentes de $\mathcal{Z}(G_\alpha)$. Pour cela nous définissons

le triplet $(\Gamma_\alpha, S_\alpha, F_\alpha)$, où Γ_α est une expression en clique représentant $\mathcal{Z}(G_\alpha)$, S_α est l'ensemble contenant les sources de G_α et F_α est l'ensemble d'arêtes de $G_\alpha[S_\alpha]$.

Afin que Γ_α utilise au plus $k+4$ étiquettes, nous présentons comment nous étiquetons les sommets de $\mathcal{Z}(G_\alpha)$. Premièrement, nous considérons un sommet cv de $\mathcal{Z}(G_\alpha)$ correspondant à un sommet v de G_α. Le sommet v ne peut donc pas être une source de G_α et nous étiquetons cv comme un « sommet central terminé » avec l'étiquette $\bullet$. Puis nous considérons que le sommet pv de $\mathcal{Z}(G_\alpha)$ correspond à une arête e de G_α. Soient cv_i et cv_j les voisins de pv dans $\mathcal{Z}(G_\alpha)$ et soient v_i et v_j les sommets correspondant dans G_α (v_i et v_j sont les incidences de e dans G_α). Par définition, au moins un des deux sommets v_i ou v_j n'est pas une source de G_α. Si ni v_i ni v_j n'est une source de G_α, alors nous étiquetons pv comme un « sommet périphérique terminé » avec l'étiquette $\diamond$. Sinon, sans perte de généralité, nous supposons que v_i est une source de G_α et, dans ce cas, nous étiquetons pv comme un « sommet périphérique en attente de connexion avec cv_i » en utilisant l'étiquette $\ell(v_i)$.

Nous passons maintenant à la procédure montante sur T.

Opération λ_i. Soit un terme arborescent $T_\alpha = \lambda_i$. Le graphe $\mathcal{S}(G_\alpha)$ est alors un sommet isolé dont le correspondant dans G_α est la source s_i, ce qui implique que le graphe $\mathcal{Z}(G_\alpha)$ est vide. Par conséquent, le triplet associé à T_α est $(\emptyset, \{s_i\}, \emptyset)$.

Opération $\overline{\lambda_i \lambda_j}$. Soit un terme arborescent $T_\alpha = \overline{\lambda_i \lambda_j}$. Dans ce cas, le graphe $\mathcal{S}(G_\alpha)$ est un chemin de longueur 2 dont les extrémités correspondent aux sources s_i et s_j dans G_α, impliquant encore une fois que le graphe $\mathcal{Z}(G_\alpha)$ est vide. En remarquant que $s_i s_j$ est une arête de G_α, nous associons à T_α le triplet $(\emptyset, \{s_i, s_j\}, \{s_i s_j\})$.

Opération $T_\beta /\!\!/ T_\gamma$. Soit un terme arborescent $T_\alpha = T_\beta /\!\!/ T_\gamma$, et soient $(\Gamma_\beta, S_\beta, F_\beta)$ et $(\Gamma_\gamma, S_\gamma, F_\gamma)$ les triplets associés à T_β et T_γ respectivement. Par définition, le graphe $\mathcal{S}(G_\alpha)$ est obtenu en fusionnant les graphes $\mathcal{S}(G_\beta)$ et $\mathcal{S}(G_\gamma)$ suivant les sommets correspondant aux sources communes de G_β et G_γ. Étant donné que ces sommets ne sont pas représentés dans $\mathcal{Z}(G_\beta)$ et $\mathcal{Z}(G_\gamma)$, le graphe $\mathcal{Z}(G_\alpha)$ provient alors de l'union disjointe de $\mathcal{Z}(G_\beta)$ et $\mathcal{Z}(G_\gamma)$, durant laquelle il ne faut pas oublier de connecter les « sommets centraux oubliés » de $\mathcal{Z}(G_\beta)$ à ceux de $\mathcal{Z}(G_\gamma)$. De plus, l'ensemble des sources de G_α correspond à l'union des sources de G_β et G_γ. Finalement, l'ensemble d'arêtes reliant les sources de G_α contient les arêtes reliant les sources de G_β ainsi que les arêtes reliant les sources de G_γ. Il s'ensuit que le triplet correspondant à T_α est $(\rho_{\star\to\bullet} \circ \eta_{\star,\bullet}(\Gamma_\beta \oplus \rho_{\bullet\to\star}(\Gamma_\gamma)), S_\beta \cup S_\gamma, F_\beta \cup F_\gamma)$.

Opération $fg_{\lambda_i}(T_\beta)$. Soit un terme arborescent $T_\alpha = fg_{\lambda_i}(T_\beta)$, et soit $(\Gamma_\beta, S_\beta, F_\beta)$ le triplet associé à T_β. Ici les graphes $\mathcal{S}(G_\alpha)$ et $\mathcal{S}(G_\beta)$ sont isomorphes, ce qui n'est pas le cas des graphes $\mathcal{Z}(G_\alpha)$ et $\mathcal{Z}(G_\beta)$. En effet, si nous considérons s_i, la source de G_β que nous oublions dans G_α, alors s_i n'a pas de correspondant dans $\mathcal{Z}(G_\beta)$ alors qu'il en a un dans $\mathcal{Z}(G_\alpha)$. Nous appelons cv_i le correspondant de s_i dans $\mathcal{Z}(G_\alpha)$. Par conséquent, les « sommets périphériques en attente de connexion avec cv_i » ainsi que les « sommets centraux oubliés » de $\mathcal{Z}(G_\beta)$ sont connectés à cv_i dans $\mathcal{Z}(G_\alpha)$. De plus, toutes les arêtes reliant s_i aux autres sources de G_α n'ont pas de sommet périphérique correspondant dans $\mathcal{Z}(G_\beta)$ alors qu'elles en ont dans $\mathcal{Z}(G_\alpha)$, et ces derniers sont respectivement étiquetés avec

l'étiquette de la source de G_β qu'ils « attendent ». Nous construisons donc l'expression :

$$\Gamma_\alpha := \rho_{\star\to\bullet}\big(\Delta_{k+1}\big),$$

où Δ_{k+1} provient de la récursion suivante, avec j allant de 1 à $k+1$:

$$\Delta_0 := \rho_{i\to\diamond} \circ \eta_{\star,i} \circ \eta_{\star,\bullet}\big(\Gamma_\beta \oplus \star\big)$$
$$\Delta_j := \begin{cases} \rho_{i\to j} \circ \eta_{\star,i}\big(\Delta_{j-1} \oplus \lambda_i\big) & \text{si } s_i s_j \in F_\beta \\ \Delta_{j-1} & \text{si } s_i s_j \notin F_\beta \end{cases}$$

Nous définissons également, $S_\alpha := S_\beta \setminus \{s_i\}$ et $F_\alpha := F_\beta \setminus \{s_i s_j : s_j \in S_\alpha\}$, afin d'associer le triplet $(\Gamma_\alpha, S_\alpha, F_\alpha)$ à T_α.

Pour conclure, nous faisons l'inventaire des étiquettes utilisées pour la construction de Γ. Les $k+1$ sources de T donnent naissance à autant d'étiquettes dans Γ, auxquelles nous ajoutons $\diamond$, $\bullet$ et $\star$. La construction de Γ requiert donc un maximum de $k+4$ étiquettes. □

La borne inférieure

La preuve de la borne supérieure est complète, nous pouvons donc passer à celle de la borne inférieure. Pour cela nous utilisons, comme nous l'avons déjà mentionné, la notion d'arbre de super-groupes à laquelle nous rajoutons un peu de vocabulaire. Un super-groupe est dit *singleton* s'il ne contient qu'un seul sommet. Dans le cas où il contient plusieurs sommets, il est appelé super-groupe *non-singleton*.

Lemme 3.17. *Soit $(\mathcal{J}, \mathcal{Y})$ un arbre de super-groupes de largeur k pour un graphe $\mathcal{S}(G)$. Alors il existe une décomposition arborescente $(\mathcal{T}, \mathcal{X})$ de largeur au plus $k-1$ définissant G.*

Démonstration. Le squelette de l'arbre de décomposition $\mathcal{T}$ est obtenu en supprimant les feuilles d'une copie de $\mathcal{J}$. Nous devons alors remplir les nœuds de $\mathcal{T}$ avec les représentants, dans G, des sommets centraux de $\mathcal{S}(G)$ tout en vérifiant que, pour chaque sommet périphérique $pv_{i,j}$ de $\mathcal{S}(G)$, il existe un nœud de $\mathcal{T}$ contenant à la fois v_i et v_j (qui sont les repésentants des sommets centraux cv_i et cv_j). Il est également de notre devoir de vérifier que chaque nœud de $\mathcal{T}$ contient au maximum k sommets et que l'ensemble des nœuds contenant un sommet donné de G induisent un sous-arbre de $\mathcal{T}$. Ces trois contraintes que nous avons à satisfaire sont respectivement appelées les contraintes d'*arête*, de *taille* et de *connexité*.

Donc, étant donné un sommet interne α de $\mathcal{J}$, il existe un nœud correspondant dans $\mathcal{T}$, que nous appelons a. L'étape de remplissage des nœuds de $\mathcal{T}$ consiste à mettre, dans X_a, le sommet v_i si et seulement si cv_i est dans un singleton de Y_α ; notons qu'alors les contraintes de taille et de connexité sont respectées. Il nous faut, par conséquent, entreprendre la vérification de la contrainte d'arête. Si nous considérons un sommet périphérique $pv_{1,2}$, voisin de cv_1 et cv_2, alors nous observons trois cas qui reflètent les trois différentes façons de créer les arêtes entre $pv_{1,2}$ et cv_1 ainsi qu'entre $pv_{1,2}$ et cv_2.

Dans le premier cas, $pv_{1,2}$, cv_1 et cv_2 sont tous les trois dans des singletons de Y_α. Nous avons donc mis v_1 et v_2 dans X_a lors de l'étape de remplissage, assurant ainsi la validité de la contrainte d'arête.

Dans le second cas $pv_{1,2}$ est toujours dans un singleton de Y_α mais, cette fois-ci, cv_1 et cv_2 sont dans un super-groupe non-singleton de Y_α (notons que ce super-groupe contient exactement cv_1 et cv_2). En descendant dans $\mathcal{J}$ et grâce à la propriété de rafinement ainsi qu'au fait qu'il existe une arête entre cv_1 et cv_2 dans $\mathcal{S}(G)$, il arrive un moment où nous rencontrons un nœud β de $\mathcal{J}$ tel que cv_1 et cv_2 sont tous deux dans des singletons de Y_β. Il existe donc un nœud correspondant dans $\mathcal{T}$ et contenant à la fois les sommets v_1 et v_2. La contrainte d'arête est une nouvelle fois respectée et ce même si $pv_{1,2}$ n'est pas présent dans Y_β.

Le troisième cas est un peu plus excentrique : il existe deux nœuds α et β de $\mathcal{J}$, où α est un ancêtre de β, tels que Y_β contient $pv_{1,2}$ et cv_2 dans deux super-groupes singletons et Y_α contient $pv_{1,2}$, cv_1 et cv_2. Par définition, le sommet cv_1 est dans un singleton de Y_α et cv_2 se trouve être dans un super-groupe non-singleton de Y_α, sans quoi nous pourions nous ramener au premier cas. Par ailleurs, il existe un super-groupe A de Y_α tel que $\{pv_{1,2}\} \subseteq A \subseteq \{pv_{1,2}, \ldots, pv_{1,n}\} \cup \{cv_2, \ldots, cv_n\}$, où n représente le nombre de sommets de G. Soit $\mathcal{P}$ le sous-chemin de $\mathcal{J}$ reliant Y_α à Y_β. Alors, pour chaque nœud γ de $\mathcal{P}$, il existe un super-groupe C de Y_γ tel que $\{pv_{1,2}\} \subseteq C \subseteq A$. Par conséquent, en ajoutant v_1 dans X_c nous fournissons une solution satisfaisant la contrainte d'arête au niveau de X_b, où b et c sont les correspondants respectifs de β et γ dans $\mathcal{T}$. En effet, le remplissage avait mis v_2 dans X_b et nous venons d'y ajouter v_1. L'utilisation du chemin $\mathcal{P}$ assure la contrainte de connexité. De plus, le super-groupe C occupe un espace dans Y_γ qui garantit à v_1 une place libre dans X_c, et permet ainsi la vérification de la contrainte de taille.

Pour conclure, nous notons que la méthode que nous venons de décrire n'introduit pas « de problèmes de collisions » relatifs aux différents sommets périphériques que nous avons à traiter. □

3.2.3. La scission de K_n

Nous restreignons encore un peu plus les graphes que nous considérons pour nous rapprocher du résultat de Makowsky et Rotics [37] montrant que la largeur de clique du graphe $\mathcal{S}(K_n)$ est au moins $n/72$ pour $n \geq 20$. La preuve qu'ils proposent n'utilise pas directement la largeur de clique, mais une autre largeur de graphes, appelée largeur de 2-coloration et, pour atteindre leur but, ils se servent du fait que la largeur de clique d'un graphe est minorée seulement si la largeur de 2-coloration de ce graphe l'est également.

Ce que nous proposons ici consiste à utiliser les arbres de super-groupes pour déterminer la largeur de clique des graphes de scissions des graphes complets. Ceci nous permet, d'une part, d'améliorer un résultat existant et, d'autre part, de renforcer l'idée que les arbres de super-groupes semblent être un outil efficace.

Nous faisons alors appel au Théorème 3.9 que nous avions laissé de côté, pour conclure que la largeur de clique de $\mathcal{S}(K_n)$ est majorée par $n+1$. Puis, nous utilisons les arbres de super-groupes afin de prouver que $n+1$ est également un minorant de $\mathrm{cw}(\mathcal{S}(K_n))$, pour tout $n \geq 4$. Dans les cas où $n \leq 3$, le Théorème 3.14 nous indique que n est un minorant

et on vérifie assez facilement que n est en fait la largeur de clique de $\mathcal{S}(K_n)$.

Théorème 3.18. *La largeur de clique de $\mathcal{S}(K_n)$ est au moins $n+1$, pour tout $n \geq 4$.*

Afin de rendre la preuve de ce théorème plus lisible, nous faisons appel à plusieurs lemmes. Nous commençons donc par les deux Lemmes 3.19 et 3.20 qui sont complémentaires et proposent des résultats généraux, puis nous nous enfonçons plus profondément dans la preuve grâce à d'autres lemmes. Mais, avant d'aller plus loin, nous introduisons quelques notions qui nous seront utiles par la suite.

Un super-groupe A est dit *hétérogène* s'il contient au moins un sommets central et un sommet périphérique, sinon A est dit *homogène*. Nous disons qu'un arbre de super-groupes $(\mathcal{J},\mathcal{Y})$ est *hétérogène* si il existe un nœud interne α de $\mathcal{J}$ tel qu'au moins un des super-groupes de α est hétérogène ; si $(\mathcal{J},\mathcal{Y})$ n'est pas hétérogène alors il est *homogène*.

À partir de maintenant, tous les graphes complets que nous considérons sont d'ordre au moins quatre.

Lemme 3.19. *Soit $(\mathcal{J},\mathcal{Y})$ un arbre de super-groupes homogène pour $\mathcal{S}(K_n)$, alors la largeur de $(\mathcal{J},\mathcal{Y})$ est au moins $n+1$.*

Démonstration. Soit α la racine de $\mathcal{J}$. Si Y_α ne contient que des super-groupes centraux singletons, alors la largeur de Y_α est au moins $n+1$ (un super-groupe par sommet central et au moins un super-groupe pour les sommets périphériques). Par conséquent nous considérons que $A_1,\dots,A_r$ sont les super-groupes centraux non-singletons de Y_α. Si $r=1$, c'est-à-dire que Y_α contient exactement un super-groupe central non-singleton, alors le résultat découle du Lemme 3.24. Dans le cas contraire, le sous-graphe de $\mathcal{S}(K_n)$ induit par $A_1 \cup \dots \cup A_r$ est connexe, et donc l'un des fils de α dans $\mathcal{J}$ contient les sommets de $A_1 \cup \dots \cup A_r$. Nous appelons β ce nœud et nous considérons H_β, le sous-graphe de $\mathcal{S}(K_n)$ associé à β.

Nous observons alors trois cas : soit Y_β contient au moins deux super-groupes centraux non-singletons, soit il admet exactement un super-groupe central non-singleton, soit il ne contient que des super-groupes centraux singletons. Dans le premier cas, le sous-graphe de H_β induit par les super-groupes centraux non-singletons de Y_β est connexe. Nous réutilisons alors l'argument utilisé pour le nœud α afin de descendre dans $\mathcal{J}$ et, par raffinement, nous rencontrons inévitablement l'un des deux autres cas, lesquels sont résolus grâce au Lemme 3.23 d'une part et en appliquant le Lemme 3.24, d'autre part. □

Lemme 3.20. *Soit $(\mathcal{J},\mathcal{Y})$ un arbre de super-groupes hétérogène pour $\mathcal{S}(K_n)$, alors la largeur de $(\mathcal{J},\mathcal{Y})$ est au moins $n+1$.*

Démonstration. Soit α un nœud interne de $\mathcal{J}$ avec pour fils β et γ, et tel que $\mathcal{J}_\alpha$ est hétérogène et $\mathcal{J}_\beta$ et homogène. Premièrement, si Y_β contient au moins un super-groupe central non-singleton, alors nous utilisons les Lemmes 3.23 et 3.24. Nous considérons donc que Y_β ne contient que des super-groupes centraux singletons et nous observons alors deux cas.

Dans le premier cas, l'un des super-groupes hétérogènes de Y_α contient cv_1 et $pv_{1,2}$ (qui sont adjacents dans $\mathcal{S}(K_n)$). Par définition, le sous-graphe de H_α induit par l'ensemble $V = \{cv_1\} \cup \{cv_3, \ldots, cv_n\} \cup \{pv_{1,2}, \ldots, pv_{1,n}\}$ est connexe. Donc V est un sous-ensemble de $V(H_\beta)$ et, en nous référant au Lemme 3.22, nous supposons que $V(H_\beta) = V$. La répartition des super-groupes dans Y_β est la suivante : un super-groupe pour chaque sommet central, ce qui fait $n-1$ super-groupes, plus un super-groupe pour $pv_{1,2}$ et encore au moins un autre pour les sommets périphériques restants. Par conséquent le nombre de super-groupes de Y_β atteind au moins $n+1$.

Dans le deuxième cas, il existe un super-groupe hétérogène de Y_α qui contient cv_1 et $pv_{2,3}$ (qui ne sont pas adjacents dans $\mathcal{S}(K_n)$). Cette fois-ci, la définition nous dit que c'est le sous-graphe de H_α induit par l'ensemble $V = \{cv_1\} \cup \{cv_4, \ldots, cv_n\} \cup \{pv_{1,2}, \ldots, pv_{1,n}\} \cup \{pv_{2,3}\}$ qui est connexe et, ici encore, nous supposons que $V(H_\beta) = V$. Du coup, Y_β est composé de $n-2$ super-groupes pour les sommets centraux, plus deux super-groupes différents pour $pv_{1,2}$ et $pv_{2,3}$, ainsi que d'un dernier super-groupe pour les autres sommets périphériques de V. La largeur de Y_β est donc au moins égale à $n+1$. □

Avant que nous entrions dans les entrailles de la preuve, nous regardons les deux lemmes ci-dessous, qui sont à la fois simples et utiles.

Lemme 3.21. *Soit $(\mathcal{J}, \mathcal{Y})$ un arbre de super-groupes pour $\mathcal{S}(K_n)$. Soit α un nœud interne de $\mathcal{J}$ tel que $pv_{i,j}$ et $pv_{i,k}$ sont des sommets de H_α et que au moins une des arêtes $cv_j pv_{i,j}$ ou $cv_k pv_{i,k}$ est manquante dans H_α. Alors, $pv_{i,j}$ et $pv_{i,k}$ doivent être dans deux super-groupes différents de Y_α.*

Démonstration. Découle directement de la définition d'un super-groupe. □

Lemme 3.22. *Soit $(\mathcal{J}, \mathcal{Y})$ un arbre de super-groupes pour $\mathcal{S}(K_n)$. Soit α un nœud interne de $\mathcal{J}$ tel que $V(H_a) = \{cv_i\} \cup \{pv_{i,j}\}$, avec $1 \leq i \leq k$ et $i < j \leq n$, pour tout $k \geq 2$ fixé. De plus, nous supposons que, quand il est ajouté dans H_α, le sommet cv_j est dans un singleton de Y_α. Alors l'ajout de sommets (périphériques ou centraux) dans H_α augmente le nombre de super-groupes de Y_α d'au moins 1.*

Démonstration. Premièrement, si nous ajoutons, dans H_α, les sommets $cv_i, \ldots, cv_j$, avec $k < i \leq j$, alors nous pouvons regrouper les sommets $pv_{1,i}, \ldots, pv_{1,j}, \ldots, pv_{k,i}, \ldots, pv_{k,j}$ dans un unique super-groupe au lieu de $i-j+1$. Mais, par hypothèse, les sommets $cv_i, \ldots, cv_j$ occupent $i-j+1$ super-groupes, impliquant une augmentation de la largeur de Y_α d'au moins 1. De plus, l'ajout d'un sommet périphérique $pv_{i,j}$ dans H_α, avec $k < i < j$, augmente également le nombre de super-groupes de Y_α. □

Les deux lemmes suivants, couvrent les cas des nœuds qui contiennent au moins un super-groupe central non-singleton, et nous commençons par le cas d'un nœud admettant au moins deux super-groupes centraux non-singletons.

Lemme 3.23. *Soit $(\mathcal{J}, \mathcal{Y})$ un arbre de super-groupes pour $\mathcal{S}(K_n)$. Soit α un nœud interne de $\mathcal{J}$ avec $r \geq 2$ super-groupes centraux non-singletons et tel que $\mathcal{J}_\alpha$ est homogène. Alors la largeur de $(\mathcal{J}, \mathcal{Y})$ est au moins $n+1$.*

Démonstration. Soient $A_1,\dots,A_r$ les super-groupes centraux non-singletons de Y_α. Nous pouvons considérer que tous ces super-groupes proviennent du même fils β de α dans $\mathcal{J}$. Comme $A_1 \cup \dots \cup A_r \subseteq V(H_\beta)$, nous avons que $V(H_\beta)$ contient au moins $2r$ sommets centraux. Si ces sommets centraux sont $cv_1,\dots,cv_p$, avec $p \geq 2r$ alors, par définition, $V(H_\beta)$ contient également les sommets périphériques $pv_{i,j}$, pour tout $1 \leq i < p$ et tout $i < j \leq n$. Encore une fois grâce au Lemme 3.22, nous supposons que $V(H_\beta)$ ne contient que les sommets précités. Le Lemme 3.21 nous indique que nous avons besoin d'au moins $n-p+1$ super-groupes pour stocker les sommets périphériques. Comme Y_β ne contient que des super-groupes centraux singletons, nous devons ajouter p super-groupes pour contenir les sommets centraux. Le nœud β à donc besoin d'au moins $n+1$ super-groupes. □

Le prochain lemme traite le cas des nœuds admettant un unique super-groupe central non-singleton.

Lemme 3.24. *Soit $(\mathcal{J},\mathcal{Y})$ un arbre de super-groupes pour $\mathcal{S}(K_n)$. Soit α un nœud interne de $\mathcal{J}$ tel que $\mathcal{J}_\alpha$ est homogène et que Y_α contient exactement un super-groupe central non-singleton.*

Démonstration. Soit A_1 le super-groupe central non-singleton de Y_α. Nous prouvons d'abord ce lemme dans le cas où $|A_1| \geq 3$, puis nous étudions le cas où A_1 est de taille 2.

Nous considérons donc que $cv_1,\dots,cv_k$ sont les sommets centraux de A_1. Par définition, l'ensemble de sommets $V = \{cv_i\} \cup \{pv_{i,j}\}$, pout tout $1 \leq i \leq k$ et tout $i < j \leq n$, définit un sous-graphe induit de H_α. Nous pouvons alors assumer qu'il existe un nœud interne β de $\mathcal{J}_\alpha$ tel que $V \subseteq V(H_\beta)$ et pour lequel $cv_1,\dots,cv_k$ sont dans k super-groupes centraux singletons (si ce n'est pas le cas, alors la configuration de Y_β est capturée par un des lemmes jusqu'ici définis). De plus, nous supposons que $V(H_\beta) = V$, en faisant appel au Lemme 3.22. Et, en utilisant le Lemme 3.21, nous pouvons affirmer que, pout tout $k+1 \leq i \leq n$, il existe un super-groupe $B_i = \{pv_{1,i},\dots,pv_{k,i}\}$; il y a également un super-groupe qui contient l'ensemble de sommets périphériques $\{pv_{i,j} : 1 \leq i < j \leq k\}$. La largeur de Y_β est par conséquent au moins égale à $n+1$.

Nous considérons maitenant le cas $|A_1| = 2$. Soit cv_1 et cv_2 les deux sommets centraux de A_1. Par définition, l'ensemble de sommets $V = \{cv_1,cv_2\} \cup \{pv_{1,3},\dots,pv_{1,n}\} \cup \{pv_{2,3},\dots,pv_{2,n}\}$ définit un sous-graphe induit de H_α. De plus, pour chaque couple $(pv_{1,i},pv_{2,i})$, il existe un super-groupe de Y_α contenant les sommets de ce couple. Comme le sous-graphe de H_α induit par V est connexe, $V(H_\alpha)$ doit contenir un sommet supplémentaire qui ne peut pas être placé dans l'un des super-groupes de Y_α précédemment défini (nous utilisons pour cela les arguments du Lemme 3.22). La largeur de Y_α est donc n. Nous nous intéressons maintenant au père de α dans $\mathcal{J}$, nous l'appelons β. En nous référent à l'argument précédent, nous constatons qu'aucuns des super-groupes de α ne peuvent être fusionnés dans Y_β. Nous appliquons le même raisonnement aux super-groupes issus du second fils de β dans $\mathcal{J}$. La largeur de Y_β est par conséquent au moins égale à $n+1$. □

Finalement, la preuve du Théorème 3.18 consiste à appliquer le Lemme 3.19, si nous avons à faire à un arbre de super-groupes homogène, et le Lemme 3.20 sinon.

3.3. Les graphes d'incidence

Définition 3.25 (Graphe d'incidence)**.** Soit $G = (V, E)$ un graphe. Le graphe dont les sommets sont les éléments de $V \cup E$ et les arêtes sont les couples (v, e), avec $v \in V$ et $e = \{u, v\} \in E$, est appelé le *graphe d'incidence* de G. Nous notons ce graphe $\mathcal{I}(G)$.

Cette définition fait apparaitre un parallèle entre les graphes scindés et les graphes d'incidence. En effet, si nous partitionnons les sommets d'un graphe $\mathcal{S}(G)$ en une clique et un stable et que nous appliquons cette même partition aux sommets du graphe $\mathcal{I}(G)$ alors nous obtenons deux stables. En d'autre termes, le graphe $\mathcal{I}(G)$ peut être obtenu à partir du graphe $\mathcal{S}(G)$ en supprimant toutes les arêtes induites par la clique de $\mathcal{S}(G)$.

Théorème 3.26. *Soit G un graphe, alors* $\frac{1}{2}(\mathrm{tw}(G) + 1) \leq \mathrm{cw}(\mathcal{I}(G)) \leq \mathrm{tw}(G) + 3$.

Démonstration. La borne supérieure est obtenue en adaptant le Lemme 3.16. Nous définissons alors la notion de *graphe d'incidence partiel* d'un graphe G, que nous notons $\mathcal{J}(G)$. Comme les sommets centraux de $\mathcal{I}(G_\beta)$ ne sont pas deux à deux adjacents, nous pouvons étiqueter les « sommets centraux oubliés » de $\mathcal{J}(G_\beta)$ avec la même étiquette que les « sommets périphériques oubliés », laquelle se nomme $\diamond$. Le traîtement des opérations λ_i et $\overline{\lambda_i \lambda_j}$ ne change pas. Dans le cas de l'opération $T_\beta /\!\!/ T_\gamma$, nous associons à T_α le triplet $(\Gamma_\beta \oplus \Gamma_\gamma, S_\beta \cup S_\gamma, F_\beta \cup F_\gamma)$. Enfin, lors de l'opération $fg_{\lambda_i}(T_\beta)$, nous supprimons l'opération $\eta_{\star,\bullet}$ au sein de l'expression Δ_0 et nous remplaçons $\Gamma_\alpha := \rho_{\star\to\bullet}(\Delta_{k+1})$ par $\Gamma_\alpha := \rho_{\star\to\diamond}(\Delta_{k+1})$. L'expression Γ ainsi obtenue ne fait plus usage de l'étiquette $\bullet$ et totalise donc un maximum de $k + 3$ étiquettes.

Le minorant est issu du Lemme 3.17. Étant donné α, un nœud interne de $\mathcal{J}$, et son correspondant a dans $\mathcal{T}$, nous redéfinissons la procédure de remplissage de X_a en insérant un sommet v_i dans X_a si et seulement si cv_i est dans un super-groupe de α de taille au plus 2. Par conséquent $|X_a| \leq 2 \cdot |Y_\alpha|$, ce qui nous donne le minorant annoncé. Le fait que nous soyons obligés de considérer les super-groupes de taille 2 provient du deuxième cas de la vérification de la propriété d'arête. En effet, comme il n'existe pas d'arête entre cv_1 et cv_2 dans $\mathcal{I}(G)$, il n'existe pas à coup sûr un descendant de α contenant cv_1 et cv_2 dans deux super-groupes singletons. Cependant la nouvelle procédure de remplissage fait que le nœud a de $\mathcal{T}$ respecte la condition d'arête dans ce cas là et n'impacte pas sur la vérification des deux autres cas. □

Une autre méthode pour obtenir $\mathcal{I}(G)$ consiste à subdiviser une et une seule fois chaque arête de G. Ce qui nous amène à considérer un résultat récent de Courcelle [19] résidant dans le lemme suivant.

Lemme 3.27 (Courcelle [19])**.** *Soit G un graphe et soit H un mineur topologique induit* [1] *de G, nous avons* $\mathrm{cw}(H) \leq 2^{\mathrm{cw}(G)+1} - 1$.

Découle naturellement de ce résultat la question de savoir s'il existe ou non un majorant polynomial. Le Théorème 3.26 que nous venons de prouver apporte une réponse négative à cette question. En effet, si nous le combinons avec le Théorème 2.15, nous obtenons le lemme ci-dessous.

1. Obtenu en lissant des sommets d'un sous-graphe induit de G.

Lemme 3.28. *Pour tout k, il existe un graphe $G = \mathcal{I}(H)$ tel que* $\mathrm{cw}(G) = k$ *et* $2^{\lceil k/2 \rceil - 3} \leq \mathrm{cw}(H)$.

Démonstration. Le Théorème 2.15 nous dit qu'il existe un graphe H pour lequel nous avons la relation : $2^{\lfloor \mathrm{tw}(H)/2 \rfloor - 1} \leq \mathrm{cw}(H)$. Et, en nous référant au Théorème 3.26, nous obtenons $\mathrm{cw}(G) = \mathrm{cw}(\mathcal{I}(H)) \leq \mathrm{tw}(H) + 3$ et donc $\mathrm{cw}(G) - 3 \leq \mathrm{tw}(H)$. Il nous suffit alors de remplacer ce dernier résultat dans le premier. □

Une autre question que soulève la définition proche de celle d'un graphe d'incidence est de savoir combien d'étiquettes sont nécessaires pour construire l'expression en clique d'un graphe $\mathcal{I}^+(G)$, lequel est obtenu en subdiviant au moins une fois chaque arête de G. La réponse est, encore une fois, $\mathrm{tw}(G) + 3$. En effet, lorsque nous considérons l'expression $T_\alpha = fg_{\lambda_i}(T_\beta)$, alors dans le graphe $\mathcal{I}^+(G_\beta)$ associé, au lieu de considérer un sommet reliant deux sources v_i et v_j de G_β, nous considérons le chemin P reliant, sans les contenir, les sommets v_i et v_j. Et, si P contient au moins deux sommets, nous construisons l'expression Π_j, utilisant les étiquettes λ_i, $\lambda_j = \ell(v_j)$, $\diamond$ et $\star$, telle que les extrémités de P sont étiquetées avec λ_i et λ_j et que les sommets internes de P sont étiquetés avec $\diamond$; l'étiquette $\star$ est utilisée pour étendre le chemin. Nous remplaçons alors l'étape de récursion par :

$$\Delta_j := \begin{cases} \rho_{\lambda_i \to \lambda_j} \circ \eta_{\star,\lambda_i}\big(\Delta_{j-1} \oplus \lambda_i)\big) & \text{si } v_i v_j \in F_\beta \text{ et } |P| = 1 \\ \rho_{\lambda_i \to \diamond} \circ \eta_{\star,\lambda_i}\big(\Delta_{j-1} \oplus \Pi_j\big) & \text{si } v_i v_j \in F_\beta \text{ et } |P| \geq 2 \\ \Delta_{j-1} & \text{si } v_i v_j \notin F_\beta \end{cases}$$

Comme nous l'avons vu lors de la définition, la largeur de clique s'applique également aux graphes orientés. La définition des graphes d'incidence étant elle aussi compatible avec la notion de graphes orientés, il est tout à fait raisonnable de considérer le cas desdits graphes orientés.

Théorème 3.29. *Soit G un graphe orienté, alors* $\mathrm{cw}(\mathcal{I}(G)) \leq 2 \cdot \mathrm{tw}(G) + 4$.

Démonstration. La preuve de ce théorème est à peu de choses près équivalente à celle du Théorème 3.26, sauf que nous devons retenir l'orientation des arcs de G. Pour cela, nous redéfinissons la manière dont nous étiquetons le graphe d'incidence partiel. Pour cela, nous considérons T_α un sous-terme de T ainsi que $\mathcal{J}(G_\alpha)$ le graphe d'incidence partiel associé, avec $G_\alpha = \mathrm{val}(T_\alpha)$.

Premièrement, nous considérons un sommet cv de $\mathcal{J}(G_\alpha)$ correspondant à un sommet v de G_α. Le sommet v ne peut donc pas être une source de G_α et nous étiquettons cv comme un « sommet central oublié » avec l'étiquette $\diamond$. Puis nous considérons que le sommet pv de $\mathcal{J}(G_\alpha)$ correspond à un arc a de G_α. Soit cv_i le voisin entrant de pv dans $\mathcal{J}(G_\alpha)$, soit cv_j le voisin sortant de de pv dans $\mathcal{J}(G_\alpha)$ et soient v_i et v_j les sommets correspondant dans G_α. Par définition, au moins un des deux sommets v_i et v_j n'est pas une source de G_α. Si ni v_i ni v_j n'est une source de G_α, alors nous étiquetons pv comme un « sommet périphérique oublié », encore une fois grâce à l'étiquette $\diamond$. Sinon, si v_i est une source de G_α, alors nous étiquetons pv comme un « sommet périphérique en

attente d'une connexion entrante venant de cv_i » en utilisant l'étiquette λ_i^-, où $\lambda_i = \ell(v_i)$. Sinon, v_j est une source de G_α et, dans ce cas, nous étiquetons pv comme un « sommet périphérique en attente d'une connexion sortante vers cv_j » en utilisant l'étiquette λ_j^+, où $\lambda_j = \ell(v_j)$.

La fin de la preuve peut alors aisément être adaptée en se basant sur celle du Théorème 3.26. Le dernier décompte nous donne $k+1$ étiquettes « entrantes » et autant d'étiquettes « sortantes ». Ainsi, en ajoutant les étiquettes $\diamond$ et $\star$, nous atteignons un maximum de $2k+4$ étiquettes. $\square$

CHAPITRE

4

LA LARGEUR ARBORESCENTE SPÉCIALE

Sommaire

4.1. Définitions **40**
4.1.1. Comparaison avec les largeurs existantes 43
4.2. Quelques résultats de complexité **44**
4.2.1. Graphes de largeurs égales 44
4.2.2. Graphes récalcitrants 47
4.2.3. Un résultat polynomial intéressant 50
4.2.4. Graphes de « petite » largeur arborescente 51
4.3. Largeur arborescente spéciale et mineurs de graphes **53**
4.4. Récapitulatif **57**

Dans ce chapitre, nous restons au contact des deux largeurs précédentes au travers d'une nouvelle largeur appelée largeur arborescente spéciale. Ce paramètre relativement récent, introduit par Courcelle [17, 18], peut en effet être défini à partir de restrictions sur la largeur arborescente « classique » d'une part ainsi que sur la largeur de clique d'autre part.

L'intérêt majeur de ce nouvel invariant de graphe est qu'il permet le traitement de problèmes exprimables en logique monadique du second ordre avec quantification sur les sommets et les arêtes ($MSOL_2$), en adaptant les automates conçus pour les expressions en clique. Sans cette adaptation, les automates liés aux expressions en clique se limitent à des problèmes avec quantification sur les sommets ($MSOL_1$). Les automates associés à la largeur arborescente spéciale sont aussi plus petits et donc plus performants

que ceux associés à la largeur arborescente « classique », lesquels résolvent également des problèmes avec quantification sur les sommets et les arêtes.

Le but de notre travail n'est pas d'approfondir l'étude des automates précédement cités, mais plutôt de détermier quelles sont les classes de graphes auxquelles nous pouvons appliquer ces automates. En effet, afin de résoudre de manière efficace les divers problèmes exprimables en logique monadique du second ordre, il faut que les paramètres associés aux graphes sur lequels nous travaillons ne soient pas trop élevés. Nous allons donc nous concentrer sur le problème de décision lié à la largeur arborescente spéciale qui, étant donné un entier k et un graphe G, retourne *vrai* si et seulement si la largeur arborescente spéciale de G est au plus k. Comme pour les problèmes de décisions liés à la largeur arborescente « classique » ainsi qu'à la largeur de clique, ce problème est NP-complet dans le cas général, comme nous le montre par exemple le Théorème 4.23. C'est pourquoi nous nous penchons sur l'étude de différentes classes de graphes bien connues.

Dans la Section 4.1 nous introduisons la largeur arborescente spéciale en temps que restriction de la largeur arborescente « classique ». Nous définissons également une restriction sur la largeur de clique et nous montrons que la largeur de clique spéciale ainsi obtenue est équivalente à la largeur arborescente spéciale. Puis, nous proposons une comparaison rapide de la nouvelle largeur avec celles dont elle est issue. La Section 4.2 présente un certain nombre de résultats de complexité concernant le problème de décision lié à la largeur arborescente spéciale. Nous étudions ensuite, en Section 4.3, le comportement de la largeur arborescente spéciale sur les classes de graphes stables par minoration. Pour finir, la Section 4.4 propose un tableau récapitulant les différents résultats de complexité que nous avons obtenus, agrémenté d'une comparaison avec les largeurs de graphes en corrélation directe avec la largeur arborescente spéciale.

4.1. Définitions

Dans cette section nous présentons en premier lieu la largeur arborescente spéciale, puis la largeur de clique spéciale et enfin nous montrons comment passer d'une décomposition arborescente spéciale de largeur k à une $(k+1)$-expression spéciale et *vice versa*. Les algorithmes présentés ont un temps d'exécution linéaire en fonction du nombre de sommets du graphe en entrée.

Définition 4.1 (Décomposition arborescente spéciale)**.** Soit $G = (V, E)$ un graphe. Une *décomposition arborescente spéciale* de G est un couple $(\mathcal{T}, \mathcal{X})$, avec $\mathcal{X} = \{X_1, \ldots, X_p\}$ une famille de sous-ensembles de V et $\mathcal{T}$ un arbre enraciné et orienté dont les nœuds sont étiquetés par ces sous-ensembles, tel que :

- l'union de tous les éléments de $\mathcal{X}$ est égale à V ;
- pour chaque arête e de E, il existe un $X_i \in \mathcal{X}$ qui contient tous les sommets de e ;
- pour tout sommet v de V, l'ensemble des éléments de $\mathcal{X}$ contenant v induit un sous-chemin (orienté) dans $\mathcal{T}$.

La *largeur* de $(\mathcal{T}, \mathcal{X})$ est définie par le maximum sur la taille des éléments de $\mathcal{X}$ moins 1. La largeur minimale, parmi toutes les décompositions arborescentes spéciales de G,

représente la *largeur arborescente spéciale* de G et est notée $\mathrm{stw}(G)$.

Une conséquence directe de cette définition est que, étant donné un nœud α de $\mathcal{T}$, tous les fils de α dans $\mathcal{T}$ sont deux à deux disjoints. Ceci implique qu'il ne peut exister de jolie décomposition arborescente spéciale sans une adaptation de la définition correspondante (Définition 2.7). Pour ce faire, nous gardons inchangée la définition des nœuds de départ, d'oubli et d'introduction. Un *nœud de jonction* α a toujours deux fils β et γ cependant, dans le cadre de la décomposition arborescente spéciale, ces trois nœuds doivent être tels que $X_\alpha = X_\beta \cup X_\gamma$ et $X_\beta \cap X_\gamma = \emptyset$. Nous pouvons alors énoncer l'équivalent du Lemme 2.8.

Lemme 4.2. *Pour toute décomposition arborescente spéciale de G il existe une jolie décomposition arborescente spéciale de même largeur qui représente G.*

Comme annoncé plus haut, la largeur arborescente spéciale est également issue d'une restriction appliquée aux expressions en clique. Dans la définition suivante on dénote par $\pi(\Gamma)$ l'ensemble des étiquettes présentes dans $\mathrm{val}(\Gamma)$.

Définition 4.3 (Expression en clique spéciale)**.** Soit $G = (V, E)$ un graphe. Une expression en clique Γ est dite *spéciale* si elle respecte les contraintes suivantes :
- pour tout sous-terme Γ_α de Γ, seul $\bullet$ peut étiqueter plusieurs sommets de $\mathrm{val}(\Gamma_\alpha)$;
- l'opération $\Gamma_\alpha \oplus \Gamma_\beta$ n'est autorisée que si $\pi(\Gamma_\alpha) \cap \pi(\Gamma_\beta) \subseteq \{\bullet\}$;
- l'opération $\eta_{\lambda_i,\lambda_j}(\Gamma)$ n'est autorisée que si $\lambda_i \neq \bullet$ et $\lambda_j \neq \bullet$;
- l'opération $\rho_{\bullet \to \lambda_i}(\Gamma)$ n'est pas autorisée ;
- l'opération $\bullet$ n'est pas autorisée.

La *largeur* de Γ représente le nombre d'étiquettes différentes de $\bullet$ utilisées pour la construction du terme. La *largeur de clique spéciale* de G, notée $\mathrm{scw}(G)$, est la largeur minimale parmi toutes les expressions en clique spéciales qui représentent un étiquetage de G.

Les méta-opérations définies pour les expressions en clique « classiques » restent valables dans le cadre des expressions en clique spéciales, sous réserve de satisfaire les restrictions précédemment introduites.

Théorème 4.4 (Courcelle [18])**.** *Pour tout graphe G nous avons* $\mathrm{stw}(G) = \mathrm{scw}(G) - 1$. *De plus, il existe des algorithmes linéaires pour passer d'une décomposition arborescente spéciale de G de largeur k à une $(k+1)$-expression spéciale de G et inversement.*

Démonstration. L'équivalence entre les deux largeurs découle directement de l'existence des algorithmes de l'énoncé, lesquels sont étudiés dans les Lemmes 4.5 et 4.6. □

Étant donné un graphe G, le lemme qui suit montre comment passer d'une décomposition arborescente spéciale de G de largeur k à une $(k+1)$-expression spéciale de G. En particulier il existe une $(\mathrm{stw}(G)+1)$-expression spéciale de G, ce qui nous permet d'établir l'inégalité $\mathrm{scw}(G) \leq \mathrm{stw}(G) + 1$.

Lemme 4.5. *Il existe un algorithme linéaire qui, à partir d'une décomposition arborescente spéciale $(\mathcal{T}, \mathcal{X})$ de G de largeur k, construit une $(k+1)$-expression spéciale $\Gamma(G)$.*

Démonstration. D'après le Lemme 4.2, nous pouvons considérer que $(\mathcal{T},\mathcal{X})$ est jolie. Avant de commencer nous rappelons quelques notations. Soit α un nœud de $\mathcal{T}$ et X_α le sous ensemble de V qui lui correspond. Le sous arbre de $\mathcal{T}$ enraciné en α est noté T_α et induit le sous-graphe de G appelé G_α. Le terme $\Gamma(G_\alpha)$ est le sous-terme spécial de $\Gamma(G)$ qui représente un étiquetage de G_α. Comme chaque étiquette de $\Gamma(G_\alpha)$ (hormis $\bullet$) ne peut étiqueter qu'un seul sommet de G_α, nous nous servons de ces étiquettes pour identifier les sommets de X_α. C'est-à-dire que si $X_\alpha = \{v_1,\dots,v_p\}$, chaque v_i est étiqueté λ_i dans G_α, les autres sommets de G_α sont étiqueté avec $\bullet$.

Nous étudions à présent chacun des quatres types de nœuds de $(\mathcal{T},\mathcal{X})$.

Les nœuds de départ. Soit α un nœud de départ de $\mathcal{T}$. Dans ce cas très simple nous construisons le terme :

$$\Gamma(G_\alpha) := \lambda_1.$$

Les nœuds d'oubli. Soit α un nœud d'oubli de $\mathcal{T}$ avec pour fils β. Nous avons $X_\alpha = \{v_1,\dots,v_{p-1}\}$ et $X_\beta = \{v_1,\dots,v_p\}$; v_p est le sommet oublié. Avant de ré-étiqueter v_p avec $\bullet$, nous ajoutons à G_β les arêtes $v_iv_p \in E(G[X_\beta])$, afin d'obtenir le graph G_α. Cette suite d'opérations se traduit par :

$$\Gamma(G_\alpha) := \rho_{\lambda_p\to\bullet} \circ \eta_{\lambda_p,\Lambda}\Big(\Gamma(G_\beta)\Big),$$

avec $\Lambda = \{\lambda_i : v_iv_p \in E(G[X_\beta])\}$.

Les nœuds d'introduction. Soit α un nœud d'introduction de $\mathcal{T}$ avec pour fils β. Nous avons $X_\alpha = \{v_1,\dots,v_p\}$ et $X_\beta = \{v_1,\dots,v_{p-1}\}$; v_p est le sommet introduit. Pour obtenir G_α il nous suffit d'ajouter v_p à G_β avec l'étiquette λ_p qui n'est pas présente dans G_β. Nous obtenons donc le terme :

$$\Gamma(G_\alpha) := \Gamma(G_\beta) \oplus \lambda_p.$$

Les nœuds de jonction. Soit α un nœud de jonction avec pour fils β et γ. Nous avons $X_\beta = \{u_1,\dots,u_p\}$ et $X_\gamma = \{v_1,\dots,v_q\}$. Pour le moment $\pi(\Gamma(G_\beta))$ et $\pi(\Gamma(G_\gamma))$ ont en commun $\min\{p,q\}$ étiquettes différentes de $\bullet$, ce qui nous empèche de construire l'union disjointe de G_β et G_γ. Pour remédier à ce problème, nous ré-étiquetons les sommets de G_γ avec les étiquettes $\lambda_{p+1},\dots,\lambda_{p+q}$. Ce qui nous amène à :

$$\Gamma(G_\alpha) := \Gamma(G_\beta) \oplus \rho_M\Big(\Gamma(G_\gamma)\Big),$$

avec $M = \{(\lambda_i,\lambda_{p+i}) : 1 \le i \le q\}$. Remarquons en passant que $|X_\alpha| = p+q$.

Si k est la largeur de $(\mathcal{T},\mathcal{X})$, alors la taille maximale d'un nœud de $\mathcal{T}$ est $k+1$. Et donc, pour chacun des nœuds α de $\mathcal{T}$ nous construisons une expression en clique spéciale de taille au plus $k+1$ qui représente le graph G_α associé. □

Nous allons maintenant montrer comment passer d'une k-expression spéciale de G à une décomposition arborescente spéciale de G de largeur $k-1$. Il est alors possible d'obtenir une décomposition arborescente spéciale de G de largeur $\mathrm{scw}(G)-1$, impliquant de ce fait l'inégalité $\mathrm{stw}(G) \le \mathrm{scw}(G)-1$. En combinant ce résultat avec celui du lemme précédent, nous arrivons à la conclusion que $\mathrm{stw}(G) = \mathrm{scw}(G)-1$.

Lemme 4.6. *Il existe un algorithme linéaire qui, à partir d'une k-expression spéciale* Γ, *construit une décomposition arborescente spéciale* $(\mathcal{T},\mathcal{X})$ *de* G *de largeur au plus* $k-1$.

Démonstration. Nous construisons $(\mathcal{T},\mathcal{X})$ grâce à une procédure montante sur $\Gamma(G)$. Au début de la procédure, nous considérons que $\mathcal{X}$ est vide. Puis, pour chaque sous-expression Γ_α de Γ, nous construisons un arbre T_α que nous transmettons à l'opération suivante et nous ajoutons, dans $\mathcal{X}$, l'ensemble X_α qui étiquette la racine de T_α.

Opération λ_i. Soit une expression $\Gamma_\alpha = \lambda_i$ et soit v le sommet du graphe $\mathrm{val}(\Gamma_\alpha)$. Nous ajoutons alors l'élément $X_\alpha = \{v\}$ dans $\mathcal{X}$ puis nous transmettons l'arbre T_α qui est un sommet isolé.

Opération $\Gamma_\beta \oplus \Gamma_\gamma$. Soit une expression $\Gamma_\alpha = \Gamma_\beta \oplus \Gamma_\gamma$, où Γ_β et Γ_γ sont respectivement associées aux arbres T_β et T_γ, dont les racines sont étiquetées avec X_β et X_γ. L'arbre T_α est obtenu à partir de l'union disjointe de T_β et T_γ, puis en ajoutant un nouveau nœud en temps que père des racines de T_β et T_γ. Enfin, nous ajoutons à $\mathcal{X}$ l'ensemble $X_\alpha = X_\beta \cup X_\gamma$.

Opération $\eta_{\lambda_i,\lambda_j}(\Gamma_\beta)$. Soit une expression $\Gamma_\alpha = \eta_{\lambda_i,\lambda_j}(\Gamma_\beta)$, où Γ_β est associée à l'arbre T_β, dont la racine est étiquetée avec X_β. Dans cette situation, nous n'apportons pas de changement à $\mathcal{X}$ et nous transmettons T_β. Notons que X_β contient les sommets étiquetés avec λ_i et λ_j, il contient donc l'arête entre ces deux sommets.

Opération $\rho_{\lambda_i\to\lambda_j}(\Gamma_\beta)$. Soit une expression $\Gamma_\alpha = \rho_{\lambda_i\to\lambda_j}(\Gamma_\beta)$, où Γ_β est associée à l'arbre T_β, dont la racine est étiquetée avec X_β. Dans le cas où $\lambda_j \neq \bullet$, nous n'apportons pas de changement à $\mathcal{X}$ et nous transmettons T_β. Par contre, si $\lambda_j = \bullet$, alors nous définissons T_α en ajoutant un père à la racine de T_β et nous ajoutons l'ensemble $X_\alpha = X_\beta \setminus \{v\}$ dans $\mathcal{X}$, avec $\ell(v) = \lambda_i$ dans $\mathrm{val}(\Gamma_\beta)$.

Soit $\mathcal{T}$ l'arbre transmis par la dernière opértion de Γ. Il est assez aisé de vérifier que, après traitement de la dernière opération de Γ, le couple $(\mathcal{T},\mathcal{X})$ respecte bien les propriétés d'une décomposition arborescente spéciale de G. De plus, pour chaque sous-expression Γ_α de Γ, le nombre de sommets étiquetés avec une étiquette différente de $\bullet$ est au plus k, par conséquent, le traitement des opérations $\Gamma_\beta \oplus \Gamma_\gamma$ et $\rho_{\lambda_i\to\lambda_j}(\Gamma_\beta)$ nous garantit que tous les éléments de $\mathcal{X}$ sont de taille au plus k. Finalement la décomposition arborescente spéciale que nous obtenons est de largeur au plus $k-1$. □

4.1.1. Comparaison avec les largeurs existantes

D'après la définition de la largeur arborescente spéciale, il va de soi qu'une décomposition arborescente spéciale est une décomposition arborescente « classique ». De plus, il est assez évident qu'une décomposition en chemin est une décomposition arborescente spéciale. Nous pouvons alors déduire de ces deux observations l'inégalité suivante.

Lemme 4.7. *Soit* G *un graphe, alors* $\mathrm{tw}(G) \leq \mathrm{stw}(G) \leq \mathrm{pw}(G)$.

De façon similaire, la définition de la largeur de clique spéciale implique qu'une expression en clique spéciale est également une expression en clique « classique ». Nous avons donc le résultat suivant.

Lemme 4.8. *Soit G un graphe, alors* $\mathrm{cw}(G) \leq \mathrm{stw}(G) + 2$.

Un sommet adjacent à tous les autres sommets du graphe est appelé sommet universel. Nous définissons l'ajout d'un sommet universel v à un graphe G par la notation $G \otimes v$.

Lemme 4.9 (Courcelle [18])**.** *Soit G un graphe, alors* $\mathrm{stw}(G \otimes v) = \mathrm{pw}(G \otimes v) = \mathrm{pw}(G) + 1$.

Étant donné une classe de graphes $\mathcal{C}$, nous définissons $\mathrm{tw}(\mathcal{C})$ comme étant la borne supérieure des largeurs arborescentes des graphes de $\mathcal{C}$; de façon similaire nous définissons $\mathrm{stw}(\mathcal{C})$, $\mathrm{pw}(\mathcal{C})$ et $\mathrm{cw}(\mathcal{C})$.

Lemme 4.10 (Courcelle [18])**.** *Pour toute classe de graphes $\mathcal{C}$, nous avons* $\mathrm{pw}(\mathcal{C}) < \infty \Rightarrow \mathrm{stw}(\mathcal{C}) < \infty \Rightarrow \mathrm{tw}(\mathcal{C}) < \infty$ *et* $\mathrm{stw}(\mathcal{C}) < \infty \Rightarrow \mathrm{cw}(\mathcal{C}) < \infty$. *Les implications réciproques ne sont cependant pas vérifiées.*

Démonstration. Les différentes implications nous sont données par les Lemmes 4.7 et 4.8. Pour montrer que $\mathrm{stw}(\mathcal{C}) < \infty \not\Rightarrow \mathrm{pw}(\mathcal{C}) < \infty$, il nous suffit de considérer la classe des arbres finis dont la largeur arborescente spéciale est au plus 1 mais pour laquelle la largeur de chemin n'est pas majorée. En ajoutant un sommet universel à tous les arbres finis, nous obtenons une classe de graphes de largeur arborescente au plus 2 et, d'après le Lemme 4.9, il n'existe pas de majorant pour la largeur arborescente spéciale des graphes de cette classe. Le fait que $\mathrm{cw}(\mathcal{C}) < \infty \not\Rightarrow \mathrm{stw}(\mathcal{C}) < \infty$ s'établit en considérant la classe des graphes complets. En effet la largeur de clique des ces graphes est au plus 2 quand leur largeur arborescente, et donc leur largeur arborescente spéciale, n'admet pas de majorant constant. □

Nous avons maintenant assez de matière pour passer aux résultats de complexité qui nous intéressent.

4.2. Quelques résultats de complexité

4.2.1. Graphes de largeurs égales

Soit I un ensemble d'intervalles réels. Les sommets du *graphe d'intervalles* G correspondant à I sont les intervalles de I et il y a une arête entre deux sommets de G si et seulement si l'intersection des deux intervalles correspondant est non vide.

Lemme 4.11. *Soit G un graphe d'intervalles, alors* $\mathrm{tw}(G) = \mathrm{pw}(G)$.

Un graphe est un *cographe* s'il peut être obtenu par une suite d'unions disjointes et de complémentations en partant de sommets isolés. Les graphes compléments sont les graphes dont la largeur de clique est au plus 2 [23].

Lemme 4.12 (Bodlaender et Möhring [8])**.** *Soit G un cographe, alors* $\mathrm{tw}(G) = \mathrm{pw}(G)$.

Soit $\sigma = (\sigma_1, \ldots, \sigma_n)$ une permutation des entiers de 1 à n. Le *graphe de permutation* G représentant σ contient n sommets et il y a une arête entre deux sommets v_i et v_j de G si et seulement si $i < j$ et $\sigma_i > \sigma_j$.

Lemme 4.13 (Bodlaender *et al.* [6])**.** *Soit G un graphe de permutation, alors* $\mathrm{tw}(G) = \mathrm{pw}(G)$.

Ce résultat généralise le précédent étant donné que tout cographe est un graphe de permutation. En effet, soient G_σ et G_τ deux graphes de permutations, avec σ une permutation des entiers de 1 à n et τ une permutation des entiers de 1 à m. Le graphe $G_\pi = G_\sigma \oplus G_\tau$ est définit par la permutation π sur les entiers de 1 à $n+m$, où $\pi_i = \sigma_i$ pour $1 \leq i \leq n$ et $\pi_{i+n} = \tau_i + n$ pour $1 \leq i \leq m$. Le graphe $G_{\overline{\sigma}} = \overline{G_\sigma}$ est définit par la permutation $\overline{\sigma}$ sur les entiers de 1 à n, où $\overline{\sigma}_i = \sigma_{n+1-i}$. En d'autres termes, l'opération d'union disjointe de deux cographes se traduit par une « concaténation » de deux permutations et l'opération de complémentation équivaut à « retourner » une permutation.

Comme nous l'avons vu dans le Lemme 4.7, la largeur arborescente spéciale est comprise entre la largeur arborescente et la largeur de chemin, de ce fait ces trois largeurs sont égales pour chaque graphe membre de l'une des trois classes précédentes. De plus, il existe un algorithme linéaire [38] qui résoud le problème de décision lié à la largeur arborescente pour les graphes de permutations. Il nous suffit ensuite de rappeler que les graphes d'intervalles sont des graphes triangulés et qu'il existe un algorithme polynomial résolvant le problème de décision lié à la largeur arborescente sur ces graphes. En effet, la largeur arborescente d'un graphe triangulé G est égale à la taille de la plus grande clique de G moins 1 et Gavril à montré dans [29] qu'il est possible d'énumérer les cliques maximales d'un graphe triangulé en temps polynomial.

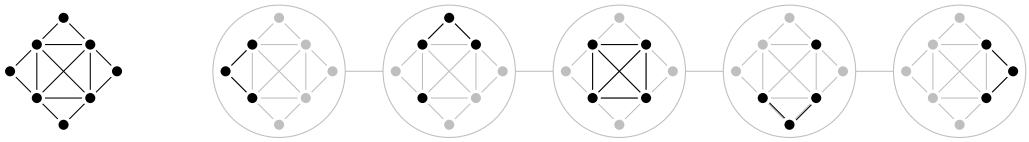
Théorème 4.14. *Le problème de décision lié à la largeur arborescente spéciale se résoud en temps polynomial pour les graphes d'intervalles et en temps linéaire pour les graphes de permutations ainsi que pour les graphes compléments.*

Le cas des graphes scindés

Nous reprenons les notations introduites dans le Chapitre 3. La classe des graphes scindés est exactement la classe des graphes qui sont à la fois triangulés et co-triangulés. Par conséquent la largeur arborescente d'un graphe scindé G est égale à $\omega - 1$ (où ω est la taille de la clique centrale de G). Cette valeur est aussi un minorant pour la largeur de chemin de G. Le majorant pour la largeur de chemin de G est ω. En effet, pour chaque clique périphérique X_i de G, nous construisons un nœud $Y_i = X_i \cup X_0$ de taille $\omega + 1$ et, en arangeant ces nœuds dans un chemin, nous obtenons une décomposition en chemin de G de largeur ω. Nous avons donc la relation $\mathrm{tw}(G) \leq \mathrm{pw}(G) \leq \mathrm{tw}(G) + 1$, pour tout graphe scindé G.

Contrairement à ce que prétend le Lemme 9 de [34], il existe des graphes scindés qui ne sont pas des graphes d'intervalles et dont la largeur de chemin est $\omega - 1$. Pour s'en convaincre il suffit de considérer le contre exemple illustré en Figure 4.1 mettant en scène le graphe S_4 (une caractérisation des graphes d'intervalles par sous-graphes induits interdits a été donnée par Lekkerkerker et Boland dans [36]).

Une bonne caractérisation de la largeur de chemin des graphes scindés est la suivante.



(a) Le graphe S_4. (b) Une décomposition en chemin de S_4 de largeur 3.

FIGURE 4.1. : Un graphe scindé qui n'est pas un graphe d'intervalle dont la largeur de chemin est 3 et dont la clique centrale est de taille 4.

Lemme 4.15. *Soit G un graphe scindé, alors* $\mathrm{pw}(G) = \omega - 1$ *si et seulement si il existe une bipartition (L, R) des cliques périphériques de G telle que $X_0 \not\subset \bigcup_{i \in L} X_i$ et $X_0 \not\subset \bigcup_{i \in R} X_i$.*

Démonstration. Nous commençons par prouver la condition suffisante. Dans ce cas, il existe un sommet v_1 de X_0 qui n'apparait dans aucune des cliques périphériques de L. De même, il existe un sommet v_2 de X_0 qui n'apparait dans aucune des cliques périphériques de R. Nous construisons alors les nœuds $Y_i = X_i \cup X_0 \setminus \{v_1\}$ si $i \in L$ et $Y_i = X_i \cup X_0 \setminus \{v_2\}$ si $i \in R$, ainsi que le nœud $Y_0 = X_0$. Une bonne décomposition en chemin $(\mathcal{P}, \mathcal{Y})$ de G vérifie que si $i \in L$, alors Y_i est « à gauche » de Y_0 dans $\mathcal{P}$ et si $i \in R$, alors Y_i est « à droite » de Y_0 dans $\mathcal{P}$.

Pour la condition nécessaire, nous considérons que $(\mathcal{P}, \mathcal{Y})$ est une décomposition en chemin de G, de largeur $\omega - 1$ et que, pour toute bipartition (L, R) des cliques périphériques de G, nous avons $X_0 \subset \bigcup_{i \in L} X_i$. Le nœud $Y_0 = X_0$ est un nœud de $\mathcal{Y}$. En supprimant Y_0 de $\mathcal{P}$ nous obtenons deux sous-chemins P_1 et P_2, lesquels définissent une bipartition (L, R) des cliques périphériques de G. Soit Y_i le nœud de P_1 voisin de Y_0 dans $\mathcal{P}$. Sans perte de généralité, nous supposons que $X_i \subseteq Y_i$. Par définition nous avons $Y_i = X_i \cup X_0$, et donc $|Y_i| = \omega + 1$. Ce dernier argument révèle une contradiction. □

Il est possible de prouver un résultat équivalent pour la largeur arborescente spéciale, mais avant cela nous avons besoin de quelques résultats supplémentaires concernant les décompositions arborescentes spéciales des graphes scindés. Dans toute décomposition arborescente $(\mathcal{T}, \mathcal{Y})$, spéciale ou non, pour chaque clique maximale X_i de G, il existe un nœud Y_i de la décomposition qui contient X_i. En particulier, il existe un nœud Y_0 contenant X_0.

Lemme 4.16. *Soit Y_r le nœud racine de $(\mathcal{T}, \mathcal{Y})$ et soit $\mathcal{P}$ le chemin reliant Y_r à Y_0 dans $\mathcal{T}$. Tous les nœuds internes de $\mathcal{P}$ sont de degré 2 dans $\mathcal{T}$.*

Démonstration. Dans G, toutes les cliques périphériques ont une intersection non vide avec la clique centrale, en conséquence, tous les nœuds de $\mathcal{T}$ contienent au moins un sommet central. Si un nœud interne du chemin racine avait un degré supérieur à 2, alors il induirait au moins deux sous arbres T_1 et T_2, chacun contenant au moins un sommet central. Sans perte de généralité, nous considérons que T_1 contient Y_0 qui, rapellons le, contient tous les sommets centraux. Et donc, l'intersection de T_1 et T_2 n'est pas vide. Ce dernier état de fait viole la définition de la décomposition arborescente spéciale. □

Le lemme ci-dessus implique que, pour toute décomposition arborescente spéciale $(\mathcal{T},\mathcal{Y})$ de G, il existe un nœud Y_0, contenant X_0, qui « sépare » $\mathcal{T}$ en un chemin $\mathcal{P}$ (reliant la racine de $\mathcal{T}$ à Y_0) et une forêt $\mathcal{F}$. En supprimant Y_0 de $\mathcal{T}$ nous obtenons une partition $(L,R_1,\dots,R_p)$, où L contient les nœuds de $\mathcal{P}\setminus\{Y_0\}$ et où chaque R_i représente un arbre de $\mathcal{F}$. Nous pouvons alors énoncer le lemme suivant.

Lemme 4.17. *Soit G un graphe scindé, alors* $\mathrm{stw}(G)=\omega-1$ *si et seulment si il existe une bipartition (L,R) des cliques périphériques de G telle que $X_0\not\subset\bigcup_{i\in L}X_i$ et $X_0\not\subset\bigcup_{i\in R}X_i$.*

Démonstration. La condition suffisante découle directement du Lemme 4.15.

Nous passons donc à la condition nécessaire. Pour cela nous considérons que $(\mathcal{T},\mathcal{Y})$ est une décomposition arborescente spéciale de G, de largeur $\omega-1$ et que, pour toute bipartition (L,R) des cliques périphériques de G, nous avons $X_0\subset\bigcup_{i\in L}X_i$ ou $X_0\subset\bigcup_{i\in R}X_i$ avec, dans le second cas, l'impossibilité de partitionner R en une partition (R_1,R_2) telle que $X_i\cap X_j=\emptyset$, pour tout couple $(X_i,X_j)\in R_1\times R_2$. En réutilisant les arguments du Lemme 4.15 il est possible de faire apparaître une contradiction dans l'énoncé précédent.

Il nous reste à montrer que si, dans le second cas, il existe une partition (R_1,R_2) telle que $X_i\cap X_j=\emptyset$, pour tout couple $(X_i,X_j)\in R_1\times R_2$, alors il existe une partition (L',R') respectant la condition suffisante. Si nous supposons que $X_0\not\subset\bigcup_{i\in L\cup R_1}X_i$, alors la bipartition $(L\cup R_1,R_2)$ fait l'affaire, sinon il nous suffit de considérer $(L\cup R_2,R_1)$, car $X_0\cap\bigcup_{i\in R_2}X_i$ est un sous ensemble de $X_0\cap\bigcup_{i\in L}X_i$. □

En combinant les Lemmes 4.15 et 4.17, nous obtenons le théorème suivant.

Théorème 4.18. *Soit G un graphe scindé, alors* $\mathrm{stw}(G)=\mathrm{pw}(G)$.

Il est alors possible de calculer la largeur arborescente spéciale d'un graphe scindé en temps $O(n^3)$, où n est le nombre de sommets du graphe ; ce résultat nous est donné par Gustedt [30].

4.2.2. Graphes récalcitrants

Théorème 4.19. *Soit $\mathcal{C}$ une classe de graphes close par ajout d'un sommet universel. Il existe une réduction polynomiale du problème de décision lié à la largeur de chemin au problème de décision lié à la largeur arborescente spéciale pour chacun des graphes de $\mathcal{C}$.*

Démonstration. Pour chaque instance du problème de décision lié à la largeur de chemin sur un graphe G, nous construisons le graphe $G\otimes v$. Il nous suffit ensuite d'invoquer le Lemme 4.9. □

Un graphe est dit *triangulé* s'il ne contient pas de cycle de taille 4 ou plus sans corde.

Lemme 4.20. *La classe des graphes triangulés est stable par ajout d'un sommet universel.*

Démonstration. Soit G un graphe triangulé, et soit $G^*=G\otimes v$. Tous les cycles de G^* de taille 4 ou plus contenant v ont une corde. En effet si C est un cycle de taille $n\geq 4$ de G^* qui contient v, alors il existe un sommet u de C qui n'est pas adjacent à v dans C. Comme v est universel, il y a une arête entre u et v dans G^*, et donc C à une corde. □

Soit G un graphe triangulé, si G possède une décomposition arborescente $(\mathcal{T},\mathcal{X})$ telle que $\mathcal{T}$ est une étoile (un arbre de hauteur au plus 1), alors G est un *graphe étoilé*.

Lemme 4.21. *La classe des graphes étoilés est stable par ajout d'un sommet universel.*

Démonstration. Soit G un graphe étoilé, et soit $G^* = G \otimes v$. D'après le Lemme 4.20, G^* est triangulé. Si $(\mathcal{T},\mathcal{X})$ est une décomposition arborescente de G où $\mathcal{T}$ est une étoile, alors $(\mathcal{T},\mathcal{Y})$ est une décomposition arborescente de G^*, avec $\mathcal{Y} = \{X_i \cup \{v\} : X_i \in \mathcal{X}\}$. □

Un *graphe de comparabilité* est définit par un ensemble de sommets V et une relation d'ordre partielle $\leq$ sur cet ensemble, telle qu'il y a une arête entre deux sommets u et v si et seulement si $u \leq v$ ou $v \leq u$. Un *graphe de co-comparabilité* est le complémentaire d'un graphe de comparabilité.

Lemme 4.22. *La classe des graphes de co-comparabilité est close par ajout d'un sommet universel.*

Démonstration. Ajouter un sommet universel à un graphe de co-comparabilité G revient à ajouter un sommet isolé dans le graphe complémentaire. L'ajout d'un sommet isolé dans un graphe de comparabilité consiste à ajouter un élément qui ne peut être comparé avec aucun autre, créant de ce fait un nouveau graphe de comparabilité, dont le complémentaire n'est autre que G augmenté d'un sommet universel. □

Le Théorème 4.19 nous permettent d'énoncer le théorème suivant.

Théorème 4.23. *Le problème de décision lié à la largeur arborescente spéciale est NP-complet pour les graphes triangulés, les graphes étoilés ainsi que pour les graphes de co-comparabilité.*

Démonstration. La NP-complétude du problème de décision lié à la largeur de chemin pour les graphes triangulés, les graphes étoilés et les graphes de co-comparabilité nous ait donné par [2, 30, 31]. □

Une conséquence directe de ce théorème est que le problème de décision lié à la largeur arborescente spéciale est NP-complet dans le cas général.

Le cas des graphes bipartis

Un *graphe biparti* G est un graphe dont les sommets peuvent être partitionnés en deux ensembles, chacun d'eux induisant un stable de G. De la même manière que pour les graphes scindés, nous utilisons la notation $G = (V_1, V_2, E)$, où V_1 et V_2 représentent les deux ensembles de la bi-partition des sommets de G.

L'ajout d'un sommet universel à un graphe biparti n'est pas possible sans briser la condition de bi-partition. Cependant nous pouvons ajouter deux sommets : le premier dans V_1 en le rendant adjacent à tous les sommets de V_2 et le second dans V_2 en le rendant adjacent à tous les sommets de V_1. Nous ajoutons également une arête entre ces deux nouveaux sommets. Cette arête n'est pas vraiment utile dans le cas des graphes bipartis mais elle l'est pour les graphes bipartis triangulés que nous présentons ci-dessous. C'est donc par souci d'homogénéité que nous utilisons cette arête dans le cas

présent. Nous appelons l'opération que nous venons de décrire l'opération d'ajout d'une *paire universelle*, que nous érivons $G \otimes (u,v)$, avec u et v les deux sommets qui forment la paire universelle.

Un graphe biparti est dit *biparti triangulé* s'il ne contient pas de cycle de taille 6 ou plus sans corde. Pour la même raison que celle discutée précédemment, l'ajout d'un sommet universel n'est pas possible pour les graphes bipartis triangulés. L'astuce précédente fonctionne, à condition de ne pas oublier l'arête entre les deux nouveaux sommets afin de conserver la propriété de cordalité.

Lemme 4.24. *Soit G un graphe biparti, alors* $\mathrm{stw}(G \otimes (u,v)) = \mathrm{pw}(G \otimes (u,v))$. *De plus,* $\mathrm{pw}(G) + 1 \leq \mathrm{pw}(G \otimes (u,v)) \leq \mathrm{pw}(G) + 2$.

Démonstration. Soit (U,V) la bi-partition des sommets de G, telle que, dans $G \otimes (u,v)$, tous les sommets de U sont adjacents à u et tous les sommets de V sont adjacents à v. Soit $(\mathcal{T},\mathcal{X})$ une décomposition arborescente spéciale de $G \otimes (u,v)$ et soient $\mathcal{P}_u$ et $\mathcal{P}_v$ les sous-chemins induits par les nœuds de $\mathcal{T}$ contenant respectivement u et v. Comme l'arête uv existe, l'intersection de $\mathcal{P}_u$ et $\mathcal{P}_v$ n'est pas nulle. Nous notons alors $\mathcal{P}_{uv}$ le sous-chemin induits par les nœuds de $\mathcal{T}$ contenant à la fois u et v.

D'après le Lemme 4.9, les nœuds de $\mathcal{P}_u$ couvrent les sommets de U et les nœuds de $\mathcal{P}_v$ couvrent les sommets de V. Par conséquent, si $\mathcal{P}_u = \mathcal{P}_v = \mathcal{P}_{uv}$, alors nous avons $\mathrm{stw}(G \otimes (u,v)) = \mathrm{pw}(G \otimes (u,v)) = \mathrm{pw}(G) + 2$. Dans le cas où $\mathcal{P}_u$ est un sous-chemin de $\mathcal{P}_v$ nous avons encore $\mathrm{stw}(G \otimes (u,v)) = \mathrm{pw}(G \otimes (u,v))$; le cas où $\mathcal{P}_v$ est un sous-chemin de $\mathcal{P}_u$ est équivalent. Par contre déterminer si $\mathrm{pw}(G \otimes (u,v))$ est égale à $\mathrm{pw}(G) + 1$ ou $\mathrm{pw}(G) + 2$ est moins évident.

Avant de passer au dernier cas de figure, nous nous attardons sur le contenu des nœuds de $\mathcal{P}_v$ qui ne contiennent pas u (quand $\mathcal{P}_u$ est un sous-chemin strict de $\mathcal{P}_v$ bien entendu). Le chemin $\mathcal{P}_v$ peut donc être divisé en trois sous-chemins $\mathcal{P}_v^+$, $\mathcal{P}_{uv}$ et $\mathcal{P}_v^-$, tels que tous les nœuds de $\mathcal{P}_v^+$ sont les ancêtres du « plus ancien » nœud, noté X_{uv}^+, de $\mathcal{P}_{uv}$ et que tous les nœuds de $\mathcal{P}_v^-$ sont les descendants du « plus jeune » nœud, noté X_{uv}^-, de $\mathcal{P}_{uv}$. Soit V^+ le sous-ensemble de V couvert par les nœuds de $\mathcal{P}_v^+$. Le chemin $\mathcal{P}_v^+$ est alors composé de $|V^+|$ nœuds, contenant chacun un unique sommet de V^+ ainsi que les tous les sommets de $X_{uv}^+ \cap U$, sans oublié le sommet v. De même, si V^- est le sous-ensemble de V couvert par les nœuds de $\mathcal{P}_v^-$, alors le chemin $\mathcal{P}_v^-$ est composé de $|V^-|$ nœuds, contenant chacun un unique sommet de V^- ainsi que les tous les sommets de $X_{uv}^- \cap U$, sans oublié, ici encore, le sommet v.

Nous passons maintenant au dernier cas de figure, dans lequel $\mathcal{P}_u$ et $\mathcal{P}_v$ ne sont pas inclus l'un dans l'autre. De la même façon que le paragraphe précédent, nous pouvons séparer $\mathcal{T}$, l'arbre de décomposition spécial de $G \otimes (u,v)$ en quatre sous-chemins : le chemin $\mathcal{P}_{uv}$ regroupant tous les nœuds de $\mathcal{T}$ contenant à la fois u et v, le chemin $\mathcal{P}^+$, dont tous les nœuds sont les ancêtres du « plus ancien » nœud de $\mathcal{P}_{uv}$, le chemin $\mathcal{P}_v^-$ qui regroupe tous les nœuds de $\mathcal{T}$ qui sont les descendants du « plus jeune » nœud, noté X_{uv}^-, de $\mathcal{P}_{uv}$ et qui contiennent v et finalement $\mathcal{P}_u^-$ qui, au même titre que $\mathcal{P}_v^-$, regroupe tous les nœuds de $\mathcal{T}$ qui sont les descendants de X_{uv}^- et qui contiennent u. Dans ce cas de figure, $\mathcal{T}$ n'est clairement pas un chemin. Nous pouvons cependant réaranger les nœuds de $\mathcal{T}$ afin de former un chemin. Rappelons d'abord que chaque nœud de $\mathcal{P}_v^-$

contient tous les sommets de $X_{uv}^- \cap U$ et chaque nœud de $\mathcal{P}_u^-$ contient tous les sommets de $X_{uv}^- \cap V$. Si X_{uv}^- ne fait pas partie des nœuds de taille maximum, alors nous pouvons intercaler les nœuds de $\mathcal{P}_v^-$ entre X_{uv}^- et son père, noté Z, dans $\mathcal{T}$, tout en y ajoutant les sommets de $X_{uv}^- \cap V$ afin de conserver la propriété de connexité. Par contre, si X_{uv}^- est un nœuds de taille maximum, alors nous pouvons supposer qu'il existe un sommet $x \in X_{uv}^-$ qui n'apparait pas dans Z. Par conséquent, si x est un sommet de U, alors nous pouvons intercaler les nœuds de $\mathcal{P}_u^-$ entre X_{uv}^- et Z, tout en y ajoutant les sommets de $X_{uv}^- \cap (U \setminus \{x\})$. À l'inverse, si x est un sommet de V, alors nous pouvons intercaler les nœuds de $\mathcal{P}_v^-$ entre X_{uv}^- et Z, tout en y ajoutant les sommets de $X_{uv}^- \cap (V \setminus \{x\})$.

Dans tous les cas nous obtenons bien une décomposition en chemin de $G \otimes (u,v)$, dont les nœuds maximaux contiennent $\mathrm{pw}(G)+1$ ou $\mathrm{pw}(G)+2$ sommets. □

En donnant un algorithme polynomial permettant d'identifier les graphes bipartis pour lesquels $\mathrm{pw}(G)+1 = \mathrm{pw}(G \otimes (u,v)) = \mathrm{stw}(G \otimes (u,v))$, nous montrerions, à l'aide du Lemme 4.25 ci-dessous, que le problème de décision lié à la largeur arborescente spéciale est NP-complet pour les graphes bipartis et bipartis triangulés.

Lemme 4.25. *Les classes des graphes bipartis et bipartis triangulés sont closes par ajout d'une paire universelle.*

Démonstration. La conservation de la bipartition est triviale.

Soit G un graphe biparti triangulé et soient u et v les deux sommets qui forment la paire universelle que nous voulons ajouter. Par définition, tout cycle C de taille au moins 6 contenant uniquement u possède au moins trois sommets qui ne sont pas dans la même partition que u. Il y a donc au moins un sommet qui n'est pas voisin de u dans C mais qui l'est dans $G \otimes (u,v)$, faisant ainsi aparaître une corde de C. Le même raisonement est utilisé pour les cycles de taille au moins 6 contenant uniquement v. Nous nous plaçons maintenent dans le cas où C contient à la fois u et v. Si u et v sont voisins dans C, nous réutilisons l'argument précédent. Sinon, u et v ne sont pas voisins dans C et, par conséquent, l'arête entre ces deux sommets est une corde de C. □

4.2.3. Un résultat polynomial intéressant

Pour toutes les classes de graphes que nous avons étudiées jusqu'à présent, le problème de décision lié à la largeur arborescente spéciale a une complexité similaire au problème de décision lié à la largeur de chemin. Il est donc légitime de se demander si c'est toujours le cas. Nous apportons une réponse négative à cette interrogation, en montrant qu'il existe une classe de graphes pour laquelle la résolution du problème de décision lié à la largeur arborescente spéciale se fait en temps polynomial alors que le problème de décision lié à la largeur de chemin est NP-complet.

Un graphe est un *graphe de chemins enracinés* s'il correspond au graphe d'intersection d'un ensemble de sous-chemins d'un arbre enraciné (et par définition orienté de la racine vers les feuilles). Savoir si un graphe appartient à la classe des graphes de chemins enracinés se décide en temps linéaire [25]. Les graphes de chemins enracinés sont à la largeur arborescente spéciale ce que les graphes triangulés sont à la largeur arborescente

« classique », tout comme les graphes d'intervalles avec la largeur de chemin. En d'autres termes, si nous remplissons tous les nœuds d'un arbre de décomposition spéciale avec toutes les arêtes possibles, alors nous obtenons un graphe de chemins enracinés. Cette observation implique que la classe des graphes de chemins enracinés est une sous-classe des graphes triangulés, par conséquent la largeur arborescente spéciale d'un tel graphe est égale à sa largeur arborescente « classique » qui, elle même, est égale à la taille d'une clique maximale du graphe moins 1. La résolution du problème de décision lié à la largeur arborescente spéciale se fait donc en temps polynomial si nous considérons un graphe de chemins enracinés, ce qui n'est pas le cas du problème de décision lié à la largeur de chemin, come nous l'indique le théorème suivant.

Théorème 4.26. *Le problème de décision lié à la largeur de chemin est NP-complet pour les graphes de chemins enracinés.*

Ce théorème est une conséquence directe du résultat présenté par Suchan et Todinca dans [47] qui dit que le problème de décision lié à la largeur de chemin est NP-complet pour les graphes « pieuvres », lesquels forment une sous-classe des graphes de chemins enracinés.

4.2.4. Graphes de « petite » largeur arborescente

Grâce au Théorème 2.17, nous savons que les problèmes de reconnaissance liés à la largeur arborescente « classique » ainsi qu'à la largeur de chemin sont des problèmes de complexité parametrée linéaire. Récement, Bodlaender, Kratsch et Kreuzen ont adapté l'algorithme de ce théorème dans le cadre de la largeur arborescente spéciale.

Théorème 4.27 (Bodlaender *et al.* [7])**.** *Pour tout graphe, il existe un algorithme constructif linéaire résolvant le problème de reconnaissance lié à la largeur arborescente spéciale.*

Nous rappelons que la largeur arborescente spéciale des graphes de largeur arborescente majorée n'est pas nécessairement majorée (Lemme 4.10). Aussi il peut s'avérer utile de déterminer quelles sont les classes de graphes pour lesquels il est possible de majorer la largeur arborescente spéciale.

Un *graphe de Halin* est un graphe que l'on obtient à partir d'un plongement planaire d'un arbre sans sommet de degré 2, en connectant ses feuilles suivant l'ordre cyclique définit par le plongement. Nous considérons aussi un *arbre planaire*, qui est un arbre enraciné avec un plongement planaire ; cet arbre planaire définit l'arbre sous-jacent d'un graphe de Halin. La *feuille gauche* d'un arbre planaire est la première feuille à être visitée lors d'un parcours en profondeur, la *feuille droite* est celle qui est visitée en dernier lors de ce même parcours.

Théorème 4.28. *La largeur arborescente spéciale des graphes de Halin est au plus* 5.

Démonstration. Soit H un graphe de Halin et soit T l'arbre planaire sous-jacent. Soit uv une arête de T telle que ni u ni v ne soit une feuille de T (les seuls graphes de Halin qui n'admettent pas une telle arête sont ceux dont l'arbre sous-jacent est une étoile, et pour

FIGURE 4.2. : Une 3-arêtes séparation d'un graphe de Halin (a) et une 2-arêtes séparation d'un graphe de Halin ouvert (b).

lesquels nous vérifions assez facilement que la largeur arborescente spéciale est égale à 3). Maintenant, nous considérons le sous-arbre planaire de T, enraciné en u et qui ne contient pas v ; nous l'appelons T_u et de la même manière, T_v est le sous-arbre planaire de T enraciné en v et qui ne contient pas u. Soient u_l et v_l les feuilles gauches de T_u et T_v et soient u_r et v_r leur feuilles droites. Il en découle que $u_l v_r$ et $u_r v_l$ sont des arêtes de H et, avec uv, ces trois arêtes séparent H en deux sous-graphes disjoints. Nous appelons H_u le sous-graphe induit par $V(T_u)$ et H_v celui induit par $V(T_v)$. On dit alors que H_u et H_v sont des *graphes de Halin ouverts*. La FIGURE 4.2a illustre les notions introduite jusqu'ici.

Nous pouvons alors commencer la décomposition arborescente spéciale de H. Pour ce faire, nous construisons $\mathcal{D}$ avec des nœuds de taille au plus 6. La racine de $\mathcal{D}$ contient les sommets u, u_l, u_r, v, v_l et v_r et a deux fils, lesquels sont les racines des décompositions arborescentes spéciales de H_u et H_v.

Il nous reste à montrer comment construire une décomposition arborescente spéciale de largeur au plus 5 pour un graphe de Halin ouvert. À partir de maintenant, nous travaillons avec un graphe de Halin ouvert G enraciné en un sommet appelé u, et avec l et r comme feuilles gauche et droite. Ces trois sommets doivent être présents dans le nœud racine ρ de la décomposition arborescente spéciale de G.

Premièrement, si u n'a qu'un seul voisin dans G, que nous appelons v, alors ρ est le nœud contenant les quatre sommets u, v, l et r. Après quoi nous continuons notre décomposition avec le sous-graphe de Halin ouvert de G enraciné en v, la racine de cette décomposition sera le fils de ρ.

Nous supposons à présent que u a au moins deux fils, et nous appelons v celui qui est le plus à gauche. Si v est une feuille de l'arbre sous-jacent T de G (c'est à dire $v = l$), alors $X_\rho = \{u, v, l', r\}$, avec l' la feuille de T juste à la droite de v, toujours en considérant l'ordre cyclique du plongement. Ensuite nous continuons la décomposition avec le sous-graphe de Halin ouvert de G enraciné en u avec l' et r comme feuilles gauche et droite.

Considérons alors que v n'est pas une feuille de T. Soit G_v le sous-graphe de Halin ouvert de G enraciné en v et soit G_u le sous-graphe de Halin ouvert de G enraciné en u et qui ne contient pas v. Nous appelons v_r la feuille droite de G_v et u_l la feuille gauche de G_u. Les arêtes uv et $u_l v_r$ forment une 2-arêtes séparation de G comme montré

en FIGURE 4.2b. Finalement, le nœud ρ contient les sommets u, u_l, v, v_r, l et r, puis nous traitons séparement les sous-graphes de Halin ouverts G_u et G_v ; les racines de sous-décompositions correspondantes sont rattachées en temps que fils de ρ. □

Les graphes de Halin admettent le graphe K_4 comme mineur, de ce fait leur largeur arborescente (spéciale ou non) est au moins 3. Par conséquent la largeur arborescente spéciale des graphes de Halin est comprise entre 3 et 5.

4.3. Largeur arborescente spéciale et mineurs de graphes

Dans son introduction de la largeur arborescente spéciale, Courcelle [18] a montré que les classes de graphes de largeur arborescente au plus k étaient closes par minoration topologiques, pour tout k. Il a aussi montré que ces classes n'étaient pas close par minoration pour $k \geq 5$. Dans cette section nous renforçons ce résultat en montrant que les classes de graphes de largeur arborescente au plus k, pour $k \geq 3$, ne sont pas closes par minoration. Nous montrons également que la classe de graphes de largeur arborescente spéciale au plus 2 est close par minoration et nous donnons une caractérisation complète des mineurs interdits. Ces deux résultats ont été indépendammant étudiés par Bodlaender, Kratsch et Kreuzen [7].

Théorème 4.29. *La classe des graphes de largeur arborescente spéciale au plus k n'est pas close par minoration pour $k \geq 3$.*

Démonstration. Pour montrer ce résultat il nous suffit de regarder le contre-exemple suivant. Considérons tout d'abord le cycle de taille 4, composé des arêtes e_1, e_2, e_3 et e_4. Puis, pour chaque arête e_i avec $1 \leq i \leq 3$, nous construisons un graphe complet à $k-1$ sommets et nous connectons tous ses sommets aux extrémités de e_i. Le graphe ainsi obtenu est de largeur arborescente spéciale k. Or, si nous contractons l'arête e_4, nous obtenons un graphe dont la largeur arborescente spéciale est $k+1$. □

Il nous reste à considérer deux classes de graphes qui sont celles contenant les graphes de largeur arborescente spéciale au plus 1 et au plus 2. La classe de graphes de largeur arborescente spéciale au plus 1 est celle qui contient toutes les forêts et est, bien entendue, close par minoration. La dernière classe qu'il nous faut étudier est elle aussi close par minoration, comme le démontre le théorème suivant.

Théorème 4.30. *Un graphe G est de largeur arborescente spéciale au plus 2 si et seulement si G n'admet aucun des mineurs de $\mathcal{F} = \{K_4, S_3, \theta_2, F_1, F_2, F_3\}$, illustrés* FIGURE *4.3.*

Démonstration. Ce résultat découle directement du Lemme 4.31 pour la condition nécessaire, et des Lemmes 4.32 et 4.33 pour la condition suffisante. □

Le prochain lemme donne la preuve de la condition nécessaire du Théorème 4.30. Dans la preuve de ce lemme nous aurons besoin de *diviser* certains sommets, ce qui est équivalent à la démarche suivante. Pour diviser un sommet x dans un graphe G, nous commençons par le supprimer de G, après quoi nous ajoutons deux nouveaux sommets

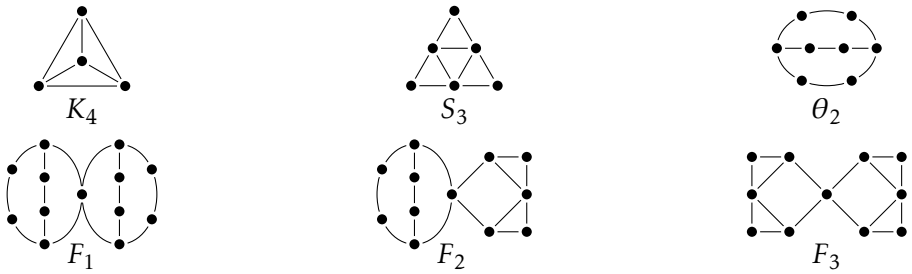


FIGURE 4.3. : Les six mineurs à exclure pour obtenir un graphe de largeur arborescente spéciale au plus 2.

u et v, reliés par une arête, enfin, pour chaque sommet y qui était un voisin de x dans G, nous créons une adjacence entre y et soit u soit v.

Lemme 4.31. *Si un graphe G admet un graphe de $\mathcal{F}$ comme mineur, alors* $\mathrm{stw}(G) \geq 3$.

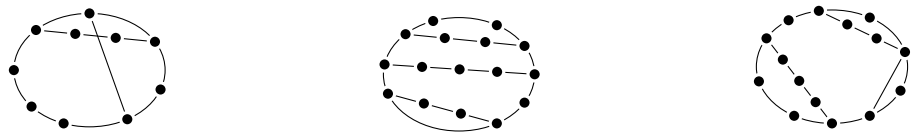
Démonstration. Commençons tout d'abord par rappeler que la classe des graphes de largeur arborescente spéciale au plus k est close par minoration topologique [18], ce qui signifie que si H est un mineur topologique de G, alors $\mathrm{stw}(H) \leq \mathrm{stw}(G)$. De plus, d'après la Proposition 1.7.4 de [24], si H est un graphe de degré maximum 3 et qu'il est un mineur de G, alors H est aussi un mineur topologique de G. En conséquence, la largeur arborescente spéciale de G est au moins 3 si G admet K_4 ou θ_2 comme mineur, car ces deux graphes ont une largeur arborescente spéciale égale à 3.

Pour traiter le cas de S_3, nous ne pouvons pas appliquer directement le raisonement précédent, car S_3 contient des sommets de degré 4. Cependant, nous pouvons considérer l'ensemble $\mathcal{M}$ des mineurs topologiques minimaux qui admettent S_3 comme mineur. Nous obtenons ces graphes en divisant les sommets de degré 4 de S_3 en sommets de degré 3 (voir l'Annexe A pour plus de détails). L'ensemble $\mathcal{M}$ contient alors des graphes de largeur arborescente spéciale au moins 3. Par conséquent, si G admet S_3 comme mineur, alors sa largeur arborescente spéciale est au moins 3, ceci est dû au fait que G admet un mineur S_3 si et seulement si il admet un des graphes de $\mathcal{M}$ comme mineur topologique.

Nous utilisons la même procédure de division de sommets pour les graphes F_1, F_2 et F_3 (le détail se trouve également en Annexe A). □

La preuve de la condition suffisante du Théorème 4.30 est divisée en deux parties. Premièrement, nous discutons du cas des graphes bi-connexes, puis nous étendons ce résultat aux graphes qui ont des sommets d'articulations. Mais, juste avant de commencer, nous nous devons d'introduire les notions suivantes.

Un *raccourci* dans un cycle C est soit une corde e de C, soit une chaîne obtenu en subdivisant e au moins une fois. Deux raccourcis P_1 et P_2 dans un cycle C, joignant respectivement x_1 à y_1 et x_2 à y_2, sont dit *croisés* si x_1, x_2, y_1 et y_2 sont visités dans cet ordre lors d'une marche le long de C (ces quatre sommets doivent tous être différents dans C). Si nous ajoutons un troisième raccourci P_3, joignant x_3 à y_3, alors les raccourcis



(a) (b) (c)

FIGURE 4.4. : Cycles avec deux racourcis croisés (a), trois racourcis parallèles (b) et trois racourcis qui ne sont ni croisés ni parallèles (c).

P_1, P_2 et P_3 sont *parallèles* si les sommets x_1, x_2, x_3, y_3, y_2 et y_1 apparaissent dans cet ordre durant la marche précédemment définie (x_1, x_2 et x_3 peuvent représenter un à trois sommets différents de C, le même comportement est attendu de la part des sommets y_1, y_2 et y_3). La FIGURE 4.4 illustre les trois differentes configurations de raccourcis.

Soit G un graphe formé d'un cycle C et de k raccourcis parallèles $P_1, \ldots, P_k$; les extrémités de chaque P_i sont indexées x_i et y_i. Nous définissons alors deux ensembles qui identifient les *bons* et les *mauvais* sommets. Soit P_0 la sous-chaîne de C qui relie x_1 à y_1 en évitant tous les $x_i \neq x_1$ et $y_i \neq y_1$, pour $2 \leq i \leq k$, et soit P_{k+1} la sous-chaîne de C qui relie x_k à y_k en évitant tous les $x_i \neq x_k$ et $y_i \neq y_k$, pour $1 \leq i \leq k-1$. Les deux chaînes P_0 et P_{k+1} couvrent l'ensemble des bons sommets (x_1, y_1, x_k et y_k inclus). Les autres sommets, qui sont les sommets internes des raccourcis et les sommets « pris en sandwich » entre deux raccourcis, sont des mauvais sommets.

Lemme 4.32. *Un graphe bi-connexe et sans mineur K_4, S_3 ou θ_2 a une largeur arborescente spéciale au plus 2.*

Démonstration. Cette preuve est par contradiction. Soit G un contre exemple, alors G n'admet ni K_4 ni S_3 ni θ_2 comme mineur et est de largeur arborescente spéciale au moins 3. Comme nous ne considérons que des graphes bi-connexes, G contient au moins un cycle que nous appelons C. Tout d'abord, C ne peut pas avoir deux raccourcis croisés car G n'admet pas de mineur K_4. Et un mineur S_3 apparaît si C contient trois raccourcis qui ne sont pas parallèles. Par conséquent, G est un cycle avec possiblement des raccourcis qui doivent être parallèles. Soit x et y deux sommets de C et soient P_1, P_2 et P_3 trois chemins disjoints (qui n'empreintent pas nécessairement des racourcis) de longueur au moins 3 et reliant x à y. Cette configuration implique l'existance d'un mineur θ_2 de G. C'est pour cela que les racourcis autorisés sont de longueur 1 ou 2. En d'autres termes, G est obtenu depuis un cycle C avec des cordes parallèles dont certaines peuvent être subdivisées une fois au plus (G peut aussi être un cycle sans cordes).

Dans la dernière partie de cette preuve, nous présentons un algorithme qui calcule une décomposition arborescente spéciale de largeur 2 pour tout cycle avec des racourcis parallèles de longeur au plus 2. Soit G un tel graphe avec C le cycle de base et $P_1, \ldots, P_k$ les racourcis ; les extrémités de chaque P_i sont toujours indexées x_i et y_i. De plus nous choisissons un bon sommet de C et nous l'appelons x_0. L'algorithme est le suivant. Premièrement nous construisons une décomposition arborescente spéciale (qui est en fait une décomposition en chemin) pour le chemin reliant x_0 à x_1 et dans laquelle nous gardons la trace de x_0 ; la largeur de cette décomposition est clairement 2, au plus.

Puis nous continuons avec le chemin qui relie x_0 à y_1 et, cette fois-ci, c'est x_1 que nous gardons en mémoire. Ensuite nous construisons le racourcis reliant x_1 à y_1 ; nous devons retenir à la fois x_1 et y_1 mais, comme la taille des racourcis est au plus 2, le nœud qui en résulte est de taille au plus 3 (et donc la largeur de la décomposition est au plus 2). À l'étape suivante, nous décomposons le chemin reliant x_1 à x_2 sans oublier y_1 puis nous décomposons le chemin entre y_1 et y_2 en gardant un œil sur x_2 afin de traiter le raccourci reliant x_2 et y_2. Nous réitérons cette procédure jusqu'à avoir traité P_k. L'étape finale de notre algorithme construit le chemin de C qui relie x_k à y_k.

Cet algorithme révèle une contradiction en ce qui concerne l'existance d'un contre exemple et conclut ainsi la preuve de ce lemme. □

Afin d'étendre le résultat du lemme précédent aux graphes connexes nous étudions plus attentivement leur structure. Premièrement, leurs blocs (ou composantes bi-connexes) doivent exclure les mineurs K_4, S_3 et θ_2, ce sont donc des cycles qui peuvent avoir des raccourcis parallèles de longueur au plus 2. Les différents blocs de ces graphes peuvent être connectés les uns aux autres à l'aide de structures arborescentes en respectant le schéma suivant.

Soit T un sous-arbre de taille maximale de G tel que tous les sommets de T sont soit des sommets d'articulations soit des sommets de degré 1, alors T est un *arbre de branchement* de G. Un bloc B qui partage un sommet v avec T est dit *incident* à T. De plus v est le *sommet de branchement* de B et T, aussi noté $\mathrm{br}(B,T)$. L'ensemble $N(T)$ des *voisins* de T est l'ensemble des blocs de G incidents à T.

Maintenant nous définissons une relation d'ordre sur les blocs de G. Soit T un arbre de branchement de G et soit B_1 un bloc de G incident à T tel que $\mathrm{br}(B_1,T)$ est un mauvais sommet de B_1. Alors B_1 précède tous les blocs incidents à T, et nous notons $B_1 \preccurlyeq B_i$, pour chaque $B_i \in N(T)$. Nous considerons maintenant T', un autre arbre de branchement de G, ainsi que B_2 et B_3, deux blocs de G tels que B_2 est incident à la fois à T et T' et B_3 est incident à T' seulement. Si B_1 précède B_2, alors B_2 est un prédécesseur de B_3 et, par transitivité, B_1 précède B_3.

Lemme 4.33. *Soit B_1 et B_2 deux blocs différents de G. Si $B_1 \preccurlyeq B_2$ et $B_2 \preccurlyeq B_1$, alors G admet F_1, F_2 ou F_3 comme mineur. Sinon la largeur arborescente spéciale de G est au plus 2.*

Démonstration. Premièrement, si $B_1 \preccurlyeq B_2$, alors il existe un bloc B_1' et un arbre de branchement T_1 de G tels que $B_1' \preccurlyeq B_1$ et $\mathrm{br}(B_1',T_1)$ est un mauvais sommets de B_1'. De la même façon, il existe un bloc B_2' et un arbre de branchement T_2 de G tels que $B_2' \preccurlyeq B_2$ et $\mathrm{br}(B_2',T_2)$ est un mauvais sommets de B_2'. Dans ce cas de figure, il existe un sous-chemin P de G qui relie $\mathrm{br}(B_1',T_1)$ à $\mathrm{br}(B_2',T_2)$ et tel que $V(B_1') \cap V(P) = \mathrm{br}(B_1',T_1)$ et $V(B_2') \cap V(P) = \mathrm{br}(B_2',T_2)$. Et, par conséquent, le sous-graphe de G induit par les sommets $V(B_1') \cup V(B_2') \cup V(P)$ admet un des graphes F_1, F_2 ou F_3 comme mineur. Le mineur F_1 apparait si P relie un sommet interne d'un raccourci de B_1' à un sommet interne d'un raccourci de B_2'. Pour avoir un mineur F_2, P doit relier un sommet interne d'un raccourci de B_1' à un sommet « pris en sandwich » entre deux raccourcis de B_2'. Enfin, si P relie un sommet « pris en sandwich » entre deux raccourcis de B_1' à un sommet « pris en sandwich » entre deux raccourcis de B_2' alors c'est le mineur F_3 qui apparait.

Nous nous plaçons à présent dans le cas où il n'existe pas deux blocs B_1 et B_2 différents tels que $B_1 \preccurlyeq B_2$ et $B_2 \preccurlyeq B_1$. Il y a alors deux possibilités : soit il existe au moins un sommet de branchement de G qui est mauvais, soit tous les sommets de branchements de G sont bons. Dans le premier cas, les blocs de G définissent un ordre partiel qui peut être représenté par un arbre enraciné, lequel nous servira de guide dans l'algorithme présenté ci-dessous. Dans le second cas, il n'y a absolument aucune relation entre les blocs de G. Cependant nous pouvons choisir arbitrairement un bloc de G comme racine et ainsi définir un arbre avec lequel travailler.

Les blocs de G sont donc maintenant ordonnés dans un arbre enraciné en un bloc B. Pour construire une décomposition spéciale de G, nous construisons en premier lieu la décomposition spéciale $\mathcal{D}$ de B en nous référant à l'algorithme décrit dans le Lemme 4.32. Soit v un sommet de branchement de B, soit X_v le « plus jeune » nœud de $\mathcal{D}$ contenant v et soit T_v l'arbre de branchement de G tel que $\mathrm{br}(B, T_v) = v$. Nous construisons alors une décomposition arborescente spéciale pour T_v que nous attachons à $\mathcal{D}$ en temps que déscendance de X_v. Soit u un sommet de branchement de T_v et soit B_u un bloc de G tel que $\mathrm{br}(B_u, T_v) = u$. Toujours grâce à l'algorithme du Lemme 4.32, nous construisons la décomposition arborescente spéciale de B_u, avec u dans le rôle du bon sommet de départ (le premier paragraphe de la preuve nous assure que u est un bon sommet). Nous greffons alors cette décomposition arborescente spéciale à $\mathcal{D}$ en la plaçant en temps que descendante de X_u, qui est le « plus jeune » nœud de $\mathcal{D}$ contenant u. L'algorithme poursuit ainsi le traitement de tous les sommets de branchements de G jusqu'à produire une décomposition arborescente spéciale de largeur au plus 2. □

4.4. Récapitulatif

Pour conclure ce chapitre nous dressons un tableau qui résume les différentes classes de complexité pour le calcul du problème de décision lié à la largeur arborescente spéciale appliqué à un certain nombre de classes de graphes. Ce tableau met aussi à disposition un comparatif avec le problème de décision lié à la largeur arborescente, le problème de décision lié à la largeur de chemin et le problème de décision lié à la largeur de clique.

Dans la Table 4.1, la notation B indique que la classe de graphes est de largeur majorée et qu'il existe un algorithme linéaire pour calculer la largeur en question. La notation P est utilisée dans le cas où la largeur est calculable en temps polynomial. La notation NPC signifie que le problème est NP-complet. Enfin, les points d'interrogations désignent des problèmes ouverts. Les valeurs en gras sont les résultats qui ont été ajoutés ou qui diffèrent du tableau original établi par Bodlaender [3]. Les notations TWP, STWP, PWP et CWP, dénotent les problèmes de décisions respectivement liés à la largeur arborescente « classique », à la largeur arborescente spéciale, à la largeur de chemin et à la largeur de clique.

Les résultats polynomiaux pour la largeur de chemin des classes de graphes dont la largeur arborescente est majorée sont dus aux algorithmes présentés dans [4] et [5]. Toujours en ce qui concerne les graphes de largeur arborescente majorée, l'algorithme

Classes de graphes	Classe de complexité pour			
	TWP	STWP	PWP	CWP
Graphes de degré borné	NPC	?	NPC	?
Forêts	B	**B**	P	**B**
Graphes séries-paralleles	B	?	**P**	**B**
Graphes planaires extérieurs	B	?	**P**	**B**
Graphes de Halin	B	**B**	**P**	**B**
Graphes planaires k-extérieurs	B	?	**P**	**B**
Graphes planaires	?	?	NPC	?
Graphes triangulés	P	**NPC**	NPC	?
Graphes étoilés	P	**NPC**	NPC	?
Graphes k-étoilés	P	?	P	?
Graphes co-triangulés	P	?	P	?
Graphes scindés	P	**P**	P	?
Graphes bipartis	NPC	?	NPC	?
Graphes de permutations	P	**P**	P	?
Graphes de permutations circulaires	P	?	?	?
Graphes de co-comparabilité	NPC	**NPC**	NPC	?
Cographes	P	**P**	P	**B**
Graphes bipartis triangulés	P	?	NPC	?
Graphes d'intervalles	P	**P**	P	?
Graphes d'intervalles circulaires	P	?	**P**	?
Graphes de cordes	P	?	**NPC**	?
Graphes de chemins enracinés	**P**	**P**	**NPC**	?

TABLE 4.1. : Classes de complexité des problèmes de décision liés à différentes largeurs de graphes en fonction de la classe de graphes considérée.

linéaire permettant de calculer la largeur de clique est donné dans [27]. Les autres résultats prenant en compte la largeur de clique sont présentés dans [23]. Pour finir, les résultats statuant de la complexité du problème de décision lié à la largeur de chemin pour les graphes d'intervalles circulaires, les graphes de cordes ou les graphes de chemins enracinés sont respectivement étudiés dans [47], [35] et [39].

CHAPITRE

5

MÉTA-ALGORITHME POUR LARGEURS D'ARBORESCENCES

Un algorithme pour les gouverner toutes, un algorithme pour les trouver,
Un algorithme pour les amener toutes et dans les mathématiques les lier.

Sommaire

5.1. Définitions **61**
5.1.1. Largeurs de graphes 61
5.1.2. Fonction de partition et arbre de partitions 62
5.1.3. Largeurs de graphes et arbres de partitions 64
5.2. Problème(s) de reconnaissance **67**
5.2.1. Le théorème et ses corollaires 67
5.2.2. Un algorithme naïf 73
5.3. Outils de compression **78**
5.3.1. Chemins étiquetés 78
5.3.2. Caractéristique 82
5.3.3. Une relation d'ordre sur les arbres de partitions 85
5.4. Algorithme utilisant la compression **86**
5.4.1. Les nœuds de départ 86
5.4.2. Les nœuds d'oubli 86
5.4.3. Les nœuds d'introduction 93
5.4.4. Les nœuds de jonction 106
5.4.5. Retour sur les largeurs de graphes 119

De nos jours, il existe un certain nombre de largeurs de graphes, toutes différentes, avec cependant des comportements relativement proches. En effet, pour chacune des largeurs que sont la largeur arborescente et la largeur de chemin [5], la largeur arborescente spéciale [7], la largeur de branches [9], la largeur linéaire [10], la largeur de découpe et la largeur de coupe [49], il existe un algorithme linéaire à paramètre fixé qui permet de résoudre les problèmes de reconnaissance liés à ces différentes largeurs. Le méta-algorithme que nous décrivons dans ce chapitre va nous permettre de regrouper les algorithmes précédents en un seul.

Pour cela nous exprimons les largeurs concernées à l'aide de « fonctions de partition » et « d'arbres de partitions ». Ces deux nouvelles notions sont plus difficiles à appréhender que les notions de largeurs qu'elles généralisent car elles nécessitent un niveau d'abstraction supplémentaire. De plus les fonctions de partition ont un champ d'action qui s'étend, par-delà les graphes, jusqu'aux ensembles. Un graphe est alors vu soit comme un ensemble de sommets avec une relation d'adjacence, soit comme un ensemble d'arêtes avec une relation d'incidence.

Bien que le principe du méta-algorithme que nous décrivons soit assez simple, la preuve de son exactitude est difficile, notamment à cause du niveau d'abstraction des notions qu'elle manipule, mais aussi car elle nécessite une étude de cas relativement conséquente. La grande majorité de ce chapitre est consacrée à la preuve de ce méta-algorithme ainsi qu'à l'étude d'un outil de compression nous permettant de réduire l'espace de recherche, afin d'obtenir un temps d'exécution linéaire en fonction de la taille de l'ensemble (ou du graphe) passé en entrée de l'algorithme.

Ce chapitre comporte donc une première section de définitions dans laquelle nous présentons les différentes largeurs que nous souhaitons généraliser ainsi que les notions de fonctions de partition et d'arbres de partitions. Cette section contient également les fonctions de partition associées à chacune des largeurs introduites.

La Section 5.2 contient l'énoncé du théorème principal, lequel nécessite au préalable l'introduction d'une notion de jolie décomposition d'un ensemble ainsi que la mise en avant de certaines propriétés que les fonctions de partition précédemment définies doivent respecter. Puis nous présentons comment ce théorème s'applique aux largeurs qu'il prend en compte. Enfin nous proposons un premier algorithme « naïf », au sens ou il parcours la totalité de l'espace de recherche, lequel est bien trop vaste pour prétendre à une complexité linéaire.

Dans la section suivante, nous présentons l'outil de compression que nous utilisons afin de réduire l'espace de recherche. La Section 5.3 introduit également un certain nombre de propriétés élémentaires de l'outil qu'elle introduit. Cet outil de compression est une adaptation de celui qui a été introduit par Bodlaender et Kloks [5] afin de résoudre le problème dans le cas de la largeur arborescente.

Finalement, la Section 5.4 combine l'algorithme naïf de la Section 5.2 avec la compression de l'espace de recherche définie dans la Section 5.3. C'est aussi dans cette section que nous verifions que la compression que nous avons définie réduit suffisamment l'espace de recherche en gardant néanmoins une quantité d'information suffisante. C'est ici que se concentre toute la difficulté de la preuve.

5.1. Définitions

5.1.1. Largeurs de graphes

Avant de nous attaquer aux fonctions de partitions et aux arbres associés, nous (ré)-introduisons les largeurs de graphes qui sont exprimable grâce à ces fonctions et arbres de partitions.

La première de ces largeur est la largeur arborescente q-ramifiée qui est une généralisation des largeurs arborescentes et de chemins.

Définition 5.1 (Arbre q-ramifié)**.** Soit T un arbre enraciné, nous disons alors que T est *q-ramifié* si, pour tout chemin P reliant une feuille de T à sa racine, P contient au plus q sommets qui ont au moins deux fils dans T.

Définition 5.2 (Largeur arborescente q-ramifiée)**.** Une *décomposition arborescente q-ramifiée* est une décomposition arborescente enracinée dont l'arbre de décomposition est q-ramifié.

La largeur arborescente et la largeur de chemin sont aux deux « extrémités » de la largeur arborescente q-ramifiée. En effet, la largeur arborescente est équivalente à la largeur arborescente ∞-ramifiée, et la largeur de chemin est équivalente à la largeur arborescente 0-ramifiée.

Nous retouvons également la largeur arborescente spéciale que nous avons étudiée dans le Chapitre 4.

Viennent ensuite les largeurs de branches et de découpes. La largeur de branche, qui est une mesure de graphe proche de la largeur arborescente, a été introduite par Robertson et Seymour [42], au cours de leur étude des mineurs de graphes. C'est un peu plus tard que Seymour et Thomas ont défini la largeur de découpe [46], laquelle à une définition très similaire à celle de la largeur de branches. En effet ces deux largeurs font appel à la notion d'arbre de découpe d'un ensemble et, quand l'une « découpe les arêtes en suivant les sommets », l'autre « découpe les sommets en suivant les arêtes ». Notons que les problèmes de décisions liés à ces deux largeurs sont NP-complets [46].

Définition 5.3 (Arbre de découpe)**.** Soit S un ensemble. Nous appelons *arbre de découpe* de S, un arbre dont les sommets internes sont de degré 3 et dont les feuilles sont en bijection avec les éléments de S.

Soit T un arbre de découpe d'un ensemble S. Pour toute arête e de T, nous considérons la bipartition $\{S_1, S_2\}$ de S définie par les deux composantes connexes de $T \setminus e$.

Définition 5.4 (Largeur de branches)**.** Soit $G = (V, E)$ un graphe. Une *décomposition en branches* de G est un arbre de découpe T de E. Pour toute arête e de T nous définissons la frontière de e comme étant l'ensemble des sommets de V incidents à la fois à (au moins) une arête de E_1 et (au moins) une arête de E_2. La *largeur* de T est égale à la taille de la plus grande frontière parmi les arêtes de T. La *largeur de branches* de G est la plus petite largeur parmi toutes les décompositions en branches possibles de G.

Définition 5.5 (Largeur de découpe)**.** Soit $G = (V, E)$ un graphe et soit T arbre de découpe de V. Pour toute arête e de T nous définissons la frontière de e comme étant l'ensemble des arêtes de E reliant les sommets de V_1 aux sommets de V_2. La *largeur* de T est égale à la taille de la plus grande frontière parmi les arêtes de T. La *largeur de découpe* de G est la plus petite largeur parmi toutes les découpes possibles de G.

Les deux largeurs suivantes sont la largeur linéaire et la largeur de coupe. Tout comme les deux largeurs précédentes, la largeur linéaire et la largeur de coupe sont similaires dans leurs définitions : la première « ordonne les arêtes en fonctions des sommets » et l'autre « ordonne les sommets en fonction des arêtes ». Ici encore, les problèmes de décisions liés à ces deux largeurs sont NP-complets [40, 48].

Définition 5.6 (Largeur linéaire)**.** Soit $G = (V, E)$ un graphe. La *largeur linéaire* de G est le plus petit entier k tel que E peut être ordonné suivant un ordre linéaire $(e_1, \ldots, e_m)$ respectant la propriété suivante : pour tout $1 \leq i \leq m-1$, il existe au plus k sommets à la fois incidents à (au moins) une arête de $\{e_1, \ldots, e_i\}$ et à (au moins) une arête de $\{e_{i+1}, \ldots, e_m\}$.

Définition 5.7 (Largeur de coupe)**.** Soit $G = (V, E)$ un graphe. La *largeur de coupe* de G est le plus petit entier k tel que V peut être ordonné suivant un ordre linéaire $(v_1, \ldots, v_n)$ respectant la propriété suivante : pour tout $1 \leq i \leq n-1$, il existe au plus k arêtes à la fois incidentes à un sommet de $\{v_1, \ldots, v_i\}$ et à un sommet de $\{v_{i+1}, \ldots, v_n\}$.

Il existe aussi un rapprochement entre la largeur linéaire et la largeur en branches ainsi qu'entre la largeur de découpe et la largeur de coupe. En effet les décompositions linéaire et de coupe peuvent être respectivement vus comme des décompositions en branche et des découpes, dont les arbres de découpes sont en fait des « chenilles [1] ».

5.1.2. Fonction de partition et arbre de partitions

Comme nous l'avons déjà vu, l'outil que nous appelons arbre de partitions nous permet de généraliser un certain nombre de largeur de graphes. Mais cet outil ne s'arête pas là car il n'est pas borné au traitement de graphes. En effet, nous pouvons nous abstraire de la notion de graphe pour considérer la notion plus générale d'ensemble. Un graphe peut alors être vu, comme un ensemble de sommets liés par une relation d'adjacence, ou comme un ensemble d'arêtes liées par une relation d'incidence. Nous laissons alors de côté la notion de graphe (sur laquelle nous reviendrons ultérieurement) au profit de la notion d'ensemble.

Soit A un ensemble. Une *partition* de A, notée $\mathcal{P}$, est un ensemble de sous-ensembles disjoints de A dont l'union est égale à A. Soit $\mathcal{P} = \{P_1, \ldots, P_m\}$ une partition d'un ensemble A et soit X un sous-ensemble de A. La *restriction* de $\mathcal{P}$ à X correspond à la partition $\{P_1 \cap X, \ldots, P_m \cap X\}$ de X. Soit $\mathcal{Q} = \{Q_1, \ldots, Q_n\}$ une seconde partition de A. Nous disons que $\mathcal{Q}$ est un *raffinement* de $\mathcal{P}$ si, pour tout $Q_j \in \mathcal{Q}$, il existe $P_i \in \mathcal{P}$ tel que $Q_j \subseteq P_i$.

1. Une chenille est un arbre dont la suppression de ses feuilles en fait un chemin.

Définition 5.8 (Fonction de partition d'un ensemble)**.** Une *fonction de partition* d'un ensemble A est une fonction $\phi_A : \mathbb{N}^A \to \mathbb{N}$ qui associe un entier à chaque partition de A. Une fonction de partition ϕ_A est dite *monotone* si, pour tout raffinement $\mathcal{Q}$ d'une partition $\mathcal{P}$ de A, nous avons $\phi_A(\mathcal{P}) \leq \phi_A(\mathcal{Q})$.

À des fins de généralisation, nous aimerions que la définition d'une fonction de partition soit indépendante de l'ensemble auquel elle s'applique. En particulier, il serait utile qu'une fonction de partition d'un ensemble A induise une fonction de partition pour n'importe quel sous-ensemble de A. Nous considérons alors $\mathcal{A}$, un ensemble d'ensembles finis clos par sous-ensemble, c'est-à-dire que, pour tout ensemble X de $\mathcal{A}$ et pour tout sous-ensemble Y de X, l'ensemble Y est un élément de $\mathcal{A}$.

Définition 5.9 (Fonction de partition)**.** Une *fonction de partition monotone* ϕ sur $\mathcal{A}$ est une fonction qui associe une fonction de partition monotone ϕ_A à chaque ensemble $A \in \mathcal{A}$. La fonction ϕ est *close par sous-ensemble* si, pour chaque sous-ensemble X de A et pour toute partition $\mathcal{P}$ de A, nous avons $\phi_X(\mathcal{Q}) \leq \phi_A(\mathcal{P})$, avec $\mathcal{Q}$ la restriction de $\mathcal{P}$ à X.

Par la suite nous ne considérons plus que des fonctions de partitions monotones et closes par sous-ensemble, et nous omettons alors de le mentionner.

Définition 5.10 (Arbre de partitions)**.** Un *arbre de partitions* d'un ensemble A est un arbre T dont les feuilles sont en bijection avec les éléments de A. Pour chaque sommet interne x de T, nous considérons $C_1, \ldots, C_n$, les composantes connexes de $T \setminus x$. La partition $\mathcal{P}_x$ associée à x est alors égale à $\{P_1, \ldots, P_n\}$, où chaque P_i contient les éléments de A en bijection avec les feuilles de T présentes dans la composante C_i. De même, pour chaque arête e de T nous considérons C_1 et C_2, les composantes connexes de $T \setminus e$. La bipartition $\mathcal{P}_e$ associée à e est alors égale à $\{P_1, P_2\}$, où chaque P_i contient les éléments de A en bijection avec les feuilles de T présentes dans la composante C_i.

Dans le cas où A est un singleton, c'est-à-dire qu'il ne contient qu'un seul élément, alors, pour des raisons techniques, nous considérons que l'arbre de partitions correspondant à A est une unique arête. La première extrémité de l'arête est alors en bijection avec l'unique élément de A, alors que la seconde extrémité est considérée comme étant un sommet interne de T ; ce dernier sommet joue également le rôle de la racine de T.

Définition 5.11 (Largeur suivant ϕ)**.** Soit ϕ une fonction de partition de $\mathcal{A}$. Soit A un élément de $\mathcal{A}$ et soit T un arbre de partitions de A. La *largeur suivant* ϕ de T est égale à la valeur maximale renvoyée par ϕ lorsqu'on l'applique à chacune des partitions de A définies par T. La *largeur suivant* ϕ de A est égale à la valeur minimale des largeurs suivant ϕ parmis tous les arbres de partitions possibles pour A.

Définition 5.12 (Corps d'un arbre de partitions)**.** Le *corps*, noté $cp(T)$, d'un arbre de partitions T correspond à l'arbre obtenu à partir de T en supprimant toutes ses feuilles.

Définition 5.13 (Arbre de partitions q-ramifié)**.** Un arbre de partitions T est *q-ramifié* si $cp(T)$ est q-ramifié.

Nous disons que x est un *sommet de ramification* d'un arbre de partitions T, si x possède au moins deux fils dans T qui sont des sommets internes de T. Nous définissons également trois fonctions : $\mathrm{hr}_T : V(T) \to \mathbb{N}$ qui exprime la *hauteur de ramification* du sous arbre de T enraciné en un sommet donné ; $\mathrm{nr}_T : V(T) \to \mathbb{N}$ qui exprime le *nombre de ramifications* d'un sommet x de T, correspondant au nombre de fils de x qui sont des sommets de ramification dans T ; et $\mathrm{dr}_T(x,y) : V(T) \times V(T) \to \mathbb{N}$ qui, étant donné un sous-chemin P de T reliant un sommet x à un sommet y, exprime la *distance de ramification* entre x et y, laquelle compte le nombre de sommets de ramifications de T présents dans $P \setminus \{y\}$.

Définition 5.14 (Restriction d'un arbre de partitions)**.** Soit T un arbre de partitions d'un ensemble A et soit X un sous-ensemble de A. La *restriction* de T à X est définie par le sous-arbre minimal T^r de T qui couvre les feuilles de T en bijection avec les éléments de X, et r^r est le sommet de T^r le plus proche de r dans T. Dans le cas particulier où X est un singleton, T^r correspond alors à l'arête incidente à la feuille de T en bijection avec l'élément de X, et r^r désigne l'extrémité de cette arête qui est un sommet interne de T.

5.1.3. Largeurs de graphes et arbres de partitions

Comme nous l'avons précédemment fait remarquer, les notions que nous venons de présenter s'appliquent à des ensembles. Les notions de fonctions de partition et d'arbres de partitions sont des généralisations d'un certain nombre de décompositions de graphes et des largeurs associées [1].

Il est parfois nécessaire, suivant la largeur considérée, de poser des restrictions sur la forme de l'arbre de partitions, d'ajouter des contraintes à la bijection liée aux feuilles de l'arbre de partitions ou encore d'utiliser une fonction de partition modifiée, afin d'exprimer les largeurs de graphes à l'aide de fonctions de partition et d'arbre de partitions. Dans ce qui va suivre, nous présentons les restrictions nécessaires à la représentation de la largeur arborescente spéciale, de la largeur de branches, de la largeur linéaire, de la largeur de coupe ainsi que de la largeur de découpe sous la forme de fonctions de partition et d'arbre de partitions. Nous commençons cette présentation par le cas de la largeur arborescente q-ramifiée (qui, comme nous le verrons, englobe la largeur de chemin ainsi que la largeur arborescente).

Afin de recoller à la Définition 5.9 nous considérons $\mathcal{V}$, l'ensemble des ensembles de sommets de tous les graphes finis, ainsi que $\mathcal{E}$, l'ensemble des ensembles d'arêtes de tous les graphes finis. En d'autres terme, pour tout graphe $G = (V,E)$, nous avons $V \in \mathcal{V}$ et $E \in \mathcal{E}$.

Les diverses largeurs arborescentes et leurs fonctions de partition

Pout tout graphe $G = (V,E)$, nous définissons Δ comme étant la fonction qui, à chaque partition $\mathcal{P} = \{E_1,\dots,E_r\}$ de E, associe l'ensemble des sommets qui sont à la fois incidents aux arêtes de E_i ainsi qu'à celles de E_j, avec $i \neq j$.

Définition 5.15 (Fonction de partition δ)**.** Soit E un élément de $\mathcal{E}$. La fonction δ_E est la fonction de partition qui associe à E l'entier $|\Delta(\mathcal{P})|$ à chacune des partitions $\mathcal{P}$ possible pour E. Soit δ la fonction de partition de $\mathcal{E}$ qui associe δ_E à chaque élément $E \in \mathcal{E}$.

Lemme 5.16 (Amini *et al.* [1])**.** *Pour tout graphe $G = (V, E)$, la largeur arborescente de G est au plus $k \geq 1$, si et seulement si la largeur suivant δ de E est au plus $k+1$.*

Démonstration. Soit T un arbre de partitions de E, avec une largeur suivant δ au plus égale à $k+1$. Alors, nous pouvons aisément vérifier que $(cp(T), \{\Delta(\mathcal{P}_v) : v \in V(cp(T))\})$ est une décomposition arborescente de G de largeur au plus k. Réciproquement, si $(T, \mathcal{X})$ est une décomposition arborescente de G de largeur au plus k alors, pour chaque arête $uv \in E$, nous choisissons arbitrairement un nœud α de T tel que X_α contient à la fois u et v, puis nous lui ajoutons une feuille β telle que X_β contienne uniquement u et v. Finalement nous considérons T' le sous-arbre minimal de T contenant toutes les feuilles nouvellement introduites. Finalement, l'arbre T' est un arbre de partitions de E et sa largeur suivant δ est au plus $k+1$. □

De façon similaire, il est possible de prouver le lemme suivant.

Lemme 5.17 (Amini *et al.* [1])**.** *Pour tout graphe $G = (V, E)$, la largeur de chemin de G est au plus $k \geq 1$, si et seulement si il existe un arbre de partition 0-ramifié pour E dont la largeur suivant δ est au plus $k+1$.*

Et, plus généralement, nous pouvons considérer le lemme ci-dessous.

Lemme 5.18. *Pour tout graphe $G = (V, E)$ et pour tout $q \geq 0$, la largeur arborescente q-ramifiée de G est au plus $k \geq 1$, si et seulement si il existe un arbre de partition q-ramifié pour E dont la largeur suivant δ est au plus $k+1$.*

Dans le cas de la largeur arborescente spéciale nous utilisons encore une fois la fonction δ mais, cette fois-ci, nous introduisons une restriction sur la forme des arbres de partitions. Pour tout graphe $G = (V, E)$, au lieu de considérer le plus petit δ_E parmi tous les arbres de partitions de E, nous ne considérons que ceux qui respectent la règle qui suit. Soit T un arbre de partitions de E, pour chaque sommet $v \in V$, nous considérons T_v le sous-arbre minimum de T couvrant toutes les feuilles de T telles que les arêtes de E correspondantes admettent v en tant qu'une de leurs extrémités. Nous avons alors que T_v est une « chenille ».

Les largeurs de branche et linéaire et leurs fonctions de partition

La largeur de branche ainsi que la largeur linéaire peuvent être exprimées grâce à la fonction de partition suivante.

Définition 5.19 (Fonction de partition $max\delta$)**.** Soit $max\delta_E$ la fonction de partition de E qui associe $\max_{1 \leq i \leq r} \delta(E_i, E \setminus E_i)$ à chaque partition $\{E_1, \ldots, E_r\}$ de E. Soit $max\delta$ la fonction de partition de $\mathcal{E}$ qui associe $max\delta_E$ à chaque élément $E \in \mathcal{E}$.

Lemme 5.20. *La largeur de branche de G est au plus $k \geq 1$ si et seulement si il existe un arbre de partitions de E, avec une largeur suivant $max\delta$ au plus égale à k, tel que ses sommets internes sont de degré au plus* 3.

Ce lemme est une conséquence directe de la définition de la largeur de branche.

Lemme 5.21. *La largeur linéaire de G est au plus $k \geq 2$ si et seulement si il existe un arbre de partitions* 0*-ramifié de E, avec une largeur suivant $max\delta$ au plus égale à k, tel que ses sommets internes sont de degré au plus* 3.

Ce résultat découle de la correspondance naturelle entre un tel arbre de partitions pour E et un ordre sur E. Notons que cela n'est pas vrai pour $k = 1$ (le chemin de longueur 3 en est la preuve).

Les largeurs de découpe et de coupe et leurs fonctions de partition

Pout tout graphe $G = (V, E)$, nous définissons $Edge\delta$ comme étant la fonction qui, à chaque partition $\mathcal{P} = \{V_1, \ldots, V_r\}$ de V, associe la taille de l'ensemble des arêtes qui relient les sommets de V_i et V_j, avec $i \neq j$.

Définition 5.22 (Fonction de partition $maxEdge\delta$)**.** Soit $maxEdge\delta_V$ la fonction de partition d'un ensemble $V \in \mathcal{V}$ qui associe $\max_{1 \leq i \leq r} Edge\delta(V_i, V \setminus V_i)$ à chaque partition $(V_1, \ldots, V_r)$ de V. Soit $maxEdge\delta$ la fonction de partition de $\mathcal{V}$ qui associe $maxEdge\delta_V$ à chaque élément $V \in \mathcal{V}$.

Lemme 5.23. *La largeur de découpe de G est au plus égale à $k \geq 1$ si et seulement si il existe un arbre de partitions de V, avec une largeur suivant $maxEdge\delta$ au plus égale à k, tel que ses sommets internes sont de degré au plus* 3.

Ce lemme est cette fois-ci une conséquence directe de la définition de la largeur de découpe.

La fonction de partition $maxEdge\delta$ peut aussi exprimer la largeur de coupe d'un graphe $G = (V, E)$, quand celle-ci est au moins égale au degré maximum d de G. Notons que tout arbre de partitions 0-ramifié avec un degré maximum égal à 3 est tel que sa largeur suivant $maxEdge\delta$ est au moins d. Réciproquement, tout arbre de partitions 0-ramifié de V, et avec un degré maximum égal à 3, peut être vu comme un ordre linéaire sur les sommets de G, ce qui implique que la largeur suivant $maxEdge\delta$ de G est minorée par la largeur de coupe de G. Plus précisement, la plus petite largeur suivant $maxEdge\delta$ parmis les arbres de partitions 0-ramifiés de V, avec un degré maximum égal à 3, vaut $\max\{\mathrm{cw}(G), d\}$, où $\mathrm{cw}(G)$ dénote la largeur de coupe de G. De façon générale, pour exprimer la largeur de coupe d'un graphe, nous faisons appel à une fonction de partition plus restrictive.

Définition 5.24 (Fonction de partition 3-$maxEdge\delta$)**.** Soit 3-$maxEdge\delta_V$ la fonction de partition d'un ensemble $V \in \mathcal{V}$ qui, à chaque partition (V_1, V_2, V_3) de V, avec $|V_3| = 1$, associe la fontion $\max\{Edge\delta(V_1, V \setminus V_1), Edge\delta(V_2, V \setminus V_2)\}$. Soit 3-$maxEdge\delta$ la fonction de partition de $\mathcal{V}$ qui associe 3-$maxEdge\delta_V$ à chaque élément $V \in \mathcal{V}$.

Lemme 5.25. *La largeur de coupe de G est au plus égale à $k \geq 1$ si et seulement si il existe un arbre de partitions 0-ramifié de V, avec une largeur suivant 3-maxEdgeδ au plus égale à k, tel que ses sommets internes sont de degré au plus 3.*

Ce résultat découle de la correspondance naturelle entre un tel arbre de partition pour V et un ordre sur V.

5.2. Problème(s) de reconnaissance

Le principal résultat de ce chapitre est un algorithme à paramètre fixé qui permet de déterminer si la largeur suivant une fonction de partition ϕ d'un ensemble A est majorée par un entier k fixé (le paramètre), et ce en temps linéaire sur la taille de A. Afin d'honorer la complexité annoncée il faut que ϕ satisfasse certaines conditions, lesquelles sont respectées par les largeurs que nous venons de présenter. Ce dernier état de fait implique ainsi que l'algorithme du Théorème 5.30 ci-dessous généralise les résultats de [5, 7, 9, 10, 49].

5.2.1. Le théorème et ses corollaires

Avant de pouvoir énoncer notre résultat principal, il nous faut introduire quelques notions supplémentaires ainsi que certaines restrictions sur les fonctions de partitions qu'il est possible d'utiliser.

Conditions suffisantes pour un algorithme linéaire

Étant donné que les arbres de partitions généralisent la notion de décomposition arborescente et que notre algorithme portant sur ces arbres de partitions est une adaptation de l'algorithme portant sur la décomposition arborescente, énoncé dans [5], il semble assez naturel d'étendre la notion de « jolie décomposition arborescente » à nos besoins.

Définition 5.26 (Décomposition d'un ensemble)**.** Une *décomposition* d'un ensemble fini A est un couple $(\mathcal{T},\mathcal{X})$, avec $\mathcal{X} = \{X_1,\ldots,X_p\}$ une famille de sous-ensembles de A et $\mathcal{T}$ un arbre enraciné dont les nœuds sont étiquetés par les éléments de $\mathcal{X}$, tel que :
- l'union de tous les éléments de $\mathcal{X}$ est égale à A ;
- pour chaque élément $a \in A$, l'ensemble des éléments de $\mathcal{X}$ contenant a induit un sous-arbre dans $\mathcal{T}$.

La *largeur* de $(\mathcal{T},\mathcal{X})$ est définie par le maximum sur la taille des éléments de $\mathcal{X}$.

Définition 5.27 (Jolie décomposition)**.** Une décomposition d'un ensemble A est dite *jolie* si ses nœuds sont répartis en quatres types :
- un *nœud de départ* est une feuille de $\mathcal{T}$ et contient un unique élément de A ;
- un *nœud d'oubli* α a un unique fils β tel que $X_\alpha = X_\beta \setminus \{a\}$ et $a \in X_\beta$;
- un *nœud d'introduction* α a un unique fils β tel que $X_\alpha = X_\beta \cup \{a\}$ et $a \notin X_\beta$;
- un *nœud de jonction* α a exactement deux fils β et γ tels que $X_\alpha = X_\beta = X_\gamma$.

Dans le cas d'un nœud d'oubli, l'élément a est appelé l'*élément oublié* et, dans le cas d'un nœud d'introduction, a est l'*élément introduit*.

Lemme 5.28. *Pour toute décomposition $(\mathcal{T},\mathcal{X})$ d'un ensemble A, il existe une jolie décomposition $(\mathcal{T}',\mathcal{X}')$ telle que le nombre de nœuds de $\mathcal{T}'$ est au plus 4 fois le nombre de nœuds de $\mathcal{T}$.*

Pour chaque nœud α de $\mathcal{T}$, nous notons $\mathcal{T}_\alpha$ le sous-arbre de $\mathcal{T}$ enraciné en α et A_α dénote le sous-ensemble de A couvert par les nœuds de $\mathcal{T}_\alpha$.

Définition 5.29 (Fonction de partition compatible)**.** Soit $(\mathcal{T},\mathcal{X})$ une jolie décomposition d'un ensemble finit A. La fonction de partition ϕ est dite *compatible* avec $(\mathcal{T},\mathcal{X})$ si elle respecte les conditions suivantes.

1. Soit α un nœud d'introduction avec un fils β et soit a l'élément introduit. Il existe une fonction I_ϕ prenant en paramètres un entier i, une partition $\mathcal{P}$ de A restreinte à X_α ainsi que l'élément a et retourne un nouvel entier. La fonction I_ϕ est telle que si $i < i'$, alors $I_\phi(i,\mathcal{P}\cap X_\alpha,a) < I_\phi(i',\mathcal{P}\cap X_\alpha,a)$ et que :
$$\phi_{A_\alpha}(\mathcal{P}) = I_\phi(\phi_{A_\beta}(\mathcal{P}\cap A_\beta),\mathcal{P}\cap X_\alpha,a).$$
De plus, si $\mathcal{P}\cap X_\alpha$ est constituée d'une unique partie, alors $\phi_{A_\alpha}(\mathcal{P}) = \phi_{A_\beta}(\mathcal{P}\cap A_\beta)$.
2. Soit α un nœud de jonction avec ses deux fils β et γ. Il existe une fonction J_ϕ prenant en paramètres deux entiers i et j ainsi qu'une partition $\mathcal{P}$ de A restreinte à X_α et retourne un nouvel entier. La fonction J_ϕ est telle que si $i < i'$ et $j \leq j'$ ou $i \leq i'$ et $j < j'$, alors $J_\phi(i,j,\mathcal{P}\cap X_\alpha) < J_\phi(i',j',\mathcal{P}\cap X_\alpha)$ et que :
$$\phi_{A_\alpha}(\mathcal{P}) = J_\phi(\phi_{A_\beta}(\mathcal{P}\cap A_\beta),\phi_{A_\gamma}(\mathcal{P}\cap A_\gamma),\mathcal{P}\cap X_\alpha).$$
De plus, si $\mathcal{P}\cap A_\gamma$ est constituée d'une unique partie, alors $\phi_{A_\alpha}(\mathcal{P}) = \phi_{A_\beta}(\mathcal{P}\cap A_\beta)$ et inversement, si $\mathcal{P}\cap A_\beta$ est constituée d'une unique partie, alors nous avons $\phi_{A_\alpha}(\mathcal{P}) = \phi_{A_\gamma}(\mathcal{P}\cap A_\gamma)$.

Cette définition, qui peut paraître un peu abrupte au premier abord, nous indique en fait les propriétés suivantes. Dans le cas où α est un nœud d'introduction, il est possible de calculer la largeur suivant ϕ de $\mathcal{P}$ au niveau de α en fonction de la largeur suivant ϕ de $\mathcal{P}$ au niveau de β, et ce en ne prenant en compte que les éléments de X_α et l'élément introduit. De même, dans le cas où α est un nœud de jonction, il est alors possible de calculer la largeur suivant ϕ de $\mathcal{P}$ au niveau de α en fonction des largeurs suivant ϕ de $\mathcal{P}$ aux niveaux de β et γ, et ce en ne prenant en compte que les éléments de X_α.

Nous observons alors que les fonctions I_ϕ et J_ϕ ont un temps de calcul majoré en fonction de $|X_\alpha|$.

Énoncé du théorème

Théorème 5.30. *Soit ϕ une fonction de partition monotone et close par sous-ensemble. Il existe un algorithme linéaire qui, étant donnés $k,l \geq 1$ et $q \geq 0$ trois constantes (q pouvant prendre la valeur ∞), un ensemble fini A ainsi qu'une jolie décomposition de A, de largeur au plus l et compatible avec ϕ, décide si la largeur q-ramifiée suivant ϕ de A est au plus k.*

La preuve de ce théorème est complexe et constitue l'objet principal de ce chapitre. Afin de rendre cette preuve plus simple, nous allons commencer par décrire un algorithme naïf de complexité non polynomiale puis, en adaptant la méthode de compression présentée par Bodlaender et Kloks dans [5], permettant de résoudre le problème dans le cadre de la largeur arborescente, nous montrerons comment il est possible d'obtenir un algorithme de complexité linéaire pour la largeur q-ramifiée suivant ϕ.

Mais avant de présenter cette preuve, nous nous intéressons à quelques résultats découlant de ce théorème.

Corollaire 5.31. *Soit ϕ une fonction de partition monotone et close par sous-ensemble. Il existe un algorithme linéaire qui, étant donnés $k \geq 1$ et $q \geq 0$ deux constantes (q pouvant prendre la valeur ∞), ainsi qu'une classe $\mathcal{A}$ d'ensembles finis telle que, pour tout ensemble $A \in \mathcal{A}$, il existe un algorithme linéaire calculant une jolie décomposition de A (de taille au plus $4 \cdot |A|$), compatible avec ϕ et de largeur au plus $f(k)$, décide si la largeur q-ramifiée suivant ϕ de $A \in \mathcal{A}$ est au plus k, avec f une fonction linéaire calculable.*

Paramétrisation des largeurs de graphes

Dans cette section, nous présentons une application du Théorème 5.30 aux largeurs de graphes qui entrent dans le cadre des arbres de partitions.

Théorème 5.32. *Soit k et q deux constantes. Il existe alors des algorithmes linéaires qui, étant donné un graphe G de degré maximum majoré par une fonction calculable $f(k,q)$, décident si les largeurs arborescente q-ramifiée, arborescente spéciale, de branche, linéaire, de coupe ou de découpe pour G sont majorées par k.*

Démonstration. Dans la Section 5.1.3 nous montrions que les largeurs de graphes susmentionnées peuvent être généralisées par l'utilisation d'arbres de partitions. Par conséquent, la preuve de ce théorème consiste à montrer que les fonctions de partitions correspondant à ces largeur de graphes respectent les conditions du Corollaire 5.31.

Dans [4], Bodlaender présente un algorithme linéaire qui, étant donné un graphe $G = (V, E)$ et un constante $l = \mathcal{O}(k)$, décide si la largeur arborescente de G est majorée par l et, dans le cas d'une réponse positive, construit une décomposition arborescente de G de largeur au plus l. De plus, à partir d'une décomposition arborescente quelconque de G, il est possible de construire, toujours en temps linéaire, une jolie décomposition arborescente de G de même largeur [5]. Finalement, le Lemme 5.34 ci-dessous, nous montre que toute jolie décomposition arborescente de G peut être transformée en une jolie décomposition de E de largeur au plus $d \cdot l$, où $d \leq f(k,q)$ représente le degré maximum de G, et ce en temps linéaire.

Grâce aux Lemmes 5.36, 5.39 et 5.40 nous savons que les fonctions de partitions associées aux largeurs de graphes que nous considérons sont compatibles avec les jolies décompositions correspondantes. Par conséquent, l'algorithme de [4] ainsi que le Lemme 5.34 nous offrent la possibilité de construire une jolie décomposition de E. Comme, par hypothèse, le degré maximum de G est majoré par $f(k,q)$, la largeur d'une telle jolie décomposition est elle aussi majorée par $l \cdot f(k,q)$. Les prérequis du Corollaire 5.31 sont alors satisfaits. □

Voici d'abord un premier lemme dont nous omettons la preuve car elle va d'elle même.

Lemme 5.33. *Les fonctions de partitions δ, $max\delta$, $Edge\delta$, $maxEdge\delta$ et 3-$maxEdge\delta$ sont monotones et closes par sous-ensembles.*

Nous montrons maintenant comment passer d'une jolie décomposition arborescente à une jolie décomposition.

Lemme 5.34. *Toute jolie décomposition arborescente $(\mathcal{T},\mathcal{X})$ d'un graphe $G = (V,E)$ de largeur au plus k peut être transformée en une jolie décomposition $(\mathcal{D},\mathcal{Y})$ de E. De plus, si le degré maximum de G est au plus d, alors la largeur de $(\mathcal{D},\mathcal{Y})$ est majorée par $k \cdot d$.*

Démonstration. Pour tout nœud a de $\mathcal{T}$, nous considérons $\mathcal{T}_a$ le sous-arbre de $\mathcal{T}$ enraciné en a ainsi que V_a, le sous-ensemble de V couvert par les nœuds de $\mathcal{T}_a$, et E_a, l'ensemble des arêtes de $G[V_a]$ qui ont (au moins) une incidence dans X_a.

Les nœuds de départ. Chaque nœud de départ a de $\mathcal{T}$ correspond à un nœud de départ α de $\mathcal{D}$, avec $Y_\alpha = E_a$. Notons que $E_a = \emptyset$ mais, afin de simplifier la preuve, nous gardons quand même le nœud α dans $\mathcal{D}$ jusqu'à la fin de la construction de $(\mathcal{D},\mathcal{Y})$.

Les nœuds d'oubli. Soit a un nœud d'oubli de $\mathcal{T}$ et soit b son fils ; nous appelons v le sommet oublié. En considérant les arêtes $e_1,\dots,e_p$ de $G[V_b \setminus X_a]$ incidentes à v, nous construisons un chemin $\mathcal{P} = (\alpha_p,\dots,\alpha_1)$ tel que le nœud α_i contient les arêtes $Y_{\alpha_i} = E_b \setminus \{e_j : 1 \le j \le i\}$. Puis nous ajoutons $\mathcal{P}$ dans $\mathcal{D}$, en faisant de α_1 le père de la racine de $\mathcal{D}$.

Les nœuds d'introduction. Soit a un nœud d'introduction de $\mathcal{T}$ et soit b son fils ; nous appelons v le sommet introduit. En considérant les arêtes $e_1,\dots,e_p$ de $G[X_a]$ incidentes à v, nous construisons un chemin $\mathcal{P} = (\alpha_p,\dots,\alpha_1)$ tel que le nœud α_i contient les arêtes $Y_{\alpha_i} = E_b \cup \{e_j : 1 \le j \le i\}$. Puis nous ajoutons $\mathcal{P}$ dans $\mathcal{D}$, en faisant de α_1 le père de la racine de $\mathcal{D}$.

Les nœuds de jonction. Soit a un nœud de jonction de $\mathcal{T}$ et soient b et c ses fils. Soit $\mathcal{D}_\beta$ la jolie décomposition correspondant à la jolie décomposition arborescente $\mathcal{T}_b$. Nous ajoutons alors les arêtes de $E_c \setminus E_b$, à $\mathcal{D}_\beta$ en suivant la phase d'introduction décrite ci-dessus et nous obtenons la jolie décomposition $\mathcal{D}'_\beta$. Nous effectuons la même manœuvre en considérant $\mathcal{D}_\gamma$, la jolie décomposition correspondant à $\mathcal{T}_c$, ainsi que les arêtes de $E_b \setminus E_c$, afin de construire $\mathcal{D}'_\gamma$. Ainsi, les racines de $\mathcal{D}'_\beta$ et $\mathcal{D}'_\gamma$ contiennent le même ensemble d'arêtes $E_a = E_b \cup E_c$. Il nous suffit alors de créer un nouveau nœud α, avec $Y_\alpha = E_a$ et de le définir en temps que père des racines de $\mathcal{D}'_\beta$ et $\mathcal{D}'_\gamma$.

Avant de conclure, nous supprimons les feuilles de $\mathcal{D}$, lequelles ne contiennent aucune arête. La décomposition $(\mathcal{D},\mathcal{Y})$ ainsi obtenue est une jolie décomposition de E, et sa largeur (le nombre maximum d'arêtes contenu par les nœuds de $\mathcal{D}$) est au plus égale à la largeur de $(\mathcal{T},\mathcal{X})$ augmentée de 1 puis multipliée par le degré maximum de G. □

Remarque 5.35. Au lieu de transformer les feuilles de $\mathcal{T}$ en feuilles vides dans $\mathcal{D}$, nous pourrions simplement commencer par supprimer les feuilles de $\mathcal{T}$. Cependant rien ne nous garantit que les pères de ces feuilles dans $\mathcal{T}$ induiront des feuilles non vides dans $\mathcal{D}$. C'est pourquoi nous préférons garder les feuilles vides jusqu'à la fin de la construction de $\mathcal{D}$.

Lemme 5.36. *Soit $G = (V, E)$ un graphe de degré maximum d. Étant donné une jolie décomposition arborescente $(\mathcal{T}, \mathcal{X})$ de G de largeur au plus $k \geq 1$, nous pouvons calculer en temps linéaire une jolie décomposition $(\mathcal{D}, \mathcal{Y})$ de E de largeur au plus $k \cdot d$ et telle que la fonction de partition δ est compatible avec $(\mathcal{D}, \mathcal{Y})$.*

Démonstration. Nous considérons que $(\mathcal{D}, \mathcal{Y})$ est la jolie décomposition de E obtenue à partir de $(\mathcal{T}, \mathcal{X})$ en utilisant le Lemme 5.34. Il nous faut alors montrer que la fonction de partition δ est compatible avec $(\mathcal{D}, \mathcal{Y})$. Pour cela nous décrivons explicitement les fonctions I_δ et J_δ et nous montrons qu'elles respectent bien les propriétés voulues.

Durant cette preuve, étant donné α un nœud de $\mathcal{D}$, nous définissons $\mathcal{D}_\alpha$, le sous-arbre de $\mathcal{D}$ enraciné en α, ainsi que E_α, le sous-ensemble de E couvert par les nœuds de $\mathcal{D}_\alpha$.

La fonction I_δ. Soit α un nœud d'introduction de $\mathcal{D}$ et soit β son fils ; nous appelons e l'arête introduite. Soit $\mathcal{P}$ une partition de E_α et soit $\Delta(\mathcal{P})$ l'ensemble de sommets dont la taille définit $\delta_{E_\alpha}(\mathcal{P})$.

Les sommets de $\Delta(\mathcal{P})$ peuvent être divisés en deux sous-ensemble disjoints. Le premier correspond à l'ensemble des sommets incidents à au moins deux arêtes e_1 et e_2, toutes deux différentes de e, et telles que e_1 et e_2 apparaissent dans deux partie différentes de $\mathcal{P}$; ce premier ensemble définit en fait $\Delta(\mathcal{P} \cap E_\beta)$. Le second ensemble contient alors les sommets restants, qui sont les sommets incidents à e et tels que, pour un sommet $v \in e$ donné, l'ensemble des arêtes incidentes à v différentes de e, et qui n'apparaissent pas dans la partie de $\mathcal{P}$ contenant e, apparaissent dans une même partie de $\mathcal{P}$; nous notons alors cet ensemble $\Delta_{\alpha,e}(\mathcal{P})$. Nous pouvons alors écrire $\Delta(\mathcal{P}) = \Delta(\mathcal{P} \cap E_\beta) \cup \Delta_{\alpha,e}(\mathcal{P})$, avec $\Delta(\mathcal{P} \cap E_\beta) \cap \Delta_{\alpha,e}(\mathcal{P}) = \emptyset$. Ce qui implique que $|\Delta(\mathcal{P})| = \delta_{E_\beta}(\mathcal{P} \cap E_\beta) + |\Delta_{\alpha,e}(\mathcal{P})|$, en rappelant que $\delta_{E_\beta}(\mathcal{P} \cap E_\beta) = |\Delta(\mathcal{P} \cap E_\beta)|$. Par définition de $(\mathcal{D}, \mathcal{Y})$ et, comme elle a été construite à partir de $(\mathcal{T}, \mathcal{X})$, chaque arête de $E_\beta = E_\alpha \setminus \{e\}$ qui partage une incidence avec e apparait dans Y_α. Nous avons donc $\Delta_{\alpha,e}(\mathcal{P}) = \Delta_{\alpha,e}(\mathcal{P} \cap Y_\alpha)$. Et, nous définissons ainsi $I_\delta(i, \mathcal{P} \cap Y_\alpha, e) = i + |\Delta_{\alpha,e}(\mathcal{P} \cap Y_\alpha)|$. Il est assez évident que, si $i < i'$, alors $I_\delta(i, \mathcal{P} \cap Y_\alpha, e) < I_\delta(i', \mathcal{P} \cap Y_\alpha, e)$. De plus, si la taille de Y_α est majorée par une constante, alors le calcul de $I_\delta(i, \mathcal{P} \cap Y_\alpha, e)$ s'effectue en temps constant. La fonction I_δ respecte donc bien la première règle de compatibilité avec $(\mathcal{D}, \mathcal{Y})$. Notons que si $\mathcal{P} \cap Y_\alpha$ est constitué d'une unique partie, alors $\Delta_{\alpha,e}(\mathcal{P} \cap Y_\alpha) = \emptyset$, ce qui implique que $\delta_{E_\alpha}(\mathcal{P}) = \delta_{E_\beta}(\mathcal{P} \cap E_\beta)$.

La fonction I_δ. Soit α un nœud de jonction de $\mathcal{D}$ et soient β et γ ses deux fils. Soit $\mathcal{P}$ une partition de E_α et soit $\Delta(\mathcal{P})$ l'ensemble de sommets dont la taille définit $\delta_{E_\alpha}(\mathcal{P})$. Nous considérons également les ensembles $\Delta(\mathcal{P} \cap E_\beta)$ et $\Delta(\mathcal{P} \cap E_\gamma)$ respectivement associés à $\delta_{E_\beta}(\mathcal{P} \cap E_\beta)$ et $\delta_{E_\gamma}(\mathcal{P} \cap E_\gamma)$.

Nous remarquons tout d'abord que $\Delta(\mathcal{P} \cap E_\beta) \cup \Delta(\mathcal{P} \cap E_\gamma) \subseteq \Delta(\mathcal{P})$. De plus, par définition de $(\mathcal{D}, \mathcal{Y})$, aucune arête de $E_\beta \setminus Y_\alpha$ ne peut partager d'incidence avec une arête $E_\gamma \setminus Y_\alpha$.

Ce qui implique $\Delta(\mathcal{P}) \subseteq \Delta(\mathcal{P} \cap E_\beta) \cup \Delta(\mathcal{P} \cap E_\gamma)$ et donc $\Delta(\mathcal{P}) = \Delta(\mathcal{P} \cap E_\beta) \cup \Delta(\mathcal{P} \cap E_\gamma)$. Si nous ajoutons à cela, le fait que $\Delta(\mathcal{P} \cap E_\beta) \cap \Delta(\mathcal{P} \cap E_\gamma) = \Delta(\mathcal{P} \cap Y_\alpha)$, nous pouvons écrire $\Delta(\mathcal{P}) = \Delta(\mathcal{P} \cap E_\beta) \cup \big(\Delta(\mathcal{P} \cap E_\gamma) \setminus \Delta(\mathcal{P} \cap Y_\alpha)\big)$. Cette dernière égalité se traduit par $\delta_{E_\alpha}(\mathcal{P}) = \delta_{E_\beta}(\mathcal{P} \cap E_\beta) + \delta_{E_\gamma}(\mathcal{P} \cap E_\gamma) - |\Delta(\mathcal{P} \cap Y_\alpha)|$, nous permettant ainsi de définir $J_\delta(i, j, \mathcal{P} \cap Y_\alpha) = i + j - \delta(\mathcal{P} \cap Y_\alpha)$. Encore une fois, il est évident que si $i < i'$ et $j \leq j'$ ou $i \leq i'$ et $j < j'$, alors $J_\delta(i, j, \mathcal{P} \cap Y_\alpha) < J_\delta(i', j', \mathcal{P} \cap Y_\alpha)$ et, si la taille de Y_α est majorée par une constante, alors le calcul de $J_\delta(i, j, \mathcal{P} \cap Y_\alpha)$ s'effectue en temps constant. Par ailleurs, dans le cas où $\mathcal{P} \cap E_\beta$ est constitué d'une unique partie, alors $\Delta(\mathcal{P} \cap E_\beta) = \emptyset$ et $\Delta(\mathcal{P} \cap Y_\alpha) = \emptyset$, impliquant alors que $\Delta(\mathcal{P}) = \Delta(\mathcal{P} \cap E_\gamma)$ et que $\delta_{E_\alpha}(\mathcal{P}) = \delta_{E_\gamma}(\mathcal{P} \cap E_\gamma)$. De même, si $\mathcal{P} \cap E_\gamma$ est constitué d'une unique partie, alors $\delta_{E_\alpha}(\mathcal{P}) = \delta_{E_\beta}(\mathcal{P} \cap E_\beta)$. □

Le comportement de la fonction de partition $Edge\delta$ est équivalent à celui de δ si nous inversons le rôle des sommets et des arêtes.

Lemme 5.37. *Soit $G = (V, E)$ un graphe de degré maximum d. Alors, la fonction de partition $Edge\delta$ est compatible avec toute jolie décomposition arborescente $(\mathcal{T}, \mathcal{X})$ de G.*

Démonstration. Notons tout d'abord que toute jolie décomposition arborescente de G est une jolie décomposition de V. Pour montrer que $Edge\delta$ est compatible avec $(\mathcal{T}, \mathcal{X})$, nous reprenons la preuve du Lemme 5.36 en utilisant les fonctions suivantes.

Étant donnés un nœud d'introduction α de $\mathcal{T}$, son fils β et une partition $\mathcal{P}$ de V_α, la fonction $I_{Edge\delta}$ est telle que $I_{Edge\delta}(i, \mathcal{P} \cap X_\alpha, v) = i + Edge\Delta_{\alpha,v}(\mathcal{P} \cap X_\alpha)$, avec v le sommet introduit et $Edge\Delta_{\alpha,v}(\mathcal{P} \cap X_\alpha)$ l'ensemble des arêtes incidentes à v et dont les autres extrémités apparaissent dans des parties de $\mathcal{P}$ ne contenant pas v.

Étant donnés un nœud de jonction α de $\mathcal{T}$, ses deux fils β et γ ainsi qu'une partition $\mathcal{P}$ de V_α, la fonction $J_{Edge\delta}$ est telle que $J_{Edge\delta}(i, j, \mathcal{P} \cap X_\alpha) = i + j - Edge\delta(\mathcal{P} \cap X_\alpha)$. □

Afin de prouver la compatibilité des fonctions $max\delta$ et $maxEdge\delta$ avec les jolies décompositions de E et V respectivement, nous nous appuyons sur la propriété suivante.

Propriété 5.38. *Soit λ une fonction de partition quelconque, et soit $max\lambda$ qui associe $\max_{1 \leq i \leq n} \lambda(A_i, A \setminus A_i)$ à toute partition $(A_1, \ldots, A_n)$ d'un ensemble A. Si λ est compatible avec une jolie décomposition de A, alors $max\lambda$ l'est aussi.*

La preuve de cette propriété n'est pas simple et elle utilise des arguments dont nous nous servons dans la démonstration du Théorème 5.30, c'est pourquoi nous reportons la preuve de la Propriété 5.38 dans la Section 5.4.5, à la fin de la preuve du théorème.

Lemme 5.39. *Soit $G = (V, E)$ un graphe. La fonction de partition $max\delta$ est compatible avec toute jolie décomposition de E.*

Lemme 5.40. *Soit $G = (V, E)$ un graphe. Les fonctions de partition $maxEdge\delta$ ainsi que 3-$maxEdge\delta$ sont compatibles avec toute jolie décomposition de V.*

5.2.2. Un algorithme naïf

Dans cette section nous décrivons un premier algorithme relativement simple pour résoudre le problème de reconnaissance suivant. Étant donnés un ensemble A, une jolie décomposition $(\mathcal{T},\mathcal{X})$ de largeur l pour A, et une fonction de partition ϕ, compatible avec $(\mathcal{T},\mathcal{X})$, décider si la largeur q-ramifiée suivant ϕ de A ne dépasse pas un certain entier k, avec k, q et l trois constantes.

Avant d'aller plus loins, nous introduisons quelques notations supplémentaires. Soit α un nœud de $\mathcal{T}$, le sous-arbre de $\mathcal{T}$ enraciné en α est noté $\mathcal{T}_\alpha$ et il définit le sous-ensemble $A_\alpha = \cup_{\lambda\in V(\mathcal{T}_\alpha)} X_\lambda$. Nous avons également besoin de la notion d'*ensemble complet des arbres de partitions* d'un nœud α (abbrévié $FSPT_\alpha$, pour *full set of partitioning trees*) qui, comme son nom l'indique, représente tous les (k,q)-arbres de partitions étiquetés de A_α. Si ρ est la racine de $(\mathcal{T},\mathcal{X})$, alors $A = A_\rho$ et nous avons que $FSPT_\rho$ est vide si et seulement si la largeur q-ramifiée suivant ϕ de A dépasse k.

L'idée principale de l'algorithme, qui utilise une technique de programmation dynamique, est de commencer par calculer, pour chaque feuille γ de $\mathcal{T}$, l'ensemble $FSPT_\gamma$. Puis, pour chaque nœud interne α de $\mathcal{T}$, tel que pour chaque fils β de α nous avons préalablement calculé $FSPT_\beta$, nous calculons $FSPT_\alpha$. Le calcul montant se termine quand nous atteignons la racine ρ de $\mathcal{T}$. Il nous suffit alors de tester si $FSPT_\rho$ est vide ou non pour savoir si la largeur q-ramifiée suivant ϕ de A est majorée par k. Comme $(\mathcal{T},\mathcal{X})$ est une jolie décomposition, nous avons quatre types de nœuds à étudier.

Procédure de Départ. Soit α un nœud de départ de $\mathcal{T}$. Par définition, α est une feuille de $\mathcal{T}$ et elle contient un unique élément de A. Afin de determiner l'ensemble $FSPT_\alpha$, il nous suffit d'énumérer les différents (k,q)-arbres de partitions étiquetés possibles de A_α. Dans le cas présent, la génération de $FSPT_\alpha$ est triviale. ■

Lemme 5.41. *Soit α un nœud de départ. La Procédure de Départ garantit que $FSPT_\alpha$ est un ensemble complet de (k,q)-arbres de partitions étiquetés de A_α.*

Procédure d'Oubli. Soit α un nœud d'oubli de $\mathcal{T}$ avec un fils β pour lequel nous connaissons $FSPT_\beta$. Le calcul de $FSPT_\alpha$ est encore une fois très simple car il nous suffit de recopier $FSPT_\beta$. En effet, les ensembles A_β et A_α sont identiques, ils ont donc le même ensemble de (k,q)-arbres de partitions étiquetés. ■

Lemme 5.42. *Soit α un nœud d'oubli. La Procédure d'Oubli garantit que $FSPT_\alpha$ est un ensemble complet de (k,q)-arbres de partitions étiquetés de A_α.*

Procédure d'Introduction. Soit α un nœud d'introduction de $\mathcal{T}$ avec un fils β pour lequel nous connaissons $FSPT_\beta$. Nous appelons $\bullet$ l'élément que nous avons ajouté à A_β pour obtenir A_α. Nous obtenons alors $FSPT_\alpha$ à partir de $FSPT_\beta$ en appliquant les trois phases ci-dessous à chacun des (k,q)-arbres de partitions étiquetés de $FSPT_\beta$ et pour chaque exécution possible de la première phase. En d'autres termes, pour chaque $(T_\beta,\ell_\beta) \in FSPT_\beta$ nous regardons les différentes façons d'introduire $\bullet$ dans A_β et nous ajoutons à $FSPT_\alpha$ le nouvel arbre de partitions étiqueté (T_α,ℓ_α) ainsi créé, si sa largeur q-ramifiée suivant ϕ ne dépasse pas k.

▹ Introduction de $\bullet$ dans A_β. Cette opération revient à ajouter une nouvelle feuille dans (T_β, ℓ_β). Pour ce faire nous choisissons un sommet u de T_β et nous lui ajoutons une feuille v. Puis, nous opérons un nombre fini de subdivisons à l'arête reliant u à v, le chemin reliant ces deux sommets est appelé $P(u,v)$. Nous venons de créer T_α. Pour la définition de r_α nous observons deux cas en fonction de si r_β et u désignent le même sommet. Dans le cas où $r_\beta = u$, le sommet r_α peut être n'importe quel sommet de $V(P(u,v)) \setminus \{v\}$; dans l'autre cas, $r_\alpha = r_\beta$.

▹ Mise à jour des étiquettes. Nous commençons par attribuer des étiquettes aux sommets et arêtes nouvellement créés, lesquels sont les sommets internes ainsi que les arêtes de $P(u,v)$. Soit x un tel sommet, l'étiquette que nous associons à x est $\ell_\alpha(x) = \phi_{A_\alpha}(\{A_\beta, \{\bullet\}\})$. Les arêtes reçoivent également cette étiquette, c'est à dire qu'à une arête e de $P(u,v)$ nous associons l'étiquette $\ell_\alpha(e) = \phi_{A_\alpha}(\{A_\beta, \{\bullet\}\})$. Nous passons ensuite à la mise à jour des étiquettes déjà présentes et, pour le moment, associés aux étiquettes de ℓ_β. Donc, pour chaque arête e de T_α qui n'est pas une arête de $P(u,v)$, nous définissons $\ell_\alpha(e) = I_\phi(\ell_\beta(e), \mathcal{Q}_e, \bullet)$, avec $\mathcal{Q}_e$ la bipartition de X_α induite par e. Puis, pour chaque sommet interne x de T_α qui n'est pas un sommet interne de $P(u,v)$, nous définissons $\ell_\alpha(x) = I_\phi(\ell_\beta(x), \mathcal{Q}_x, \bullet)$, avec $\mathcal{Q}_x$ la partition de X_α induite par x.

▹ Mise à jour de $FSPT_\alpha$. Si (T_α, ℓ_α) est q-ramifié et que, pour chaque sommet interne x et chaque arête e de T_α, nous avons $\ell_\alpha(x) \leq k$ et $\ell_\alpha(e) \leq k$, alors nous ajoutons (T_α, ℓ_α) à $FSPT_\alpha$, sinon $FSPT_\alpha$ reste inchangé. ■

Nous allons maintenant prouver que la Procédure d'Introduction est correcte. Pour cela il faut montrer deux choses : premièrement chaque élément de $FSPT_\alpha$ doit être un (k,q)-arbre de partitions étiqueté de A_α et deuxièmement chaque (k,q)-arbre de partitions étiqueté de A_α est un élément de $FSPT_\alpha$.

Lemme 5.43. *Soit α un nœud d'introduction. La Procédure d'Introduction garantit que chaque élément de $FSPT_\alpha$ est un (k,q)-arbre de partitions étiqueté de A_α.*

Démonstration. Soit (T_α, ℓ_α) un élément de $FSPT_\alpha$, calculé par la Procédure d'Introduction. Soit $FSPT_\beta$, l'ensemble des (k,q)-arbres de partitions étiquetés de A_β. Prenons (T_β, ℓ_β) un élément quelconque de $FSPT_\beta$ et considérons que (T_α, ℓ_α) est obtenu après une exécution de la Procédure d'Introduction sur (T_β, ℓ_β).

En considérant la première phase de la Procédure d'Introduction, nous avons que T_β est un arbre de partition de A_β et que $A_\alpha \setminus A_\beta = \{\bullet\}$. Par conséquent, ajouter un sommet v à un sommet interne u de T_β et mettre $\bullet$ en bijection avec v résulte bien en un arbre de partitions pour A_α. De plus, les possibles subdivisons de l'arête uv ne changent rien au fait que T_α soit un arbre de partitions pour A_α.

Nous passons maintenant à la deuxième phase. Il nous faut montrer que les étiquettes attribuées aux sommets internes ainsi qu'aux arêtes de T_α sont correctes. La preuve étant la même pour les sommets et les arêtes nous ne traiterons ici que le cas des sommets. Donc, pour chaque sommet interne x de T_α, nous considérons la partition $\mathcal{P}_x$ de A_α induite par x. Étant donné que $FSPT_\beta$ contient tous les (k,q)-arbres de partitions étiquetés de A_β, nous avons que $\ell_\beta(x) = \phi_{A_\beta}(\mathcal{P}_x \cap A_\beta)$. Par conséquent, si x est un sommet interne de T_α mais n'est pas un sommet interne de $P(u,v)$, alors la première règle de

compatibilité entre ϕ et $(\mathcal{T},\mathcal{X})$ nous donne le résultat suivant :

$$\ell_\alpha(x) = I_\phi(\ell_\beta(x), \mathcal{Q}_x, \bullet) = I_\phi(\phi_{A_\beta}(\mathcal{P}_x \cap A_\beta), \mathcal{P}_x \cap X_\alpha, \bullet) = \phi_{A_\alpha}(\mathcal{P}_x).$$

Et, si x est un sommet interne de $P(u,v)$, alors l'étiquette $\ell_\alpha(x) = \phi_{A_\alpha}(\{A_\beta, \{\bullet\}\}) = \phi_{A_\alpha}(\mathcal{P}_x)$ lui a été attribuée lors de la deuxième phase.

La troisième phase de la procédure résoud la question de la q-ramification et de la largeur suivant ϕ étant donné que $FSPT_\alpha$ n'est mis à jour que si T_α a une largeur q-ramifiée suivant ϕ inférieure ou égale à k. □

Nous passons maintenant à la seconde partie de la vérification de la Procédure d'Introduction. Comme nous l'avons déjà fait remarquer, le schéma de la preuve qui va suivre va être régulièrement réutilisé par la suite. Nous proposons alors un aperçu rapide de ce schéma. Ici nous voulons montrer que, pour tout (k,q)-arbre de partitions T_α de A_α, il existe une exécution de la Procédure d'Introduction qui, à partir d'un élément $(T_\beta, \ell_\beta) \in FSPT_\beta$ calcule un (k,q)-arbre de partitions étiqueté $(T_\delta, \ell_\delta) \in FSPT_\alpha$ de A_α tel que $T_\alpha \cong T_\delta$. D'après le lemme précédent, la fonction d'étiquetage ℓ_δ produite par la Procédure d'Introduction est correcte, il s'en suit que $(T_\alpha, \ell_\delta) \in FSPT_\alpha$.

Lemme 5.44. *Soit α un nœud d'introduction. La Procédure d'Introduction garantit que chaque (k,q)-arbre de partitions étiqueté de A_α est un élément de $FSPT_\alpha$.*

Démonstration. Soit T_α un (k,q)-arbre de partitions de A_α. Soit T_β l'arbre de partitions correspondant à T_α restreint à A_β. Comme ϕ est monotone et que la restriction de T_α à A_β ne peut que supprimer des ramifications, l'arbre de partitions T_β a une largeur q-ramifiée suivant ϕ au plus égale à k. Donc il existe un élément (T_β, ℓ_β) dans $FSPT_\beta$.

Soit v le sommet de T_α correspondant à l'élément $\bullet$ introduit. Étant donné que T_β est un sous-arbre de T_α, nous pouvons choisir le sommet u correspondant au sommet de T_β le plus proche de v dans T_α. Le sous-chemin de T_α reliant ces deux sommets est noté $P(u,v)$. De plus, $A_\beta = A_\alpha \setminus \{\bullet\}$, ce qui implique que les sommets internes de $P(u,v)$ sont de degré 2 dans T_α. Par conséquent, il existe une exécution de la Procédure d'Introduction dont la première étape calcule un arbre T_δ, isomorphe à T_α, en attachant le sommet v au sommet u dans T_β puis en subdivisant $|V(P(u,v)) \setminus \{u,v\}|$ fois l'arête uv nouvellement créée. Le choix de la racine r_δ est determiné par la position de r_α dans T_α.

Pour conclure nous rappelons que, d'après le Lemme 5.43, la fonction d'étiquetage ℓ_δ produite par la Procédure d'Introduction est correcte et donc (T_α, ℓ_δ) est un élément de $FSPT_\alpha$. □

Corollaire 5.45. *Soit α un nœud d'introduction. La Procédure d'Introduction garantit que $FSPT_\alpha$ est un ensemble complet de (k,q)-arbres de partitions étiquetés de A_α.*

Procédure de Jonction. Soit α un nœud de jonction de $\mathcal{T}$ avec deux fils β et γ pour lesquels nous connaissons $FSPT_\beta$ et $FSPT_\gamma$. La procédure que nous décrivons construit l'ensemble des arbres de partitions issus de la fusions des éléments de $FSPT_\beta$ avec les éléments dits « compatibles » de $FSPT_\gamma$, après quoi nous ne garderons que les arbres de partitions dont la largeur q-ramifiée suivant ϕ ne dépasse pas k.

Soient (T_β, ℓ_β) et (T_γ, ℓ_γ) deux éléments pris dans $FSPT_\beta$ et $FSPT_\gamma$, respectivement. Rappelons que, par définition, $X_\alpha = X_\beta = X_\gamma$. Nous considérons alors T_β^r et T_γ^r les restrictions respectives de T_β et T_γ à X_α. Les arbres de partitions (T_β, ℓ_β) et (T_γ, ℓ_γ) sont considérés comme étant *compatibles* si $T_\beta^r \cong T_\gamma^r$.

▹ Fusion de T_β et T_γ. L'arbre de partition T_α est obtenu à partir de T_β et T_γ en identifiant les sommets de T_β^r avec leur correspondant dans T_γ^r, puis en identifiant les arêtes de chaque couple d'arêtes doubles ainsi produits. La racine r_α de T_α est choisie arbitrairement parmis les sommets internes de T_α.

▹ Mise à jours des étiquettes. Soit T_α^r la restriction de T_α à X_α. Soit x un sommet interne de T_α^r et soit $\mathcal{Q}_x$ la partition de X_α définie par x. Nous appelons y et z les sommets respectifs de T_β^r et T_γ^r qui ont été identifiés afin d'obtenir x. L'étiquette $\ell_\alpha(x)$ attribuée à x prend alors pour valeur $J_\phi(\ell_\beta(y), \ell_\gamma(z), \mathcal{Q}_x)$. Il en va de même pour chaque arête e de T_α^r, c'est à dire que nous attribuons à e l'étiquette $\ell_\alpha(e) = J_\phi(\ell_\beta(f), \ell_\gamma(g), \mathcal{Q}_e)$, où f et g sont les arêtes respectives de T_β^r et T_γ^r qui ont été identifiées afin d'obtenir e et $\mathcal{Q}_e$ est la bipartition de X_α définie par e. Pour chaque autre sommet interne x de T_α nous attribuons l'étiquette $\ell_\alpha(x)$ égale à $\ell_\beta(x)$ si x est un sommet de T_β ou à $\ell_\gamma(x)$ si x est un sommet de T_γ. Même procédure pour chaque autre arête e de T_α : $\ell_\alpha(e)$ est égale à $\ell_\beta(e)$ si e est une arête de T_β ou à $\ell_\gamma(e)$ si e est une arête de T_γ.

▹ Mise à jours de $FSPT_\alpha$. Si (T_α, ℓ_α) est q-ramifié et que, pour chaque sommet interne x et chaque arête e de T_α, nous avons $\ell_\alpha(x) \leq k$ et $\ell_\alpha(e) \leq k$, alors nous ajoutons (T_α, ℓ_α) à $FSPT_\alpha$, sinon $FSPT_\alpha$ reste inchangé. ■

Nous allons maintenant prouver que la Procédure de Jonction est correcte. Pour cela nous réutilisons le schéma de preuve utilisé pour la Procédure d'Introduction.

Lemme 5.46. *Soit α un nœud de jonction. La Procédure de Jonction garantit que chaque élément de $FSPT_\alpha$ est un (k,q)-arbre de partitions étiqueté de A_α.*

Démonstration. Soit (T_β, ℓ_β) un élément de $FSPT_\beta$ et soit (T_γ, ℓ_γ) un élément de $FSPT_\gamma$ compatible avec (T_β, ℓ_β). Si T_β^r et T_γ^r sont les restrictions respectives de T_β et T_γ à X_α, alors $T_\beta^r \cong T_\gamma^r$.

En observant la première phase de la Procédure de Jonction, nous constatons que $A_\alpha = A_\beta \cup A_\gamma$ et que $(A_\beta \setminus X_\beta) \cap (A_\gamma \setminus X_\gamma) = \emptyset$ sans oublier que $X_\alpha = X_\beta = X_\gamma$. De ce fait, l'arbre de partitions T_δ, obtenu en fusionnant T_β avec T_γ, définit bien un arbre de partitions pour A_α.

Nous devons aussi nous assurer qu'à la fin de la deuxième phase de la Procédure de Jonction, toutes les étiquettes sont correctes, c'est-à-dire que, pour chaque sommet interne x de T_α, nous avons $\ell_\alpha(x) = \phi_{A_\alpha}(\mathcal{P}_x)$, où $\mathcal{P}_x$ est la partition de A_α induite par x. Étant donné que (T_β, ℓ_β) est un élément de $FSPT_\beta$, nous savons que $\ell_\beta(x) = \phi_{A_\beta}(\mathcal{P}_x \cap A_\beta)$, si x est un sommet interne de T_β. Dans le cas où x est un sommet interne de T_γ, nous avons $\ell_\gamma(x) = \phi_{A_\gamma}(\mathcal{P}_x \cap A_\gamma)$, car (T_γ, ℓ_γ) est un élément de $FSPT_\gamma$. Nous observons alors trois cas. Dans le premier cas x est un sommet interne de T_α^r et, d'après la deuxième règle de compatibilité de ϕ avec $(\mathcal{T}, \mathcal{X})$, nous avons :

$$\ell_\alpha(x) = J_\phi(\ell_\beta(x), \ell_\gamma(x), \mathcal{Q}_x) = J_\phi(\phi_{A_\beta}(\mathcal{P}_x \cap A_\beta), \phi_{A_\gamma}(\mathcal{P}_x \cap A_\gamma), \mathcal{P}_x \cap X_\alpha) = \phi_{A_\alpha}(\mathcal{P}_x).$$

Dans le deuxième cas x est un sommet interne de T_β mais pas de T_α^r, par conséquent, $\mathcal{P}_x \cap A_\gamma$ contient une unique partition $\{A_\gamma\}$ et la deuxième règle de compatibilité de ϕ avec $(\mathcal{T}, \mathcal{X})$ nous dit que :

$$\ell_\alpha(x) = \ell_\beta(x) = \phi_{A_\beta}(\mathcal{P}_x \cap A_\beta) = \phi_{A_\alpha}(\mathcal{P}_x).$$

Dans le troisième cas x est un sommet interne de T_γ mais pas de T_α^r et nous ré-utilisons l'argument précédent pour obtenir :

$$\ell_\alpha(x) = \ell_\gamma(x) = \phi_{A_\gamma}(\mathcal{P}_x \cap A_\gamma) = \phi_{A_\alpha}(\mathcal{P}_x).$$

Il nous faut encore montrer que la largeur q-ramifiée suivant ϕ de T_α ne dépasse pas k. La dernière phase de la Procédure de Jonction ajoute (T_α, ℓ_α) à $FSPT_\alpha$ seulement si les deux conditions précitées sont vérifiées. □

Nous passons maintenant à la seconde partie de la vérification de la Procédure de Jonction, pour laquelle nous donnons un aperçu du schéma de preuve. Nous voulons prouver que pour tout (k,q)-arbre de partitions T_α de A_α, il existe une exécution de la Procédure de Jonction qui, à partir d'un élément (T_β, ℓ_β) de $FSPT_\beta$ et d'un élément (T_γ, ℓ_γ) de $FSPT_\gamma$, calcule un (k,q)-arbre de partitions étiqueté (T_δ, ℓ_δ) de A_α tel que $T_\alpha \cong T_\delta$. D'après le lemme précédent, la fonction d'étiquetage ℓ_δ produite par la Procédure de Jonction est correcte, il s'en suit que (T_α, ℓ_δ) est un élément de $FSPT_\alpha$.

Lemme 5.47. *Soit α un nœud de jonction. La Procédure de Jonction garantit que chaque (k,q)-arbre de partitions étiqueté de A_α est un élément de $FSPT_\alpha$.*

Démonstration. Soit T_α un (k,q)-arbre de partitions de A_α. Nous considérons T_β la restriction de T_α à A_β et T_γ la restriction de T_α à A_γ. Les deux arbres de partitions ainsi obtenus ont, par définition, une largeur q-ramifiée suivant ϕ qui ne dépasse pas k, il existe donc des étiquetages de ces arbres tels que (T_β, ℓ_β) est un élément de $FSPT_\beta$ et (T_γ, ℓ_γ) est un élément de $FSPT_\gamma$.

Soit T_β^r la restriction de T_β à X_α et T_γ^r la restriction de T_γ à X_α. Rapellons que $X_\alpha = X_\beta = X_\gamma$, il en découle que $T_\beta^r \cong T_\gamma^r$. Nous pouvons, de ce fait, appliquer la Procédure de Jonction à (T_β, ℓ_β) et (T_γ, ℓ_γ). Soit (T_δ, ℓ_δ) l'arbre de partitions étiqueté que nous obtenons. Le fait que $T_\alpha \cong T_\delta$ est assez évident. Comme la Procédure de Jonction choisit r_δ parmis tous les sommets internes de T_δ, il existe donc une exécution de cette procédure qui choisit r_δ en accord avec r_α. Nous avons montré dans le Lemme 5.46 que la Procédure de Jonction calcule une fonction d'étiquetage correcte et, par conséquent, (T_α, ℓ_δ) est un élément de $FSPT_\alpha$. □

Corollaire 5.48. *Soit α un nœud de jonction. La Procédure de Jonction garantit que $FSPT_\alpha$ est un ensemble complet de (k,q)-arbres de partitions étiquetés de A_α.*

Les quatres types de nœuds de notre jolie décomposition $(\mathcal{T}, \mathcal{X})$ ont été traités et nous avons maintenant un algorithme capable de décider si A à une largeur q-ramifiée suivant ϕ inférieure ou égale à k. Cependant cet algorithme n'est pas utilisable en l'état, étant

donné qu'il nécessite une grande quantité mémoire (au moins exponentielle). Dans la section suivante, nous décrivons une méthode qui va nous permettre, pour chaque nœud α de $\mathcal{T}$, de majorer, en fonction de k, q, et $|X_a|$, le nombre ainsi que la taille des arbres de partitions à retenir au niveau de α. La méthode que nous utilisons est une adaptation de celle décrite par Bodlaender et Kloks dans [5]. Puis dans une nouvelle section, nous montrerons comment combiner cette méthode de compression à l'algorithme que nous venons de décrire.

5.3. Outils de compression

Comme nous venons de le voir dans la section précédente, étant donné un nœud α de $(\mathcal{J},\mathcal{Y})$, le nombre d'éléments présents dans $FSPT_\alpha$ est très grand. Les outils de compression que nous introduisons dans cette section vont nous permettre de définir un ensemble FSC_α tel que le nombre d'éléments que FSC_α contient est majoré en fonction de k, q et $|X_\alpha|$ et que la taille d'un élément donné de FSC_α est également majorée en fonction de k, q et $|X_\alpha|$. De plus chaque élément de $FSPT_\alpha$ doit avoir un représentant dans FSC_α et chaque élément de FSC_α doit représenter au moins un élément de $FSPT_\alpha$. De ce fait, $FSPT_\alpha$ est vide si et seulement si FSC_α l'est aussi. Et donc, la largeur q-ramifiée suivant ϕ d'un ensemble A est au plus k si et seulement si FSC_ρ n'est pas vide, avec ρ la racine de la jolie décomposition associée à A.

La compression que nous présentons s'applique donc aux éléments de $FSPT_\alpha$, lesquels sont des (k,q)-arbres de partitions de A_α. Comme nous l'avons évoqué plus haut, cette compression reprend les grandes lignes de celle définie par Bodlaender et Kloks [5] et elle peut se résumer ainsi. Étant donné un (k,q)-arbre de partitions (T,ℓ) de A_α, prendre un sous-arbre T^r de T induisant un arbre de partition de X_α, puis faire l'inventaire des sommets « importants » de T^r enfin, pour chaque sous-chemin P reliant deux sommets importants de T^r et tel que les sommets internes de P ne sont pas importants, contracter les arêtes « sans importance ».

Une première observation concernant la compression de (T,ℓ) est que le nombre de feuilles de T^r est au plus $|X_\alpha|$; il en va de même pour les sommets de T^r de degré sortant au moins 2. En considérant que la quantité de sommets importants à retenir est majorée par k et q, il nous reste alors à majorer la longeur des sous-chemins reliant ces sommets importants. C'est la première étape de la présentation de nos outils de compression, dans laquelle nous définissons quelles sont les arêtes que nous pouvons contracter. Nous définissons ensuite un certain nombre d'opérations pour la manipulation des chemins dont nous avons contracté certaines arêtes.

5.3.1. Chemins étiquetés

Les chemins avec lesquels nous travaillons sont des sous-chemins d'arbres de partitions étiquetés, ils sont donc eux aussi sujets à un étiquetage. Les différentes opérations que nous sommes sur le point de définir vont nous permettre de supprimer de ces chemins un certain nombre de sommets et d'arêtes en fonction des étiquettes qui leurs sont attribuées. Ces différentes suppressions, qui sont en fait des contractions d'arêtes,

réduisent la longeur des chemins concernés de façon à pouvoir majorer leurs longueurs en fonction de la plus grande valeur qu'une étiquette peut prendre.

Définition 5.49 (Chemin étiqueté)**.** Un *chemin étiqueté* P est un chemin $(v_0,\dots,v_n)$, où chaque sommet v_i est étiqueté avec un entier $\ell(v_i)$, et chaque arête $e_i = v_{i-1}v_i$ est étiquetée avec un entier $\ell(e_i)$ tel que $\ell(e_i) \leq \ell(v_{i-1})$ et $\ell(e_i) \leq \ell(v_i)$.

Au sein d'un chemin étiqueté P, le plus grand entier qui étiquette un sommet de P est noté $\max(P)$, de façon similaire $\min(P)$ désigne le plus petit entier qui étiquette une arête de P. Notons que la monotonie de ϕ implique que l'étiquette d'une arête ne peut pas être supérieure aux étiquettes des sommets qui lui sont incidents.

Voici maintenant l'opération dans laquelle réside toute la puissance de l'algorithme que nous cherchons à construire : la contraction d'un chemin étiqueté.

Définition 5.50 (Contraction d'un chemin étiqueté)**.** La *contraction* d'un chemin étiqueté $P = (v_0,\dots,v_n)$, notée $\tau(P)$, est obtenue à partir de P en opérant les contractions d'arêtes définies par les deux opérations suivantes, et ce jusqu'à ce qu'il n'y ait plus de contraction possible.

1. S'il existe $0 < i < j \leq n$ tels que, pout tout $i < k \leq j$, nous avons $\ell(e_i) \leq \ell(e_k) \leq \ell(v_j)$ et, pour tout $i \leq k < j$, nous avons $\ell(e_i) \leq \ell(v_k) \leq \ell(v_j)$, alors nous contractons les arêtes e_{i+1} à e_j en conservant le sommet v_j.
2. S'il existe $0 \leq i < j \leq n$, avec $j - i \geq 2$ et tels que, pout tout $i < k < j$, nous avons $\ell(v_i) \geq \ell(e_k) \geq \ell(e_j)$ et $\ell(v_i) \geq \ell(v_k) \geq \ell(e_j)$, alors nous contractons les arêtes e_{i+1} à e_{j-1}, en conservant le sommet v_i.

L'ordre des contractions que nous opérons n'a pas d'importance, c'est pourquoi $\tau(P)$ est uniquement définit, pour tout chemin P donné.

Le chemin étiqueté $\tau(P)$ que nous obtenons est tout à fait comparable aux séquences caractéristiques définies par Bodlaender et Kloks, dont nous rappelons la définition. Soit $S = (s_0,\dots,s_n)$ une séquence d'entiers. La *séquence caractéristique* de S, notée $\tau(S)$, est obtenue en itérant autant que possible l'opération suivante. S'il existe deux éléments s_i et s_j, avec $j - i \geq 2$ tels que, pour tout $i < k < j$, nous avons soit $s_i \leq s_k \leq s_j$ soit $s_i \geq s_k \geq s_j$, alors nous supprimons de S les éléments s_{i+1} à s_{j-1}. Nous considérons maintenant $P = (v_0,\dots,v_n)$ un chemin étiqueté, et nous définissons la séquence d'entiers $S = (\ell(e_1),\ell(v_1),\ell(e_2),\dots,\ell(e_n))$ comme étant la séquence *induite* par P. Ce qui nous permet d'établir que la séquence d'entiers $S' = (\ell(e'_1),\ell(v'_1),\ell(e'_2),\dots,\ell(e'_m))$ induite par $\tau(P) = (v'_0,\dots,v'_m)$, avec $v'_0 = v_0$ et $v'_m = v_n$, est égale à la séquence $\tau(S)$. Notons qu'il est possible que dans P nous ayons $\ell(v_0) = \ell(e_1)$ et $\ell(e_n) = \ell(v_n)$, et que dans $\tau(P)$ nous ayons $\ell(v'_0) = \ell(e'_1)$ et $\ell(e'_n) = \ell(v'_n)$.

Nous abordons maintenant la manipulation des contractions de chemins étiquetés afin d'en tirer quelques propriétés qui nous seront utiles par la suite. La première d'entre elles se déduit directement de la définition de la contraction.

Propriété 5.51. *Soit P un chemin étiqueté, alors* $\min(\tau(P)) = \min(P)$ *et* $\max(\tau(P)) = \max(P)$.

La seconde propriété est celle qui nous permet de majorer la taille de la contraction d'un chemin P en fonction de $\max(P)$.

Propriété 5.52. *Soit P un chemin étiqueté. Alors la contraction $\tau(P)$ de P contient au plus* $\max(P)+1$ *arêtes et* $\max(P)$ *sommets internes.*

Démonstration. Dans un premier temps, nous commençons par majorer le nombre de sommets de $\tau(P)$ étiquetés avec $\max(P)$ ainsi que le nombre d'arêtes de $\tau(P)$ étiquetées avec $\min(P)$. Puis nous définissons une borne sur la taille de $\tau(P)$ quand l'une des extrémités de P représente l'unique sommet étiqueté avec $\max(P)$. Enfin nous combinons ces deux résultats afin de conclure la preuve de la propriété.

Majorant sur les sommets et arêtes « extrêmes » de $\tau(P)$. Soit $P = (v_0,\dots,v_n)$ un chemin étiqueté. Soit v_f le premier sommet de P tel que $\ell(v_f) = \max(P)$ et soit v_l le dernier sommet de P tel que $\tau(v_l) = \max(P)$. En considérant e_m une arête de P telle que $\ell(e_m) = \min(P)$, alors :
- si $m \leq f$, nous contractons P entre e_m et v_l ;
- si $f < m \leq l$, nous contractons P entre v_f et e_m ainsi qu'entre e_m et v_l ;
- si $l < m$, nous contractons P entre v_f et e_m.

Par conséquent, $\tau(P)$ admet au plus deux sommets $\dot{v}_i$ et $\dot{v}_j$ tels que $\ell(\dot{v}_i) = \ell(\dot{v}_j) = \max(P)$; auquel cas nous avons $|i-j| = 1$.

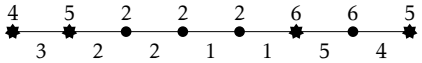
Avec le même raisonnement, en considérant e_f la première arête de P telle que $\ell(e_f) = \min(P)$, et e_l la dernière arête de P telle que $\ell(e_l) = \min(P)$, ainsi que v_M un sommet de P tel que $\ell(v_M) = \max(P)$, alors :
- si $M \leq f$, nous contractons P entre v_M et e_l ;
- si $f < M \leq l$, nous contractons P entre e_f et v_M ainsi qu'entre v_M et e_l ;
- si $l < M$, nous contractons P entre e_f et v_M.

Ce qui implique que $\tau(P)$ admet au plus deux arêtes $\dot{e}_i$ et $\dot{e}_j$ telles que $\ell(\dot{e}_i) = \ell(\dot{e}_j) = \min(P)$; auquel cas nous avons $|i-j| = 1$.

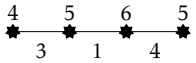
En combinant ce dernier résultat avec le précédent, nous pouvons conclure que si $\tau(P)$ admet deux sommets $\dot{v}_{i-1}$ et $\dot{v}_i$ tels que $\ell(\dot{v}_{i-1}) = \ell(\dot{v}_i) = \max(P)$, alors $\dot{e}_i$ est l'unique arête de $\tau(P)$ telle que $\ell(\dot{e}_i) = \min(P)$. Et, dans le cas où $\tau(P)$ admet deux arêtes $\dot{e}_i$ et $\dot{e}_{i+1}$ telles que $\ell(\dot{e}_i) = \ell(\dot{e}_{i+1}) = \min(P)$, alors $\dot{v}_i$ est l'unique sommet de $\tau(P)$ tel que $\ell(\dot{v}_i) = \max(P)$.

Majorant sur la longueur de $\tau(P)$. Soit $P = (v_0,\dots,v_n)$ un chemin étiqueté, dans lequel v_0 est l'unique sommet tel que $\ell(v_0) = \max(P)$, et soit $\tau(P)$ la contraction de P. Par définition nous avons que le premier sommet $\dot{v}_0$ de $\tau(P)$ correspond à v_0, que la première arête $\dot{e}_1$ de $\tau(P)$ correspond à la dernière arête e_i de P telle que $\ell(e_i) = \min(P)$ et que le second sommet $\dot{v}_1$ de $\tau(P)$ correspond au dernier sommet v_j de P tel que $i \leq j$ et $\ell(v_j) = \max(P')$, avec P' le sous-chemin de P reliant v_j à v_n. Nous pouvons ensuite réappliquer récursivement le même raisonnement au chemin P'.

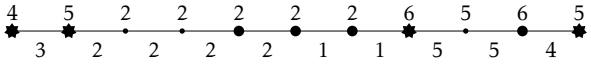
Ceci implique alors que, dans $\tau(P) = (\dot{v}_0,\dots,\dot{v}_m)$, nous avons $\ell(\dot{v}_i) < \ell(\dot{v}_{i-1}) \leq \max(P)$, pour tout $1 \leq i \leq m$, ainsi que $\ell(\dot{e}_i) > \ell(\dot{e}_{i-1}) \geq \min(P)$, pour tout $1 < i \leq m$. En particulier, nous avons $\ell(\dot{v}_m) \leq \max(P) - m$ et $\ell(\dot{e}_m) \geq m - 1 + \min(P)$. Si nous ajoutons à cela que $\ell(\dot{e}_m) \leq \ell(\dot{v}_m)$, nous obtenons $m \leq \frac{1}{2}(\max(P) + 1 - \min(P))$.



(a) Un chemin étiqueté P.



(b) Le chemin $\tau(P)$.



(c) Une extension de P.

FIGURE 5.1. : Illustration des opérations de base sur un chemin étiqueté.

Combinaison des deux majorants. Premièrement, si $\tau(P) = (\dot{v}_0, \ldots, \dot{v}_m)$ contient un unique sommet $\dot{v}_i$ étiqueté avec $\max(P)$, alors nous séparons $\tau(P)$ en deux sous-chemins $L = (\dot{v}_0, \ldots, \dot{v}_i)$ et $R = (\dot{v}_i, \ldots, \dot{v}_m)$ respectants les conditions du second argument. Nous avons donc $i \leq \frac{1}{2}(\max(L) + 1 - \min(L))$ et $m - i \leq \frac{1}{2}(\max(R) + 1 - \min(R))$. En reportant dans ces deux inégalités le fait que $\max(L) = \max(R) = \max(P)$ et que $\min(L) \geq 0$ et $\min(R) \geq 0$, nous obtenons $m \leq \max(P) + 1$.

Dans le cas où $\tau(P)$ admet deux sommets $\dot{v}_i$ et $\dot{v}_{i+1}$ étiquetés avec $\max(P)$, alors nous séparons $\tau(P)$ en deux sous-chemins $L = (\dot{v}_0, \ldots, \dot{v}_i)$ et $R = (\dot{v}_{i+1}, \ldots, \dot{v}_m)$ respectants les conditions du second argument. Cette fois-ci nous obtenons $i \leq \frac{1}{2}(\max(L) + 1 - \min(L))$ et $m - (i+1) \leq \frac{1}{2}(\max(R) + 1 - \min(R))$, avec $\max(L) = \max(R) = \max(P)$ et $\min(L) \geq 1$ et $\min(R) \geq 1$. Nous permettant encore une fois d'écrire $m \leq \max(P) + 1$.

Finalement, $\tau(P)$ contient $\max(P) + 2$ sommets indexés de $\dot{v}_0$ à $\dot{v}_{\max(P)+1}$, ce qui implique bien que $\tau(P)$ est composé d'au plus $\max(P) + 1$ arêtes et $\max(P)$ sommets internes. □

L'opération suivante est, en quelques sortes, l'opération inverse de la contraction et consiste à subdiviser certaines arêtes d'un chemin étiqueté.

Définition 5.53 (Extension d'un chemin étiqueté)**.** Une *extension* d'un chemin étiqueté P correspond à n'importe quel chemin résultant d'un nombre fini de subdivisions de certaines arêtes de P. Soit uv une arête de P sur laquelle nous opérons une subdivision, nous obtenons alors deux arêtes uw et wv. Les étiquettes associées au nouveau sommet ainsi qu'aux nouvelles arêtes sont définies par $\ell(uw) = \ell(w) = \ell(wv) := \ell(uv)$.

La FIGURE 5.1 nous donne un exemple d'un chemin étiqueté avec sa contraction ainsi qu'une extension possible.

Propriété 5.54. *Si $\bar{P}$ est une extension d'un chemin étiqueté P, alors $\tau(\bar{P}) = \tau(P)$.*

Nous pouvons alors définir une relation d'ordre sur les chemins étiquetés.

Définition 5.55 (Comparaison de chemins étiquetés)**.** Soient P et Q deux chemins étiquetés. Nous écrivons $P \preccurlyeq Q$, s'il existe des extensions $\bar{P} = (\bar{u}_0, \ldots, \bar{u}_t)$ et $\bar{Q} = (\bar{v}_0, \ldots, \bar{v}_t)$ de même longueur et telles que, pour tout $0 \leq i \leq t$, nous avons $\ell(\bar{u}_i) \leq \ell(\bar{v}_i)$ et, pour tout $0 < i \leq t$, nous avons $\ell(\bar{u}_{i-1}\bar{u}_i) \leq \ell(\bar{v}_{i-1}\bar{v}_i)$.

5.3.2. Caractéristique

Nous présentons ici la contraction d'arbres de partitions dont nous avons préalablement parlé. La structure de données que nous sommes sur le point de définir, et dont la taille est majorée en fonction de k, q et $|X_\alpha|$, va nous permettre d'encoder l'ensemble [2] des (k,q)-arbres de partitions de A_α, et ce en ne considérant que les éléments présents dans X_α. En plus de la contraction d'arbre de partitions dont nous avons déjà discuté, nous faisons appel à un certain nombre de variables afin de garder en mémoire les différentes ramifications de T_α qui n'apparaissent plus dans sa contraction.

Définition 5.56 (Caractéristique d'un arbre de partition)**.** Soit T un (k,q)-arbre de partitions d'un ensemble A et soit X un sous-ensemble de A. La *caractéristique* de T restreint à X, notée $\tau_X(T)$, est un 6-uplet qui contient les éléments suivants :
- T^c, un arbre de partition de X ;
- $\ell : V(cp(T^c)) \cup E(T^c) \to \mathbb{N}$, une fonction d'étiquetage des sommets internes et des arêtes de T^c qui nous permet de calculer la largeur suivant ϕ de T ;
- S, un sous-ensemble de sommets de T^c dont il est important de suivre l'évolution ;
- a, un entier qui compte le nombre de ramifications de $cp(T)$ se situant entre $r(T)$ et $r(T^c)$ dans T ;
- $d : V(cp(T^c)) \to \mathbb{N}$, une fonction qui, pour chaque sommet interne x de T^c, nous permet de connaître la hauteur de ramification du sous-arbre de T enraciné en x ;
- $b : V(cp(T^c)) \to \{0,1,2\}$, une fonction qui, pour chaque sommet interne x de T^c, nous permet de retenir une borne inferieur sur le nombre de ramifications de x dans T.

Soit P un sous-chemin de T^c reliant deux éléments de S et tel qu'aucun autre élément de S ne fait partie des sommets internes de P. Nous donnons alors à P le nom de *tronçon* de T^c.

Voici maintenant la Procédure de Caractérisation permetant de calculer la caractéristique d'un arbre de partition T restreint à un ensemble X. Nous profitons de cette définition pour donner un peu plus détails concernant les différents éléments liés à cette caractéristique.

Procédure de Caractérisation. Soit T^r la restriction de T à X et soient r^r et r leurs racines respectives.

▹ Définition de S. L'ensemble S regroupe les sommets de T^r qui sont soit les pères des feuilles de T^r, soit des sommets de ramification dans $cp(T)$. Notons que r^r, est soit le père d'une feuille de T^r, soit un sommet ramification de $cp(T)$.

Le but de S est de définir les tronçons de T^c auxquels nous allons pouvoir appliquer des contractions. La façon dont les sommets de S ont été choisis est telle que tous les sommets internes ainsi que les arêtes d'un tronçon donné induisent la même bipartition de X. La seule information qui différencie ces éléments est contenue dans les étiquettes représentant la largeur suivant ϕ de chacun d'eux dans T.

2. Dans les faits, nous ne considérons pas tous les (k,q)-arbres de partitions de A_α, cependant nous définissons une relation d'ordre permetant de ne garder que les « plus petits » d'entre eux.

▹ Définition de ℓ et de T^c. L'arbre de partitions T^c est obtenu après contraction d'un certains nombre de tronçons de T^r et, afin de procéder à ces contractions, il faut au préalable que les sommets internes ainsi que les arêtes de T^r soient étiquetés. Nous étendons donc le domaine de définition de ℓ et nous considérons la fonction $\ell' : V(cp(T^r)) \cup E(T^r) \to \mathbb{N}$, telle que, pour tout sommet interne x et toute arête e de T^r, nous avons $\ell'(x) = \phi_A(\mathcal{P}_x)$ et $\ell'(e) = \phi_A(\mathcal{P}_e)$, où $\mathcal{P}_x$ et $\mathcal{P}_e$ dénotent les partitions de A respectivement induites par x et e. Puis nous remplaçons chaque tronçon P de T^r par sa contraction $\tau(P)$, et nous obtenons ainsi l'arbre T^c. Nous définissons ensuite r^c comme étant le correspondant de r^r dans T^c (ce sommet existe car r^r est un sommet de S). Finalement la fonction ℓ correspond à la fonction ℓ' dont nous avons réduit le domaine de définition aux sommets internes et arêtes de T^c.

▹ Définition de a. Les sommets du sous-chemin P de T reliant r au père de r^c dans T n'ont pas de correspondant dans T^c, et parmis ces sommets il y en a un certain nombre qui peuvent être des sommets de ramification dans T. Afin de vérifier que la hauteur de ramification de T est bien majorée par q, il nous faut un moyen de retenir le nombre de sommets de ramification ancêtres de r^c dans T, et qui ont été « oubliés » dans T^c. C'est le rôle que nous donnons à l'entier $a := \mathrm{dr}_T(r, r^c)$.

▹ Définition de d. Au même titre que a compte les « ramifications ancestrales oubliées », la fonction d nous permet de garder une trace des « ramifications descendantes oubliées », toujours afin de vérifier que la hauteur de ramification de T est bien majorée par q. Dans un language plus formel, à chaque sommet interne x de T^c, nous associons la valeur $d(x) = \mathrm{hr}_T(x)$ qui, rappelons le, est la hauteur de ramification du sous-arbre de T enraciné en x.

▹ Définition de b. Soit x un sommet de $cp(T^c)$. Comme T^c est obtenu à partir d'un sous-arbre de T, il se peut qu'un certain nombre de ramifications de x n'apparaissent pas dans T^c. Et, si nous nous plaçons dans le cas où $\mathrm{nr}_{T^c}(x) < 2 \leq \mathrm{nr}_T(x)$, il est alors possible d'ajouter une ramification à x sans pour autant s'inquiter de l'augmentation de la hauteur de ramification de T. De façon similaire si $\mathrm{nr}_{T^c}(x) = 0$, mais que $\mathrm{nr}_T(x) = 1$, alors en ajoutant une ramification à x nous créons un nouveaux sommet de ramification dans $cp(T)$, ce qui n'est pas le cas dans $cp(T^c)$, mansquant de ce fait une potentielle augmentation de la hauteur de ramification de T. C'est pourquoi il est important de garder en mémoire le nombre de ramifications induites par x dans $cp(T)$. Nous associons donc, à chaque sommet interne x de T^c, la valeur $b(x) = \min\{2, \mathrm{nr}_T(x)\}$. ■

Remarque 5.57. D'après la construction de S, nous savons que les sommets de $cp(T^c)$ qui ne sont pas dans S ont exactement une ramification. La hauteur de ramification d'un tel sommet est donc égale à celle de son unique fils qui n'est pas une feuille. De plus, nous verrons pas la suite qu'il n'est pas possible d'ajouter une ramification à un sommet ne se trouvant pas dans S.

Remarque 5.58. Si nous sommes dans le cas où $q = \infty$, alors nous n'avons pas besoin de compter les sommets de ramifications de T. Par conséquent les variables a, d et b sont inutiles, et S regroupe les sommets de T qui sont soit les pères des feuilles de T^r, soit les sommets de ramification de $cp(T^r)$.

Maintenant que nous avons défini la caractéristique d'un arbre de partitions, nous étendons cette définition à l'ensemble représenté par cet arbre de partitions.

Définition 5.59 (Caractéristique d'un ensemble et (k,q)-caractéristique)**.** Soit A un ensemble et soit X un sous-ensemble de A. Le 6-uplet $C = (T^c, \ell, S, a, d, b)$ est une *caractéristique* de A restreint à X s'il existe un arbre de partitions T de A tel que $C = \tau_X(T)$. De plus, si le codomaine de ℓ est $\{0,\dots,k\}$ et que $a + d(r^c) \leq q$, alors C est une (k,q)-*caractéristique* de A restreint à X.

Lemme 5.60. *Soient A un ensemble, X un sous-ensemble de A et T un (k,q)-arbre de partitions de A. Alors la caractéristique de T restreint à X est une (k,q)-caractéristique de A restreint à X.*

Démonstration. Soit $C = (T^c, \ell, S, a, d, b)$ la caractéristique de T restreint à X. D'après la Propriété 5.51, pour chaque tronçon P de T^c, nous avons que $\max(P) = \max(Q)$, où Q désigne le sous-chemin étiqueté de T reliant les extrémités de P. Plus particulièrement, pour tout sommet interne v et toute arête e de T^c, nous avons $0 \leq \ell(v) \leq k$ et $0 \leq \ell(e) \leq k$. De plus, par définition nous avons que $a + d(r^c) \leq q$. □

Lemme 5.61. *La taille d'une (k,q)-caractéristique d'un ensemble A restreint à un sous-ensemble X de taille l est majorée par $\mathcal{O}(k \cdot l)$ si $q = \infty$, et $\mathcal{O}(k \cdot q \cdot l)$ sinon.*

Démonstration. Soit $C = (T^c, \ell, S, a, d, b)$ une (k,q)-caractéristique de A restreint à X. L'arbre T^c à donc l feuilles et au plus autant de sommets pères de ces feuilles.

Dans le cas où $q = \infty$, les sommets contenus dans S sont les pères des feuilles de T^c ainsi que les sommets de ramification de $cp(T^c)$, ce qui fait au plus $2l - 1$ sommets. Ceci implique que le nombre de tronçons de T^c est majoré par $2l - 2$. De plus, ces tronçons ont été soumis à des contractions et, par la Propriété 5.52, ils contienent au plus $k + 2$ sommets internes. La taille de C est donc majorée par $\mathcal{O}(k \cdot l)$.

Nous passons maintenant au cas où $q < \infty$. Soit v le père d'une feuille de T^c et soit $P(r^c, v)$ le sous-chemin de T^c reliant r^c et v. Par définition, $P(r^c, v)$ a au plus $q + 1$ sommets en commun avec S (le $+1$ est pour v qui n'est pas nécessairement un sommet de ramification dans le (k,q)-arbre de partitions caractérisé par C). Le nombre d'éléments de S est donc majoré par $q \cdot l + 1$. En réutilisant les arguments du cas précédent, ainsi que la Propriété 5.52, nous arrivons à la conclusion que la taille de C est majorée en fonction de $\mathcal{O}(k \cdot q \cdot l)$. □

Lemme 5.62. *Le nombre de (k,q)-caractéristiques d'un ensemble A restreint à un sous-ensemble X de taille l est majoré en fonction de k, q et l.*

Démonstration. Soit $C = (T^c, \ell, S, a, d, b)$ une (k,q)-caractéristique de A restreint à X. D'après le Lemme 5.61, le nombre de sommets de S est majoré en fonction de k, q et l quand $q < \infty$, et en fonction de k et l quand $q = \infty$. Grâce à Cayley [11], nous savons que le nombre d'arbres à n sommets est majoré par n^{n-2} (notons que les différents placements de r^c dans T^c ainsi que les différentes bijections entre les feuilles de T^c et les éléments de X sont pris en compte). De ce fait, si nous regroupons les (k,q)-caractéristiques de

A restreint à X en p ensembles, tels que toutes les (k,q)-caractéristiques d'un même ensemble utilisent le même arbre de partition de X, alors p est majoré en fonction de k, q et l quand $q < \infty$, et en fonction de k et l quand $q = \infty$.

Il nous reste alors à montrer que le nombre de (k,q)-caractéristiques contenues dans chacun de ces ensembles est également majoré. Pour cela il nous suffit de rappeler que le codomaine de ℓ est $\{0,\dots,k\}$. Et, dans le cas où $q < \infty$, le codomaine de d est $\{0,\dots,q\}$, celui de b est $\{0,1,2\}$ et nous avons $0 \le a \le q$. □

5.3.3. Une relation d'ordre sur les arbres de partitions

Pour chaque nœud α d'une jolie décomposition donnée, au lieu de calculer l'ensemble $FSPT_\alpha$ des arbres de partitions de A_α (voir Section 5.2.2), nous souhaitons calculer un ensemble de caractéristiques FSC_α. Ce dernier doit alors vérifier que chacune des caractéristiques qu'il contient caractérise au moins un arbre de $FSPT_\alpha$, mais aussi que tous les arbres de $FSPT_\alpha$ sont caractérisés par une caractéristique de FSC_α. Malheureusement, comme nous le montre l'Exemple 5.84 (présenté à la fin de la Section 5.4.3), la compression que nous utilisons ne nous permet pas de satisfaire la dernière condition. Cependant, ladite condition n'est pas nécessaire au bon fonctionnement de l'algorithme. En effet, nous allons voir qu'il est possible de définir un ordre sur les arbres de partitions de $FSPT_\alpha$, nous permetant ainsi de ne considérer que les « plus petits » d'entre eux, et pour lesquels il est toujours possible de trouver une caractéristique correspondante dans FSC_α.

Soit A un ensemble et soit X un sous-ensemble de A. Soit T un arbre de partitions de A et soit T^r la restriction de T à X. Le *tronc* de T suivant X est obtenu en lissant les sommets de T^r qui ne sont pas dans S. Nous notons alors T_X^t le tronc de T suivant X et nous observons que les racines de T^r et T_X^t identifient le même sommet de T. Soit xy une arête de T_X^t et soit $P(x,y)$ le sous-chemin de T reliant x à y, nous disons alors que $P(x,y)$ est le *chemin sous-jacent* à xy dans T.

Définition 5.63 (Comparaison d'arbres de partitions)**.** Soient T et $\check{T}$ deux arbres de partitions de A, tous deux de largeur au plus k. Soient T^r et $\check{T}^r$ les restrictions respectives de T et $\check{T}$ à un sous-ensemble X de A. Nous écrivons alors $\check{T} \preccurlyeq_X T$ si ces deux arbres satisfont les conditions suivantes :

- $T_X^t \cong \check{T}_X^t$;
- $\mathrm{dr}_{\check{T}}(\check{r},\check{r}^r) = \mathrm{dr}_T(r,r^r)$;
- pour chaque sommet interne x de $T_X^t \cong \check{T}_X^t$, nous avons $\mathrm{hr}_{\check{T}}(x) = \mathrm{hr}_T(x)$;
- pour chaque sommet interne x de $T_X^t \cong \check{T}_X^t$, nous avons $\mathrm{nr}_{\check{T}}(x) = \mathrm{nr}_T(x)$;
- pour chaque arête xy de $T_X^t \cong \check{T}_X^t$, nous avons $\check{P}(x,y) \preccurlyeq P(x,y)$, où $\check{P}(x,y)$ et $P(x,y)$ sont les chemins sous-jacents à xy dans $\check{T}$ et T respectivement.

Nous définissons maintenant les ensembles de caractéristiques que nous devons calculer pour chacun des nœuds de la jolie décomposition donnée en entrée de l'algorithme de reconnaissance.

Définition 5.64 (Ensemble complet de (k,q)-caractéristiques)**.** Soit FSC_X un ensemble de (k,q)-caractéristiques d'un ensemble A restreint à un sous-ensemble X. Si, pour tout

(k,q)-arbre de partitions T de A, il existe un (k,q)-arbre de partitions $T \preccurlyeq_X \check{T}$, tel que $\tau_X(\check{T})$ appartient à FSC_X, et que, pour toute (k,q)-caractéristique C de FSC_X, il existe un (k,q)-arbre de partitions de A tel que $C = \tau_X(T)$, alors nous disons que FSC_X est un *ensemble complet de* (k,q)*-caractéristiques* pour A restreint à X.

5.4. Algorithme utilisant la compression

Il est maintenant temps d'utiliser la compression que nous venons de définir et de vérifier par la même occasion qu'elle nous fournit bien un algorithme de reconnaissance linéaire. Pour rappel, notre algorithme prend en entrée trois constantes k, q et l, un ensemble fini A ainsi qu'une fonction de partition ϕ de A, compatible avec une jolie décomposition $(\mathcal{T},\mathcal{X})$ de A, elle aussi fournie en entrée et dont la largeur est au plus l. Puis, en suivant une procédure montante sur les nœuds de $\mathcal{T}$, l'algorithme détermine si la largeur q-ramifiée suivant ϕ de A est majorée par k.

Nous reprenons donc la procédure montante résentée dans la Section 5.2.2 mais, au lieu de construire l'ensemble complet $FSPT_\alpha$ des (k,q)-arbres de partitions associé à un nœud α donné de $\mathcal{T}$, nous calculons l'ensemble complet FSC_α des (k,q)-caractéristiques de A_α restreint à X_α. Il faut alors vérifier que chaque (k,q)-caractéristique de FSC_α caractérise bien (au moins) un (k,q)-arbre de partitions de $FSPT_\alpha$, et inversement que chaque (k,q)-arbre de partitions de $FSPT_\alpha$ à une (k,q)-caractéristique associée dans FSC_α, et également que le calcul de FSC_α s'effectue en temps constant (c'est-à-dire majorée en fonction de k, q et $|X_\alpha|$), afin que la procédure montante sur $\mathcal{T}$ soit effectivement linéaire en fonction de la taille de $\mathcal{T}$ et donc de la taille de A.

5.4.1. Les nœuds de départ

Procédure de Départ. Soit α un nœud de départ de $\mathcal{T}$. Par définition, α est une feuille de $\mathcal{T}$ et nous avons $A_\alpha = X_\alpha$. Afin de determiner l'ensemble FSC_α, il nous suffit d'énumérer l'ensemble des différentes (k,q)-caractéristiques possibles pour A_α restreint à X_α. ■

Lemme 5.65. *Soit α un nœud de départ. La Procédure de Départ garantit que FSC_α est un ensemble complet de (k,q)-caractéristiques pour A_α restreint à X_α.*

Il nous faut vérifier que le calcul de FSC_α ne dépend pas de la taille de A.

Lemme 5.66. *La Procédure de Départ s'exécute en temps constant.*

Démonstration. D'après les Lemmes 5.61 et 5.62, le nombre de (k,q)-caractéristiques de FSC_α ainsi que leurs tailles sont majorés en fonction de k, q et $|X_\alpha|$. Nous pouvons donc énumérer FSC_α en temps constant. □

5.4.2. Les nœuds d'oubli

Procédure d'Oubli. Soit α un nœud d'oubli de $\mathcal{T}$, avec β son fils pour lequel nous avons préalablement calculé l'ensemble complet FSC_β de (k,q)-caractéristiques pour

A_β restreint à X_β. Pour chaque (k,q)-caractéristique $C_\beta \in FSC_\beta$ nous calculons une (k,q)-caractéristique C_α pour $A_\alpha = A_\beta$ restreint à $X_\alpha = X_\beta \setminus \{\bullet\}$, où $\bullet$ est l'élément oublié, et nous l'ajoutons à FSC_α. Nous rappelons que C_α est une notation élégante pour $(T^c_\alpha, \ell_\alpha, S_\alpha, a_\alpha, d_\alpha, b_\alpha)$, et il en va de même pour C_β.

Soit v la feuille de T^c_β qui est en bijection avec $\bullet$ et soit u le sommet de degré au moins trois qui est le plus proche de v dans T^c_β (si un tel sommet n'existe pas, c'est que T^c_β est un chemin et, dans ce cas, u désigne le voisin de l'unique autre feuille de ce chemin). Nous appelons w le voisin de u dans le sous-chemin $P^c_\beta(u,v)$ de T^c_β reliant u et v.

▹ Définition de T^c_α. L'arbre T^c_α est obtenu en supprimant de T^c_β les sommets du chemin $P^c_\beta(w,v)$, puis en effectuant certaines contractions de chemins étiquetés que nous détaillons dans la phase de *contraction des tronçons de* T^c_α. Si r^c_β fait partie des sommets supprimés, alors u prend le rôle de la racine de T^c_α sinon r^c_β garde son rôle de racine dans T^c_α, c'est-à-dire que, dans le premier cas, $r^c_\alpha := u$ et, dans le second cas, $r^c_\alpha := r^c_\beta$.

▹ Définition de S_α. L'ensemble S_α est obtenu en supprimant de S_β les sommets que nous avons supprimés de T^c_β, soit $S_\alpha := S_\beta \setminus V(P^c_\beta(w,v))$. Dans le cas où v est l'unique fils de u étant une feuille de T^r_β et que $b_\beta(u) = 1$, alors nous supprimons également u de S_β, et nous obtenons $S_\alpha := S_\beta \setminus V(P^c_\beta(u,v))$.

▹ Contraction des tronçons de T^c_α. La seul contraction de tronçon que nous opérons survient dans le cas où nous supprimons u de S_β. Soit P le tronçon de T^c_α contenant u, nous remplaçons alors P par sa contraction $\tau(P)$ dans T^c_α.

▹ Définitions de a_α. Si r^c_β fait partie des sommets supprimés alors, étant donné x, un sommet du chemin reliant r^c_β à w, tel que $b_\beta(x) = 2$, alors x est un sommet de ramification ancêtre de r^c_α, que nous avons supprimé, il nous faut donc nous rappeler son existance. Pour cela, nous définissons a_α en ajoutant à a_β le nombre de sommets de ramification ancêtres de u que nous venons de supprimer, nous avons donc $a_\alpha := a_\beta + \mathrm{dr}_T(r^c_\beta, u)$. Par contre, si r^c_β ne fait pas partie des sommets supprimés, alors nous n'avons supprimé aucun sommet de ramification ancêtre de u et, par conséquent, $a_\alpha := a_\beta$.

▹ Définitions de d_α et b_α. Étant donné un sommet interne x de T^c_α la suppresions des sommets de $P^c_\beta(w,v)$ dans T^c_β n'apportent aucun changement aux descendants de x, c'est pourquoi nous avons $d_\alpha(x) := d_\beta(x)$. Cette suppression ne change pas non plus le nombre de ramifications de x, nous avons donc $b_\alpha(x) := b_\beta(x)$. ■

Avant de montrer que la Procédure d'Oubli calcule un ensemble complet de (k,q)-caractéristiques pour A_α en temps constant, nous introduisons quelques notions supplémentaires portées sur l'utilisation des chemins étiquetés.

Définition 5.67 (Concaténation de chemins étiquetés)**.** Soient deux chemins étiquetés $P = (u_0, \ldots, u_r)$ et $Q = (v_0, \ldots, v_s)$, tels que $u_r = v_0$ est leur unique sommet en commun. La *concaténation* de P et Q, est le chemin étiqueté $P \odot Q = (u_0, \ldots, u_r = v_0, \ldots, v_s)$. Le sommet commun à P et Q porte alors le nom de *sommet de concaténation*.

Cette nouvelle opération va nous être utile dans le cas où le sommet u est un élément de S_β mais pas de S_α. En effet, dans cette configuration il existe trois tronçons de T^c_β admettant u comme une de leurs extrémités ; nous les appelons $P^c_\beta(s,u)$, $P^c_\beta(u,w)$ et

$P^c_\beta(u,v)$. Il se trouve alors que le tronçon $P^c_\beta(u,v)$ n'existe plus dans T^c_α, et que $P^c_\beta(s,u)$ et $P^c_\beta(u,w)$ ne peuvent plus être des tronçons de T^c_α car u ne fait plus partie de S_α. Cependant, les sommets s et w sont toujours dans S_α ce qui fait que nous pouvons définir le tronçon $P^c_\alpha(s,w) = P^c_\beta(s,u) \odot P^c_\beta(u,w)$, qu'il nous faut alors contracter.

Afin de vérifier que tout se passe bien, nous introduisons les propriétés suivantes.

Propriété 5.68. *Soient P et Q deux chemins étiquetés, nous avons $\tau(P \odot Q) = \tau(\tau(P) \odot \tau(Q))$.*

Démonstration. Étant donné que l'ordre des contractions que nous appliquons n'a pas d'importance, nous pouvons calculer $\tau(P \odot Q)$ par étapes successives. Nous commençons donc par contracter les arêtes de P, suivies de celles de Q, et nous finissons par la contraction du chemin qui, à ce moment précis, est définit par $\tau(P) \odot \tau(Q)$. □

Propriété 5.69. *Si $P_1 \preccurlyeq P_2$ et $Q_1 \preccurlyeq Q_2$, alors $P_1 \odot Q_1 \preccurlyeq P_2 \odot Q_2$.*

Démonstration. Soient $\bar{P}_1$, $\bar{P}_2$, $\bar{Q}_1$ et $\bar{Q}_2$ les extensions respectives de P_1, P_2, Q_1 et Q_2 qui vérifient $\bar{P}_1 \leq \bar{P}_2$ et $\bar{Q}_1 \leq \bar{Q}_2$. Il est alors évident que $\bar{P}_1 \odot \bar{Q}_1 \leq \bar{P}_2 \odot \bar{Q}_2$. Le fait que $\bar{P}_1 \odot \bar{Q}_1$ et $\bar{P}_2 \odot \bar{Q}_2$ soient respectivement des extensions de $P_1 \odot Q_1$ et $P_2 \odot Q_2$ est également assez simple à voir. □

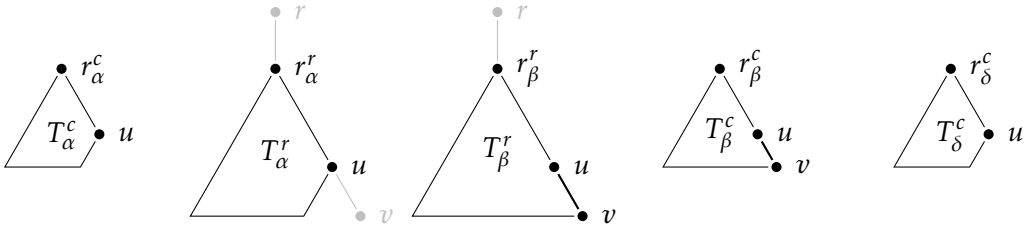
Il est maintenant temps de montrer que la Procédure d'Oubli calcule un ensemble complet de (k,q)-caractéristiques, et ce en temps constant. Pour cela nous utilisons trois lemmes. Dans le premier nous montrons que, pour chaque caractéristique que la Procédure d'Oubli calcule, il existe un (k,q)-arbre de partitions de A_α avec cette caractéristique. Le deuxième sert à prouver que l'ensemble calculé par la Procédure d'Oubli est complet, c'est-à-dire que, pour tout (k,q)-arbre de partitions de A_α, il existe une caractérisation de cet arbre (où du moins d'un arbre « plus petit »). Finalement, le dernier lemme prouve que le temps d'exécution de la Procédure d'Oubli dépend uniquement de k, q et $|X_\alpha|$.

Lemme 5.70. *Soit α un nœud d'oubli. La Procédure d'Oubli garantit que, pour chaque (k,q)-caractéristique C_δ de FSC_α, il existe un (k,q)-arbre de partitions T de A_α tel que $C_\delta = \tau_\alpha(T)$.*

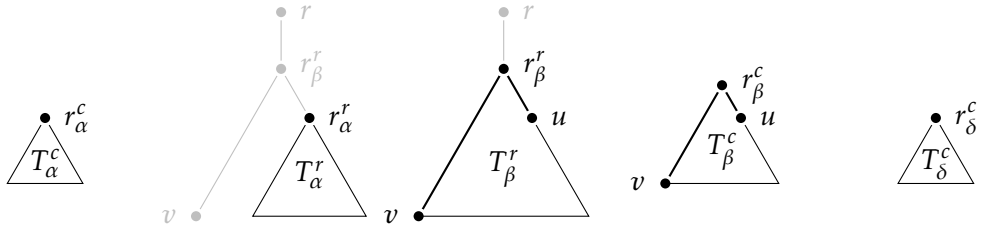
Démonstration. Soit β le fils de α dans $\mathcal{T}$ et soit C_β la (k,q)-caractéristique de A_β restreint à X_β à partir de laquelle nous avons calculé C_δ. Par hypothèse, FSC_β est un ensemble complet de (k,q)-caractéristiques pour A_β restreint à X_β, il existe donc un (k,q)-arbre de partitions T de A_β tel que $C_\beta = \tau_\beta(T)$. Comme $A_\alpha = A_\beta$, nous avons que T est également un (k,q)-arbre de partitions de A_α. Ce que nous cherchons à démontrer c'est que $C_\delta = \tau_\alpha(T)$. Pour cela, nous considérons $C_\alpha = \tau_\alpha(T)$ et nous montrons que $C_\delta = C_\alpha$.

Afin de faciliter l'écriture de la preuve, nous faisons appel aux notations suivantes. Soit T^r_α et T^r_β les sous-arbres de T respectivement restreint à X_α et X_β. Nous rappelons que C_α est une notation élégante pour $(T^c_\alpha, \ell_\alpha, S_\alpha, a_\alpha, d_\alpha, b_\alpha)$, et qu'il en va de même pour C_β et C_δ.

Lemme 5.70.1. *Si T^c_β est un chemin alors $C_\alpha = C_\delta$.*



(a) Dans le cas où r_β^r n'est pas un sommet de $P_\beta^r(u,v)$.



(b) Dans le cas où r_β^r est un sommet de $P_\beta^r(u,v)$.

FIGURE 5.2. : Les arbres utilisés par la preuve de la Procédure d'Oubli. Les parties grises schématisent certaines branches de l'arbre T qui ont été « oubliées » par les restrictions suivant X_α et X_β.

D'après la Procédure de Caractérisation, nous avons que T_β^c est un chemin si et seulement si T_β^r est un chemin. Et si T_β^r est un chemin, c'est que X_β ne contient que deux éléments. Par conséquent, X_α ne contient plus qu'un seul de ces deux éléments, et donc T_α^r est une arête. La Procédure de Caractérisation nous indique donc que C_α est une caractéristique pour A_α restreint à X_α, et que T_α^c est une arête. En parallèle, si T_β^c est un chemin, alors la Procédure d'Oubli implique que T_δ^c est une arête. Par conséquent, quand T_α^r est une arête, nous nous rendons assez vite compte que C_α et C_δ sont identiques.

Nous pouvons donc assumer que T_β^r n'est pas un chemin. Nous appelons alors u le sommet de degré au moins 3 dans T_β^r qui est le plus proche du sommet v, lequel est en bijection avec l'élément oublié (noté $\bullet$), et nous appelons $P_\beta^r(u,v)$ le sous-chemin de T_β^r reliant u et v. Comme les arbres T_α^r, T_β^r, T_α^c, T_β^c et T_δ^c engendrent un nombre non négligeable de notations, et que leurs formes varient en fonction de la position de u dans chacun d'eux, nous nous appuyons sur la FIGURE 5.2 afin de garder une bonne visibilité.

Montrer que $C_\alpha = C_\delta$, revient à montrer que chaque composante de C_α est identique à la composante correspondante dans C_δ. Nous étudions donc ces composantes petit à petit, ce qui rend la lecture de la preuve un peu plus agréable.

Lemme 5.70.2. *Les ensembles S_α et S_δ sont égaux.*

Par la Procédure de Caractérisation, S_α contient les sommets de T_α^r qui sont soit les pères des feuilles de T_α^r, soit des sommets de ramification dans $cp(T)$. De même, S_β

contient les sommets qui sont soit les pères des feuilles de T^r_β, soit des sommets de ramification dans $cp(T)$. De plus, par construction, nous avons que T^r_α est le sous-arbre de T^r_β emputé du chemin $P^r_\beta(w,v)$, où w est le voisin de u dans $P^r_\beta(u,v)$. Ces observations nous mènent à la conclusion que les seuls sommets de S_β qui ne peuvent pas faire partie de S_α sont les sommets de $P^r_\beta(w,v)$ ainsi que le sommet u dans le cas bien particulier que nous explicitons de ce pas.

Il faut tout d'abord que u soit un élément de S_β, c'est-à-dire que u est soit le père d'une feuille de T^r_β, soit un sommet de ramification de T. Si u ne fait plus partie de S_α c'est qu'alors il n'est ni le père d'une feuille dans T^r_α, ni un sommet de ramification de T. Par conséquent, pour que u soit un élément de S_β mais pas de S_α, il faut qu'il ne soit pas un sommet de ramification de T et qu'il soit le père d'une feuille dans T^r_β mais plus dans T^r_α. La seule configuration mettant en scène ce comportement est visible quand v est l'unique fils de u ayant la propriété d'être une feuille de T^r_β et que u n'est pas un sommet de ramification de $cp(T)$, ce qui se traduit par $b_\beta(u) = 1$.

En observant la Procédure d'Oubli ainsi que la discussion précédente, nous nous rendons compte que le cas où $S_\alpha = S_\beta \setminus V(P^r_\beta(w,v))$ correspond au cas où $S_\delta = S_\beta \setminus V(P^c_\beta(\dot{w},v))$, avec $\dot{w}$ le voisin de u dans $P^c_\beta(u,v) = \tau(P^r_\beta(u,v))$. De plus, par définition, si un sommet x de $P^r_\beta(w,v)$ est dans S_β, alors il a un correspondant dans $P^c_\beta(\dot{w},v)$, et nous avons $S_\delta \subseteq S_\alpha$. Inversement, si un sommet x appartient à $P^c_\beta(\dot{w},v)$ alors il a un correspondant dans $P^r_\beta(w,v)$, et donc $S_\alpha \subseteq S_\delta$. Nous pouvons réappliquer ce raisonnement dans le cas où $S_\alpha = S_\beta \setminus V(P^r_\beta(u,v))$, qui correspond au cas où $S_\delta = S_\beta \setminus V(P^c_\beta(u,v))$. Tout ceci nous montre bien que $S_\alpha = S_\delta$.

Lemme 5.70.3. *Les racines de T^c_α et T^c_δ identifient le même sommet.*

Nous rappelons que, par construction, T^r_α est le sous-arbre de T^r_β emputé du chemin $P^r_\beta(w,v)$, avec w le voisin de u dans $P^r_\beta(u,v)$. De plus, la Procédure de Caractérisation nous indique que r^c_α identifie le même sommet que r^r_α, et que r^c_β identifie le même sommet que r^r_β. Notre étude porte alors sur deux cas : soit r^r_β est un sommet de $P^r_\beta(w,v)$, soit il n'en est pas un.

Dans le cas où r^r_β n'est pas un sommet de $P^r_\beta(w,v)$ alors, par restriction, nous avons que r^r_α et r^r_β identifient un même sommet dans T. De plus, nous avons que r^c_β n'est pas un sommet de $P^c_\beta(\dot{w},v)$, avec $\dot{w}$ le voisin de u dans $P^c_\beta(u,v) = \tau(P^r_\beta(u,v))$. Et, par la Procédure d'Oubli, nous avons que r^c_δ identifie le même sommet que r^c_β. Finalement tous les sommets r^c_α, r^r_α, r^r_β, r^c_β et r^c_δ identifient le même sommet.

Dans le cas où r^r_β est un sommet de $P^r_\beta(w,v)$ alors, par restriction, nous avons que r^r_α et u identifient un même sommet dans T. De plus, nous avons que r^c_β est un sommet de $P^c_\beta(\dot{w},v)$. Et, par la Procédure d'Oubli, nous avons que r^c_δ identifie le même sommet que u. Finalement tous les sommets r^c_α, r^r_α, u et r^c_δ identifient le même sommet.

Lemme 5.70.4. *Les arbres T^c_α et T^c_δ sont isomorphes et les fonctions ℓ_α et ℓ_δ sont égales.*

Soient x et y deux éléments de S_α et soit $P^c_\alpha(x,y)$ le tronçon de T^c_α reliant x à y. Comme S_α et S_δ sont égaux, nous pouvons définir $P^c_\delta(x,y)$, le tronçon reliant x à y dans T^c_δ. De

plus, S_α est un sous-ensemble de S_β, ce qui nous permet de définir $P^c_\beta(x,y)$, qui est le chemin étiqueté reliant x à y dans T^c_β. Finalement nous considérons $P(x,y)$ qui est le chemin étiqueté reliant x à y dans T.

Pour montrer que $T^c_\alpha \cong T^c_\delta$, il nous suffit de montrer que $P^c_\alpha(x,y)$ et $P^c_\delta(x,y)$ sont formés par la même séquence de sommets. Et, pour montrer que ℓ_α et ℓ_δ sont égales, il nous suffit de montrer que les étiquettes attribuées à chaque sommet z ainsi qu'à chaque arête e de $P^c_\alpha(x,y)$ sont respectivement égales aux étiquettes attribuées aux correspondants de z et e dans $P^c_\delta(x,y)$. Pour cela, nous devons différencier le cas où u est un sommet interne de $P^c_\beta(x,y)$ du cas où il ne l'est pas. Notons que si u est un élément de S_α, alors il ne peut pas être un sommet interne de $P^c_\alpha(x,y)$; il n'est alors pas non plus un sommet interne de $P^c_\beta(x,y)$.

Dans le cas où u n'est pas un sommet interne de $P^c_\beta(x,y)$. La Procédure de Caractérisation nous indique que $P^c_\alpha(x,y) = \tau(P(x,y))$ et que $P^c_\beta(x,y) = \tau(P(x,y))$. Comme la Procédure d'Oubli ne touche pas aux chemins étiquetés qui ne contiennent pas u, nous avons que $P^c_\delta(x,y) = P^c_\beta(x,y)$. Ce qui nous donne immédiatement $P^c_\alpha(x,y) = P^c_\delta(x,y)$.

Dans le cas où u est un sommet interne de $P^c_\beta(x,y)$, c'est qu'il ne fait pas partie des éléments de S_α, il fait cependant partie des éléments de S_β. En effet comme u est un sommet de degré au moins 3 dans T^r_β, il a au moins deux fils. Il est donc soit le père d'une feuille de T^r_β, soit un sommet de ramification de $cp(T)$. Cette fois-ci, la Procédure de Caractérisation nous indique que $P^c_\alpha(x,y) = \tau(P(x,y))$ et que $P^c_\beta(x,y) = \tau(P(x,u)) \odot \tau(P(u,y))$. La Procédure d'Oubli, quant à elle, nous dit que $P^c_\delta(x,y) = \tau(P^c_\beta(x,y))$. Nous avons alors $P^c_\delta(x,y) = \tau(\tau(P(x,u)) \odot \tau(P(u,y)))$ et, en appliquant la Propriété 5.68, nous obtenons $P^c_\delta(x,y) = \tau(P(x,y)) = P^c_\alpha(x,y)$.

Lemme 5.70.5. *Les entiers a_α et a_δ sont égaux.*

Par la Procédure de Caractérisation, nous avons que $a_\alpha = \mathrm{dr}_T(r, r^r_\alpha)$ et que $a_\beta = \mathrm{dr}_T(r, r^r_\beta)$. Afin de montrer que a_α et a_δ comptent le même nombre de « sommets ancestraux oubliés », nous réutilisons la distinction concernant le possible changement de racine dû à la suppression de r^r_β.

Nous reprenons alors le cas où r^r_β n'est pas un sommet de $P^r_\beta(w,v)$, et nous avons que les sommets r^r_α et r^r_β identifient le même sommet. Il est alors possible d'écrire $\mathrm{dr}_T(r, r^r_\alpha) = \mathrm{dr}_T(r, r^r_\beta)$ et donc $a_\alpha = a_\beta$. D'après la Procédure d'Oubli, et comme r^c_β n'est pas un sommet de $P^c_\beta(\dot{w},v)$, nous avons $a_\delta = a_\beta$ et donc $a_\alpha = a_\delta$.

Dans le cas où r^r_β est un sommet de $P^r_\beta(w,v)$, alors les sommets r^r_α et u identifient le même sommet et r^r_β est un sommet du chemin $P(r,u)$. Nous avons donc la suite d'égalités $\mathrm{dr}_T(r, r^r_\alpha) = \mathrm{dr}_T(r,u) = \mathrm{dr}_T(r, r^r_\beta) + \mathrm{dr}_T(r^r_\beta, u) = a_\beta + \mathrm{dr}_T(r^c_\beta, u)$. Enfin, par la Procédure d'Oubli nous avons $a_\delta = a_\beta + \mathrm{dr}_T(r^c_\beta, u)$ et donc $a_\alpha = a_\delta$.

Lemme 5.70.6. *Les fonctions d_α et b_α sont égales aux fonctions d_δ et b_δ.*

Soit x un sommet interne de T^c_α. Par la Procédure de Caractérisation, x est également un sommet de T^r_α. Grâce au Lemme 5.70.4, nous savons que x est un élément de T^c_δ.

Puis par la Procédure d'Oubli, nous avons que x est un sommet de T^c_β. Finalement, en réutilisant la Procédure de Caractérisation, x est aussi un sommet de T^r_β.

D'après la Procédure de Caractérisation, les valeurs d_α et d_β représentent la hauteur de ramification de x dans T, notée $\mathrm{hr}_T(x)$, ce qui implique leur égalité. De plus, cette même Procédure de Caractérisation ne change pas le nombre de ramifications de x dans T. Nous avons donc $b_\alpha = b_\beta$. Enfin, en nous référant à la Procédure d'Oubli, nous avons $d_\delta = d_\beta$ et $b_\delta = b_\beta$. Les égalités recherchées suivent sans encombre.

Nous venons de montrer que chaque composante de C_α est identique à la composante correspondante dans C_δ, prouvant ainsi que $C_\alpha = C_\delta$. □

Chaque (k,q)-caractéristique de FSC_α représente donc un (k,q)-arbre de partitions de A_α. Nous montrons alors que, réciproquement, chaque (k,q)-arbre de partitions de A_α est représenté par une (k,q)-caractéristique de FSC_α.

Lemme 5.71. *Soit α un nœud d'oubli. La Procédure d'Oubli garantit que, pour chaque (k,q)-arbre de partitions T de A_α, il existe un (k,q)-arbre de partitions $\check{T} \preccurlyeq_\alpha T$ tel que $\tau_\alpha(\check{T}) \in FSC_\alpha$.*

Démonstration. Soit β le fils de α dans $\mathcal{T}$. Nous rappelons que $A_\alpha = A_\beta$, par conséquent chaque (k,q)-arbre de partitions T de A_α est également un (k,q)-arbre de partitions de A_β. Par hypothèse, nous savons que FSC_β est un ensemble complet de (k,q)-caractéristiques pour A_β restreint à X_β, il existe donc un (k,q)-arbre de partitions $\check{T} \preccurlyeq_\beta T$ tel que $\tau_\beta(\check{T}) \in FSC_\beta$. De plus le Lemme 5.70 nous indique que $\tau_\alpha(\check{T})$ est un élément de FSC_α. Il nous reste donc à montrer que $\check{T} \preccurlyeq_\alpha T$.

Par définition, $\check{T}^t_\beta$ et T^t_β sont isomorphes, $\mathrm{dr}_{\check{T}}(\check{r}, \check{r}^r_\beta) = \mathrm{dr}_T(r, r^r_\beta)$ et, pour tout sommet interne x et toute arête e de $T^t_\beta \cong \check{T}^t_\beta$, nous avons $\mathrm{hr}_{\check{T}}(x) = \mathrm{hr}_T(x)$, $\mathrm{nr}_{\check{T}}(x) = \mathrm{nr}_T(x)$ et $\check{P}(e) \preccurlyeq P(e)$, où $\check{P}(e)$ et $P(e)$ sont les chemins sous-jacents à e dans $\check{T}$ et T respectivement. Par restriction, $\check{T}^t_\alpha$ et T^t_α sont isomorphes, $\mathrm{dr}_{\check{T}}(\check{r}, \check{r}^r_\alpha) = \mathrm{dr}_T(r, r^r_\alpha)$ et, pour tout sommet interne x de $\check{T}^t_\alpha \cong T^t_\alpha$, nous avons $\mathrm{hr}_{\check{T}}(x) = \mathrm{hr}_T(x)$ ainsi que $\mathrm{nr}_{\check{T}}(x) = \mathrm{nr}_T(x)$. Toujours par restriction et en nous appuyant sur la Propriété 5.69 pour la concaténation potentielle, nous avons que $\check{P}(e) \preccurlyeq P(e)$, pour toute arête e de $\check{T}^t_\alpha \cong T^t_\alpha$, avec $\check{P}(e)$ et $P(e)$ les chemins sous-jacents à e dans $\check{T}$ et T respectivement. □

Corollaire 5.72. *Soit α un nœud d'oubli. La Procédure d'Oubli garantit que FSC_α est un ensemble complet de (k,q)-caractéristique pour A_α restreint à X_α.*

Il nous reste alors à montrer que le temps d'exécution de la Procédure d'Oubli est majoré en fonction de k, q et $|X_\alpha|$.

Lemme 5.73. *La Procédure d'Oubli s'execute en temps constant.*

Démonstration. Soit α un nœud d'oubli et soit β son fils dans $\mathcal{T}$. Par hypothèse, nous savons que FSC_β est un ensemble complet de (k,q)-caractéristiques pour A_β restreint à X_β. Et, par le Lemme 5.61, la taille d'une (k,q)-caractéristique C_β de FSC_β est majorée

en fonction de k, q et $|X_\beta|$. Les définitions des différentes composantes de C_α se font en parcourant T^c_β, et donc en temps constant. Le temps d'exécution de la phase de *contraction des tronçons de* T^c_α est majoré en fonction de la taille du chemin auquel la contraction est appliquée. Ce chemin est un sous-graphe de T^c_β, la phase de *contraction des tronçons de* T^c_α s'exécute donc en temps constant. Finalement, l'énumération des (k,q)-caractéristiques de FSC_β est majorée en fonction de k, q et $|X_\beta|$, comme nous l'indique le Lemme 5.62. □

5.4.3. Les nœuds d'introduction

Procédure d'Introduction. Soient α un nœud d'introduction et β son fils dans $\mathcal{T}$. Nous appelons $\bullet$ l'élément introduit, c'est-à-dire que $X_\alpha \setminus X_\beta = \{\bullet\}$. Afin de calculer l'ensemble complet des (k,q)-caractéristiques de α, nous considérons que nous avons préalablement calculé l'ensemble FSC_β. Puis, pour chaque (k,q)-caractéristique C_β de FSC_β et pour chaque insertion possible de $\bullet$ dans C_β, nous calculons la caractéristique C_α associée et, si C_α est une (k,q)-caractéristique, alors nous l'ajoutons à FSC_α. Ici encore C_α est une notation élégante pour $(T^c_\alpha, \ell_\alpha, S_\alpha, a_\alpha, d_\alpha, b_\alpha)$ et il en va de même pour C_β. Il existe trois façons d'insérer $\bullet$ dans T^c_β que nous décrivons séparement.

Introduction de Type 1. ▹ Définition de T^c_α. Pour insérer $\bullet$ nous choisissons u, un sommet interne de T^c_β, et nous y attachons une nouvelle feuille v en bijection avec $\bullet$, créant ainsi une nouvelle arête uv. Notons que la racine de T^c_β garde son rôle dans T^c_α.

▹ Définition de ℓ_α. Les deux éléments ajoutés à T^c_β sont la feuille v, qui n'est pas concernée, et l'arête uv, pour laquelle nous définissons $\ell_\alpha(uv) := \phi_{A_\alpha}(\{X_\beta, \{\bullet\}\})$. Puis, pour chaque sommet interne x de T^c_α, nous définissons $\ell_\alpha(x) := I_\phi(\ell_\beta(x), \mathcal{Q}_x, \bullet)$, où $\mathcal{Q}_x$ est la partition de X_α induite par x. De la même façon, pour toute arête e de T^c_α différente de uv, nous définissons $\ell_\alpha(e) := I_\phi(\ell_\beta(e), \mathcal{Q}_e, \bullet)$, où $\mathcal{Q}_e$ est la bipartition de X_α induite par e.

▹ Définition de S_α, a_α, d_α et b_α. Cette insertion ne change pas le nombre de ramifications ancestrales, et nous avons $a_\alpha := a_\beta$. De plus, pour tout sommet interne x de T^c_α, ni la hauteur de ramification de x, ni son nombre de ramifications ne sont modifiés, nous définissons alors $d_\alpha(x) := d_\beta(x)$ et $b_\alpha(x) := b_\beta(x)$. Finalement, le sommet u est le père de la feuille v dans T^c_α, il doit alors faire partie de S_α, nous définissons donc $S_\alpha := S_\beta \cup \{u\}$.

Introduction de Type 2. Cette insertion ne peut se produire que si $a_\beta = 0$ et que T^c_β n'est pas réduit à une arête.

▹ Définition de T^c_α. Pour insérer $\bullet$ dans T^c_β, nous créons un nouveau sommet u, dont les fils sont la feuille v (en bijection avec $\bullet$) et la racine de T^c_β que nous appelons w.

▹ Définition de ℓ_α. Le sommet u ainsi que les arêtes uv et uw induisent la même partition de X_α et nous avons $\ell_\alpha(u) = \ell_\alpha(uv) = \ell_\alpha(uw) := \phi_{A_\alpha}(\{X_\beta, \{\bullet\}\})$. Pour chaque sommet interne x de T^c_α, différent de u, nous définissons $\ell_\alpha(x) := I_\phi(\ell_\beta(x), \mathcal{Q}_x, \bullet)$, où $\mathcal{Q}_x$ est la partition de X_α induite par x. Il en va de même pour chaque arête e de T^c_α, différente de uv et uw, pour laquelle nous définissons $\ell_\alpha(e) := I_\phi(\ell_\beta(e), \mathcal{Q}_e, \bullet)$, où $\mathcal{Q}_e$ est la bipartition de X_α induite par e.

▹ Définition de S_α, a_α, d_α et b_α. Le fait que w identifie la racine de T_β implique que u identifie la racine de T_α. De plus, le sommet u à clairement une seule ramification et sa

hauteur de ramification est égale à celle de w. Nous avons donc $a_\alpha := 0$, $d_\alpha(u) := d_\beta(w)$ et $b_\alpha(u) := 1$. Pour tout sommet interne x de T^c_α, différent de u, ni la hauteur de ramification ni le nombre de ramifications de x ne sont modifiés, nous définissons alors $d_\alpha(x) := d_\beta(x)$ et $b_\alpha(x) := b_\beta(x)$. Finalement, en temps que père de v dans T^c_α, le sommet u doit faire partie de S_α, par conséquent $S_\alpha := S_\beta \cup \{u\}$.

Introduction de Type 3. Cette insertion ne peut être effectué que si T^c_β n'est pas réduit à une arête.

▹ Définition de T^c_α. Soit sw une arête de T^c_β, avec s le père de w. Nous effectuons alors une subdivision de sw et nous obtenons les arêtes su et uw. Puis nous ajoutons le sommet v (ici encore en bijection avec $\bullet$) en temps que fils de u. Enfin, pour que la définition de T^c_α soit complète, nous attendons la fin de la phase de *contraction des tronçons de* T^c_α. Notons que la racine de T^c_β garde son rôle dans T^c_α.

▹ Définition de ℓ_α. Avant que v ne soit inséré, la subdivision de sw nous indique que su, uw et u reçoivent l'étiquette attribuée à sw avant sa subdivision. À ce moment précis, su, uw et u induisent la même partition de X_β. Cependant, après insertion de $\bullet$, les partitions de X_α induites par su, uw et u ne sont plus égales. Ceci nous permet de définir :

- $\ell_\alpha(u) := I_\phi(\ell_\beta(sw), \mathcal{Q}_u, \bullet)$, avec $\mathcal{Q}_u$ la partition de X_α induite par u ;
- $\ell_\alpha(su) := I_\phi(\ell_\beta(sw), \mathcal{Q}_{su}, \bullet)$, avec $\mathcal{Q}_{su}$ la bipartition de X_α induite par su ;
- $\ell_\alpha(uw) := I_\phi(\ell_\beta(sw), \mathcal{Q}_{uw}, \bullet)$, avec $\mathcal{Q}_{uw}$ la bipartition de X_α induite par uw.

Nous définissons ensuite l'étiquette $\ell_\alpha(uv) := \phi_{A_\alpha}(\{X_\beta, \{\bullet\}\})$. Enfin, pour chaque sommet interne x différent de u et chaque arête e différente de su, uw et uv dans T^c_α, nous définissons $\ell_\alpha(x) := I_\phi(\ell_\beta(x), \mathcal{Q}_x, \bullet)$ et $\ell_\alpha(e) := I_\phi(\ell_\beta(e), \mathcal{Q}_e, \bullet)$, avec $\mathcal{Q}_x$ la partition de X_α induite par x et $\mathcal{Q}_e$ la bipartition de X_α induite par e.

▹ Définition de S_α. L'introduction du sommet v en temps que feuille de T^c_α fait que son père u doit faire partie de S_α, nous avons donc $S_\alpha := S_\beta \cup \{u\}$. Si w est l'unique descendant de s dans T^c_β et que $b_\beta(s) = 0$, alors s n'a pas besoin [3] de faire partie de S_α et nous avons $S_\alpha := S_\beta \cup \{u\} \setminus \{s\}$.

▹ Définition de a_α. Comme l'insertion de $\bullet$ ne change pas la racine, le nombre de ramifications ancestrales oubliées ne change pas non plus, et nous avons $a_\alpha := a_\beta$.

▹ Définition de b_α. Dans le cas où w n'est pas une feuille de T^c_β, nous définissons $b_\alpha(u) := 1$ (car w induit une ramification de u). De plus, le nombre de ramifications de s ne change pas et, si s est un élément de S_β et donc de S_α, nous avons $b_\alpha(s) := b_\beta(s)$.

Dans le cas où w est une feuille de T^c_β, le sommet u n'admet aucune ramification et nous définissons $b_\alpha(u) := 0$. La subdivision de sw fait apparaitre une nouvelle ramification au niveau de s, ce qui se traduit par $b_\alpha(s) := \min\{2, b_\beta(s) + 1\}$.

Dans tous les cas, pour chaque élément x de S_α différent de s et u, la subdivision de sw ne change pas le nombre de ramifications de x, c'est pourquoi nous avons $b_\alpha(x) := b_\beta(x)$.

▹ Définition de d_α. Premièrement, la hauteur de ramification de u est égale à celle de w, nous définissons donc $d_\alpha(u) := 0$ si w est une feuille et $d_\alpha(u) := d_\beta(z)$ sinon, avec

3. En effet, dans ce cas précis, s n'est ni le père d'une des feuilles de T^c_α, ni un sommet de ramification dans T_α.

z l'élément de S_α descendant de u le plus proche de w dans T_α^c. Dans le cas où w est une feuille de T_β^c et que $b_\beta(s) = 1$, la phase précédente nous indique que le sommet s devient un sommet de ramification, augmentant ainsi sa hauteur de ramification et potentiellement celles de ses ancêtres. Nous définissons alors $d_\alpha(s) := d_\beta(s) + 1$ et, pour chaque sommet x ancêtre de s dans T_α^c, nous avons $d_\alpha(x) := \max\{d_\beta(x), d_\alpha(y) + \varepsilon(x)\}$, où y est le fils de x dans la ramification contenant s et où $\varepsilon(x)$ vaut 1 si $b_\alpha(x) = 2$ et 0 sinon. Finalement, pour chaque sommet x interne de T_α^c qui n'a pas encore été traité (ce qui inclut s et ses ancêtres si w n'est pas une feuille de T_β^c ou que $b_\beta(s) \neq 1$), nous définissons $d_\alpha(x) := d_\beta(x)$.

▹ Contraction des tronçons de T_α^c. L'apparition de u dans S_α entraine une séparation (voir Définition 5.76 ci-après) du tronçons de T_β^c contenant l'arête sw. De plus, la possible disparition de s dans S_α entraine une concaténation des tronçons de T_β^c ayant s pour extrémité. Nous procédons alors à la contraction des tronçons ainsi induits dans T_α^c. ■

Nous avons encore une fois besoin d'introduire quelques manipulations de chemins étiquetés supplémentaires.

Propriété 5.74. *Soit $\lambda : \mathbb{N} \to \mathbb{N}$ une fonction strictement croissante, alors $\lambda(\tau(P)) = \tau(\lambda(P))$.*

Démonstration. Montrer ce résultat, revient à montrer qu'il existe une contraction dans P si et seulement si il en existe une dans $\lambda(P)$. Pour cela nous considérons que les sommets de P sont indexés de v_0 à v_n et que ses arêtes sont indexées de e_1 à e_n.

Sans perte de généralité nous considérons qu'il existe une contraction entre e_i et v_j dans P, avec $i < j$. C'est-à-dire que, pour tout $i < k \leq j$, nous avons $\ell(e_i) \leq \ell(e_k) \leq \ell(v_j)$ et, pour tout $i \leq k < j$, nous avons $\ell(e_i) \leq \ell(v_k) \leq \ell(v_j)$. Étant donné que λ est strictement croissante, pour tout $i < k \leq j$, nous avons $\lambda(\ell(e_i)) \leq \lambda(\ell(e_k)) \leq \lambda(\ell(v_j))$ et, pour tout $i \leq k < j$, nous avons $\lambda(\ell(e_i)) \leq \lambda(\ell(v_k)) \leq \lambda(\ell(v_j))$. Il existe donc une contraction entre e_i et v_j dans $\lambda(P)$.

Réciproquement, s'il existe une contraction entre e_i et v_j dans $\lambda(P)$ alors, pour tout $i < k \leq j$, nous avons $\lambda(\ell(e_i)) \leq \lambda(\ell(e_k)) \leq \lambda(\ell(v_j))$ et, pour tout $i \leq k < j$, nous avons $\lambda(\ell(e_i)) \leq \lambda(\ell(v_k)) \leq \lambda(\ell(v_j))$. Encore une fois nous utilisons le fait que λ est strictement croissante et, pour tout $i < k \leq j$, nous avons $\ell(e_i) \leq \ell(e_k) \leq \ell(v_j)$ et, pour tout $i \leq k < j$, nous avons $\ell(e_i) \leq \ell(v_k) \leq \ell(v_j)$. Il existe donc contraction entre e_i et v_j dans P. □

Propriété 5.75. *Soient P et Q deux cehmins étiquetés tels que $P \preccurlyeq Q$, et soit $\lambda : \mathbb{N} \to \mathbb{N}$ une fonction strictement croissante. Nous avons alors $\lambda(P) \preccurlyeq \lambda(Q)$.*

Démonstration. Soit $\bar{P}$ et $\bar{Q}$ des extensions respectives de P et Q telles que $\bar{P} \leq \bar{Q}$. Comme λ est strictement croissante, nous avons $\lambda(\bar{P}) \leq \lambda(\bar{Q})$. Il va de soi que $\lambda(\bar{P})$ est une extension de $\lambda(P)$ et que $\lambda(\bar{Q})$ est une extension de $\lambda(Q)$. □

Définition 5.76 (Séparation d'un chemin étiqueté)**.** Soit P un chemin étiqueté de longueur au moins 2. La *séparation de P suivant un sommet x* résulte en une paire de chemins (P_1, P_2) telle que $P = P_1 \odot P_2$ et x est le sommet de concaténation. La *séparation de P suivant une arête e* correspond à la séparation suivant x du chemin résultant d'une subdivision de e dans P, où x représente le sommet issu de cette subdivision.

Propriété 5.77. *Soit P un chemin étiqueté et soit $\dot{P} = \tau(P)$ sa contraction. Soit $\dot{e}$ une arête de $\dot{P}$ et soit e l'arête correspondante dans P. Si nous considérons les séparations (P_1, P_2) et $(\dot{P}_1, \dot{P}_2)$ de P et $\dot{P}$ suivant e et $\dot{e}$ respectivement, alors nous avons $\tau(P_1) = \tau(\dot{P}_1)$ et $\tau(P_2) = \tau(\dot{P}_2)$.*

Démonstration. Nous appelons $\dot{s}$ et $\dot{w}$ les incidences de $\dot{e}$ dans $\dot{P}$ (avec $\dot{s}$ le père de $\dot{w}$) et correspondant aux extrémités s et w du chemin sous-jacent à $\dot{e}$ dans P. Soient u et $\dot{u}$ les sommets issus des subdivisions respectives de e et $\dot{e}$. Soit P^+ le chemin résultant de la subdivision de e dans P et soit $\dot{P}^+$ le chemin résultant de la subdivision de $\dot{e}$ dans $\dot{P}$. Si nous appelons x et y les extrémités de $\dot{P}^+$ (avec x l'ancêtre de y), qui sont également les extrémités de P^+, alors il est évident que $\tau(P^+(x,s)) = \dot{P}^+(x,\dot{s})$ et que $\tau(P^+(w,y)) = \dot{P}^+(\dot{w},y)$. Le fait que nous ayons $\tau(P^+(s,u)) = \dot{P}^+(\dot{s},\dot{u})$ ainsi que $\tau(P^+(u,w)) = \dot{P}^+(\dot{u},\dot{w})$ découle de la définition d'une contraction. Finalement, en nous appuyant sur la Propriété 5.68, nous avons :

$$\tau(P^+(x,u)) = \tau(\tau(P^+(x,s) \odot \tau(P^+(s,u))) = \tau(\dot{P}^+(x,\dot{s}) \odot \dot{P}^+(\dot{s},\dot{u})) = \tau(\dot{P}^+(x,\dot{u}))$$

et, de même, $\tau(P^+(u,y)) = \tau(\dot{P}^+(\dot{u},y))$, avec $(P^+(x,u), P^+(u,y))$ la séparation de P suivant e et $(\dot{P}^+(x,\dot{u}), \dot{P}^+(\dot{u},y))$ la séparation de $\dot{P}$ suivant $\dot{e}$. □

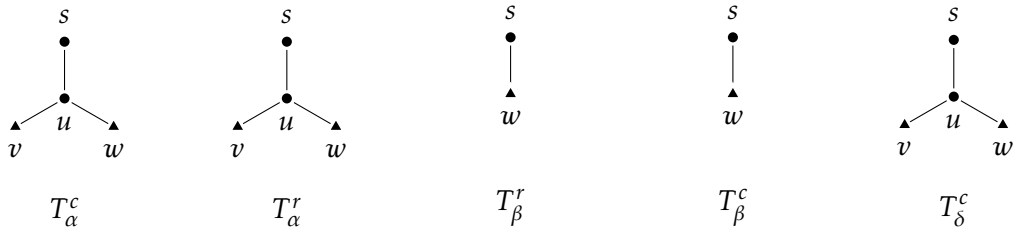
Propriété 5.78. *Soient P et Q deux chemins étiquetés tels que $P \preccurlyeq Q$. Alors, pour toute séparation (Q_1, Q_2) de Q suivant un sommet v, il existe une séparation (P_1, P_2) de P, suivant un sommet u ou une arête sw, telle que $P_1 \preccurlyeq Q_1$ et $P_2 \preccurlyeq Q_2$.*

Démonstration. Soit $\bar{P}$ et $\bar{Q}$ des extensions respectives de P et Q telles que $\bar{P} \leq \bar{Q}$. Si v correspond au i-ème sommet de $\bar{Q}$ alors nous choisissons u comme étant le i-ème sommet de $\bar{P}$. Si u n'est pas présent dans P, c'est qu'il est issu d'une arête sw de P ayant subi une subdivision. Si $(\bar{P}_1, \bar{P}_2)$ est une séparation de $\bar{P}$ suivant u et $(\bar{Q}_1, \bar{Q}_2)$ est une séparation de $\bar{Q}$ suivant v, alors nous avons $\bar{P}_1 \leq \bar{Q}_1$ et $\bar{P}_2 \leq \bar{Q}_2$. Finalement, la séparation (P_1, P_2) de P suivant u ou sw ainsi que la séparation (Q_1, Q_2) de Q suivant v, sont telles que $P_1 \preccurlyeq Q_1$ et $P_2 \preccurlyeq Q_2$. □

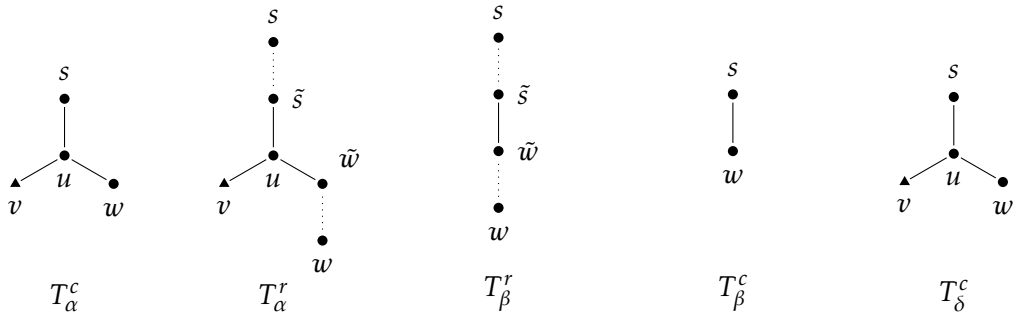
Ces nouvelles propriétés sont alors suffisantes pour prouver que la Procédure d'Introduction calcule un ensemble complet de (k,q)-caractéristiques, et ce en temps constant. À l'image de ce que nous avons fait pour la Procédure d'Oubli, nous utilisons trois lemmes.

Lemme 5.79. *Soit α un nœud d'introduction. La Procédure d'Introduction de Type 3 garantit que, pour chaque (k,q)-caractéristique C_δ qu'elle calcule, il existe un (k,q)-arbre de partitions de A_α dont la caractérisation suivant X_α est C_δ.*

Démonstration. Soit β le fils de α dans $\mathcal{T}$. Soit C_β la (k,q)-caractéristique de A_β restreint à X_β, sur laquelle nous avons appliqué une Procédure d'Introduction de Type 3 en vue d'obtenir C_δ ; nous appelons sw l'arête qui a été subdivisée. Par hypothèse, FSC_β est un ensemble complet de (k,q)-caractéristiques pour A_β, il existe donc un (k,q)-arbre de partitions T_β de A_β dont la caractéristique est C_β. D'après la Procédure de Caractérisation, l'arête sw représente un chemin $P_\beta(s,w)$ dans T_β, et elle correspond



(a) Dans le cas où w est une feuille.



(b) Dans le cas où w n'est pas une feuille.

FIGURE 5.3. : Subdivisions des arêtes sw et $\tilde{s}\tilde{w}$ utilisées lors de la preuve de la Procédure d'Insertion de Type 3. Les arêtes en pointillés représentent des chemins de longueur potentiellement nulle et les sommets triangulaires représentent des feuilles.

à une arête de $P_\beta(s,w)$ que nous appelons $\tilde{s}\tilde{w}$. Nous définissons ensuite l'arbre T_α en subdivisant l'arête $\tilde{s}\tilde{w}$ de T_β en deux arêtes $\tilde{s}u$ et $u\tilde{w}$, puis en attachant au nouveau sommet u une feuille v en bijection avec $\bullet$. Par définition, nous avons $A_\alpha = A_\beta \cup \{\bullet\}$, et donc T_α est un arbre de partitions de A_α. Ce que nous cherchons à démontrer c'est que $C_\delta = \tau_\alpha(T_\alpha)$ et que T_α a une largeur q-ramifiée suivant ϕ au plus égale à k. Pour cela, nous considérons $C_\alpha = \tau_\alpha(T_\alpha)$ et nous montrons que $C_\alpha = C_\delta$.

Comme pour la preuve de la Procédure d'Oubli, nous aurons besoin des notations suivantes. Soit T^r_α le sous-arbre de T_α restreint à X_α et soit T^r_β le sous-arbre de T_β restreint à X_β. Nous rappelons que C_α est une notation élégante pour $(T^c_\alpha, \ell_\alpha, S_\alpha, a_\alpha, d_\alpha, b_\alpha)$, et qu'il en va de même pour C_β et C_δ.

Pour montrer que $C_\alpha = C_\delta$, nous montrons que chaque composante de C_α est identique à la composante correspondante dans C_δ, et ce par étapes succéssives afin de faciliter la lecture de la preuve. Plusieurs de ces étapes sont elles mêmes divisées en deux, suivant si w est une feuille ou non (les subdivisions des arêtes sw et $\tilde{s}\tilde{w}$ ainsi étudiées sont illustrées FIGURE 5.3).

Avant de commencer, nous observons que, si w est une feuille, alors la Procédure de Caractérisation nous indique que s est le père de w dans T^r_β. Il en découle que $s = \tilde{s}$ et que $w = \tilde{w}$ et, par conséquent, $\tilde{w}$ est une feuille. Réciproquement, si $\tilde{w}$ est une feuille alors w

est une feuille. Une seconde observation nous indique que le nombre de ramifications de $\tilde{s}$ augmente si et seulement si w est une feuille (auquel cas nous avons $s = \tilde{s}$).

Lemme 5.79.1. *Les ensembles S_α et S_δ sont égaux.*

Par la Procédure de Caractérisation, S_α contient les sommets de T_α^r qui sont soit les pères des feuilles de T_α^r, soit des sommets de ramification dans $cp(T_\alpha)$. De même, S_β contient les sommets de T_β^r qui sont soit les pères des feuilles de T_β^r, soit des sommets de ramification dans $cp(T_\beta)$. Par construction ainsi que grâce aux observations précédentes, nous avons que $S_\alpha \setminus \{s,u\} = S_\beta \setminus \{s\}$. Notons de plus que, si s est un élément de S_α alors s est également un élément de S_β. Dans la Procédure d'Introduction de Type 3, nous avons fait en sorte que $S_\delta \setminus \{s,u\} = S_\beta \setminus \{s\}$ et, comme u est à la fois présent dans S_α et dans S_δ, nous avons que $S_\alpha \setminus \{s\} = S_\delta \setminus \{s\}$. Pour montrer que S_α et S_δ sont égaux, il nous faut donc montrer que s appartient à S_α si et seulement si s appartient également à S_δ.

Dans le cas où w n'est pas une feuille, nous avons que s appartient à S_α, si et seulement si s est également un élément de S_β. De même, s est un élément de S_δ si et seulement si il est également un élément de S_β. Dans ce cas là, nous avons que s appartient à S_α si et seulement si s appartient à S_δ.

Maintenant, si w est une feuille, alors s est un élément de S_β, car il est le père de w dans T_β^r. Dans le cas où s est le père d'une seconde feuille dans T_β^r, alors s fait également partie de S_α ainsi que de S_δ. Si ce n'est pas le cas, mais que s à au moins une ramification dans T_β (ce qui se traduit par $b_\beta(s) \neq 0$) alors, par la Procédure d'Introduction de Type 3, nous avons que s est un élément de S_δ. De plus, par construction, la subdivision de sw augmente le nombre de ramifications de s, ce qui en fait un sommet de ramification dans T_α, forçant alors sa présence dans S_α. Finalement, si w est le seul descendant de s dans T_β^r et que tous les descendant de s dans T_β sont de feuilles (c'est-à-dire que $b_\beta(s) = 0$), alors s n'apparait ni dans S_α ni dans S_δ.

Lemme 5.79.2. *Les arbres T_α^c et T_δ^c sont isomorphes et les fonctions ℓ_α et ℓ_δ sont égales.*

Soient x et y deux éléments de S_α et soit $P_\alpha^c(x,y)$ le tronçon de T_α^c reliant x à y, avec x l'ancêtre de y dans T_α^c. Comme S_α et S_δ sont égaux, nous pouvons définir $P_\delta^c(x,y)$, le tronçon reliant x à y dans T_δ^c. Nous considérons aussi $P_\alpha^r(x,y)$ qui est le chemin étiqueté reliant x à y dans T_α^r, et la Procédure de Caractérisation nous donne $P_\alpha^c(x,y) = \tau(P_\alpha^r(x,y))$.

Pour montrer que $T_\alpha^c \cong T_\delta^c$, il nous suffit de montrer que $P_\alpha^c(x,y)$ et $P_\delta^c(x,y)$ sont formés par la même séquence de sommets et que leurs racines identifient le même sommet (ce dernier point est trivial). Et, pour montrer que ℓ_α et ℓ_δ sont égales, il nous suffit de montrer que les étiquettes attribuées à chaque sommet z ainsi qu'à chaque arête e de $P_\alpha^c(x,y)$ sont respectivement égales aux étiquettes attribuées aux correspondants de z et e dans $P_\delta^c(x,y)$. Nous observons alors plusieurs cas suivant si u est une extrémité de $P_\alpha^c(x,y)$ ou non.

Nous choisissons de traiter en premier le cas où u n'est pas une extrémité de $P_\alpha^c(x,y)$. Nous définissons alors $P_\beta^r(x,y)$ qui est le chemin étiqueté reliant x à y dans T_β^r. La Procédure de Caractérisation nous indique alors que $P_\beta^c(x,y) = \tau(P_\beta^r(x,y))$ est un tronçon de T_β^c. Il en découle que l'ensemble des sommets internes ainsi que l'ensemble des arêtes des chemins $P_\alpha^r(x,y)$, $P_\beta^r(x,y)$, $P_\alpha^c(x,y)$, $P_\beta^c(x,y)$ et $P_\delta^c(x,y)$ induisent une unique bipartition

de X_α, que nous appelons $\mathcal{Q}$. Cette bipartition nous permet de définir la fonction $\lambda : \mathbb{N} \to \mathbb{N}$, qui à un entier i associe la valeur $I_\phi(i, \mathcal{Q}, \bullet)$, laquelle respecte la deuxième règle de compatibilité entre ϕ et $(\mathcal{T}, \mathcal{X})$. L'insertion de $\bullet$ dans T_β définit alors l'égalité $P^r_\alpha(x,y) = \lambda(P^r_\beta(x,y))$ et la Procédure d'Introduction nous donne $P^c_\delta(x,y) = \tau(\lambda(P^c_\beta(x,y)))$. En résumé nous avons les égalités suivantes :

$$\begin{aligned}
P^c_\alpha(x,y) &= \tau(P^r_\alpha(x,y)) && \text{Procédure de Caractérisation}\\
&= \tau(\lambda(P^r_\beta(x,y))) && \text{Insertion de } \bullet \text{ dans } T_\beta\\
&= \lambda(\tau(P^r_\beta(x,y))) && \text{Propriété 5.74}\\
&= \lambda(P^c_\beta(x,y)) && \text{Procédure de Caractérisation}\\
&= P^c_\delta(x,y) && \text{Procédure d'Introduction}
\end{aligned}$$

Nous traitons maintenant les trois tronçons qui admettent u en temps qu'une de leurs extrémités. C'est-à-dire que nous vonlons montrer que $P^c_\alpha(x,u) = P^c_\delta(x,u)$, $P^c_\alpha(u,y) = P^c_\delta(u,y)$ et $P^c_\alpha(u,v) = P^c_\delta(u,v)$. La vérification de la dernière égalité revient à montrer que nous attribuons la bonne étiquette à l'arête uv lors de la Procédure d'Introduction de Type 3, ce qui est facilement faisable en rappelant la première règle de compatibilité entre ϕ et $(\mathcal{T}, \mathcal{X})$. Ce qui nous donne :

$$\ell_\delta(uv) = \phi_{A_\alpha}(\{X_\beta, \{\bullet\}\}) = \phi_{A_\alpha}(\mathcal{Q}_{uv}) = \phi_{A_\alpha}(\mathcal{P}_{uv}) = \ell_\alpha(uv),$$

avec $\mathcal{Q}_{uv}$ et $\mathcal{P}_{uv}$ les bipartitions respectives de X_α et A_α induites par uv.

Il nous faut donc montrer que $P^c_\alpha(x,u) = P^c_\delta(x,u)$ et $P^c_\alpha(u,y) = P^c_\delta(u,y)$. Le problème qui survient ici est que le sommet u n'apparait pas dans T_β, nous ne pouvons donc pas définir les chemins $P^r_\beta(x,u)$ et $P^r_\beta(u,y)$ ainsi que leurs contractions respectives. Néanmoins, en considérant le chemin $P^r_\alpha(x,u) \odot P^r_\alpha(u,y)$, nous pouvons définir le chemin $\bar{P}^r_\beta(x,y) = \bar{P}^r_\beta(x,u) \odot \bar{P}^r_\beta(u,y)$, qui résulte de la subdivision de l'arête $\tilde{s}\tilde{w}$ dans le chemin $P^r_\beta(x,y)$. De même, nous définissons le chemin $\bar{P}^c_\beta(x,y) = \bar{P}^c_\beta(x,u) \odot \bar{P}^c_\beta(u,y)$, qui résulte de la subdivision de de l'arête sw dans le chemin $\tau(P^r_\beta(x,y))$. D'après la Propriété 5.77 nous avons $\tau(\bar{P}^c_\beta(x,u)) = \tau(\bar{P}^r_\beta(x,u))$ et $\tau(\bar{P}^c_\beta(u,y)) = \tau(\bar{P}^r_\beta(u,y))$. De plus, les chemins $P^r_\alpha(x,u)$, $\bar{P}^r_\beta(x,u)$, $P^c_\alpha(x,u)$, $\bar{P}^c_\beta(x,u)$ et $P^c_\delta(x,u)$ induisent la même bipartition Q_1 de X_α, et les chemins $P^r_\alpha(u,y)$, $\bar{P}^r_\beta(u,y)$, $P^c_\alpha(u,y)$, $\bar{P}^c_\beta(u,y)$ et $P^c_\delta(u,y)$ induisent la même bipartition Q_2 de X_α. Nous définissons alors les fonctions $\lambda_1 : \mathbb{N} \to \mathbb{N}$ et $\lambda_2 : \mathbb{N} \to \mathbb{N}$, telles que $\lambda_1(i) = I_\phi(i, \mathcal{Q}_1, \bullet)$ et $\lambda_2(i) = I_\phi(i, \mathcal{Q}_2, \bullet)$, et nous obtenons ainsi :

$$\begin{aligned}
P^c_\alpha(x,u) &= \tau(P^r_\alpha(x,u)) && \text{Procédure de Caractérisation} && \tau(P^r_\alpha(u,y)) = P^c_\alpha(u,y)\\
&= \tau(\lambda_1(\bar{P}^r_\beta(x,u))) && \text{Insertion de } \bullet \text{ dans } T_\beta && \tau(\lambda_2(\bar{P}^r_\beta(u,y))) =\\
&= \lambda_1(\tau(\bar{P}^r_\beta(x,u))) && \text{Propriété 5.74} && \lambda_2(\tau(\bar{P}^r_\beta(u,y))) =\\
&= \lambda_1(\tau(\bar{P}^c_\beta(x,u))) && \text{Propriété 5.77} && \lambda_2(\tau(\bar{P}^r_\beta(u,y))) =\\
&= \tau(\lambda_1(\bar{P}^c_\beta(x,u))) && \text{Propriété 5.74} && \tau(\lambda_2(\bar{P}^c_\beta(u,y))) =\\
&= P^c_\delta(x,u) && \text{Procédure d'Introduction} && P^c_\delta(u,y) =
\end{aligned}$$

Notons que, dans le cas où s n'est pas dans $S_\alpha = S_\delta$, nous sommes face à la concaténation $\bar{P}^r_\beta(x,u) = \bar{P}^r_\beta(x,s) \odot \bar{P}^r_\beta(s,u)$ que nous pouvons traiter en utilisant Propriété 5.68.

Il nous reste à montrer que les étiquettes des extrémités de $P^c_\alpha(x,y)$ et $P^c_\delta(x,y)$ sont respectivement égales. En d'autres termes, pour chaque sommet z de $S_\alpha = S_\delta$, nous devons montrer que $\ell_\alpha(z) = \ell_\delta(z)$. Ce résultat découle de la deuxième règle de compatibilité entre ϕ et $(\mathcal{T},\mathcal{X})$. En effet, quand $z \neq u$, nous avons :

$$\ell_\delta(z) = I_\phi(\ell_\beta(z), \mathcal{Q}_z, \bullet) = I_\phi(\phi_{A_\beta}(\mathcal{P}_z \cap X_\beta), \mathcal{P}_z \cap X_\alpha, \bullet) = \phi_{A_\alpha}(\mathcal{P}_z) = \ell_\alpha(z),$$

avec $\mathcal{Q}_z$ et $\mathcal{P}_z$ les partitions respectives de X_α et A_α induites par z. Et, quand $z = u$:

$$\ell_\delta(u) = I_\phi(\ell_\beta(sw), \mathcal{Q}_u, \bullet) = I_\phi(\phi_{A_\beta}(\mathcal{P}_u \cap X_\beta), \mathcal{P}_u \cap X_\alpha, \bullet) = \phi_{A_\alpha}(\mathcal{P}_u) = \ell_\alpha(u),$$

avec $\mathcal{Q}_u$ et $\mathcal{P}_u$ les partitions respectives de X_α et A_α induites par u.

Lemme 5.79.3. *Les entiers a_α et a_δ sont égaux.*

Ni la subdivision de $\tilde{s}\tilde{w}$, ni l'ajout de $\bullet$ dans T^r_β n'entrainent une redéfinission de la racine dans T^r_α, nous avons donc $\mathrm{dr}(r, r^r_\alpha) = \mathrm{dr}(r, r^r_\beta)$, ce qui se traduit par $a_\alpha = a_\beta$. D'après la Procédure d'Introduction nous avons $a_\delta = a_\beta$. Les égalités souhaitées sont ainsi vérifiées.

Lemme 5.79.4. *La fonction b_α est égale à la fonction b_δ.*

Dans un premier temps, nous considérons le cas où w n'est pas une feuille. Le sommet u que nous venons d'introduire dans T_α à donc deux fils, v et $\tilde{w}$, et nous savons que v est une feuille de T_α. Par conséquent, u à exactement une ramification dans T_α. Par la Procédure de Caractérisation, nous avons que $b_\alpha(u) = 1$ et, en faisant appel à la Procédure d'Introduction de Type 3, nous déduisons que $b_\alpha(u) = b_\delta(u)$. Finalement, pour tout élément x de S_β, la subdivision de $\tilde{s}\tilde{w}$ ne change pas le nombre de ramifications de x dans T_α (y compris pour s, si ce dernier appartient à S_β), nous avons donc $b_\alpha(x) = b_\beta(x)$. La Procédure d'Introduction de Type 3 nous indique alors que $b_\alpha(x) = b_\delta(x)$.

Dans le cas où w est une feuille, le sommet u n'admet aucune ramification dans T_α, en nous référant à la Procédure d'Introduction de Type 3, nous avons que $b_\alpha(u) = b_\delta(u) = 0$. Parallèlement, le sommet s se voit doté d'une nouvelle ramification dans T_α, nous avons donc $b_\alpha(s) = \min\{2, b_\beta(s) + 1\}$, qui se trouve être la même valeur que celle que nous avons affectée à $b_\delta(s)$. Enfin, pour tout élément x de S_β, différent de s cette fois-ci, la subdivision de $\tilde{s}\tilde{w}$ ne change pas le nombre de ramifications de x dans T_α, nous avons donc $b_\alpha(x) = b_\beta(x)$. Et, d'après la Procédure d'Introduction de Type 3 nous avons $b_\alpha(x) = b_\delta(x)$.

Lemme 5.79.5. *La fonction d_α est égale à la fonction d_δ.*

Si w n'est pas une feuille, u admet une unique ramification dans T_α, par conséquent sa hauteur de ramification est égale à celle de son plus proche descendant dans T_α appartenant à S_α, que nous appelons z. Nous avons donc $d_\alpha(u) = d_\beta(z)$ et, comme $T^c_\alpha \cong T^c_\delta$ et $S_\alpha = S_\delta$, la Procédure d'Introduction de Type 3 nous indique que $d_\delta(u) = d_\beta(z)$. Le fait que le nombre de ramifications de s n'augmente pas implique que, pour tout sommet interne x de T^c_α différent de u, nous avons $d_\alpha(x) = d_\beta(x)$. Ceci est en parfait accord avec la Procédure d'Introduction de Type 3, qui nous donne $d_\delta(x) = d_\beta(x)$.

Dans le cas où w est une feuille, u n'admet aucune ramification dans T_α nous avons donc $d_\alpha(u) = d_\delta(u) = 0$. Comme nous l'avons fait remarquer dans la Procédure d'Introduction de Type 3, le fait que s gagne une ramification supplémentaire peut engendrer une augmentation de sa hauteur de ramification ainsi que de celles de ses ancêtres dans T_α^c. Les valeurs que nous obtenons pour $d_\alpha(s)$ et pour $d_\alpha(x)$, où x est un sommet de T_α^c ancêtre de s dans T_α^c, sont tout à fait comparables à celles de $d_\delta(s)$ et $d_\delta(x)$ définies lors de la Procédure d'Introduction de Type 3. Les valeurs associées aux sommets de S_α que nous n'avons pas encore pris en compte sont elles aussi compatibles.

Lemme 5.79.6. *La largeur q-ramifiée suivant ϕ de T_α est au plus k.*

La hauteur de ramification de r_α peut être définie par $\mathrm{hr}_{T_\alpha}(r_\alpha) = \mathrm{dr}_{T_\alpha}(r_\alpha, r_\alpha^c) + \mathrm{hr}_{T_\alpha}(r_\alpha^c)$. Par la Procédure de Caractérisation, nous avons que $\mathrm{hr}_{T_\alpha}(r_\alpha) = a_\alpha + d_\alpha(r_\alpha^c)$. D'après les Lemmes 5.79.3 et 5.79.5, nous pouvons écrire $\mathrm{hr}_{T_\alpha}(r_\alpha) = a_\delta + d_\delta(r_\alpha^c)$ et, comme C_δ est une (k,q)-caractéristique, il en découle que $\mathrm{hr}_{T_\alpha}(r_\alpha) \leq q$. De plus, pour tout sous-chemin P de T_α joignant r_α à une feuille de T_α, et tel que r_α^c n'est pas un sommet de P, nous savons que P est un sous-chemin de T_β. Étant donné que T_β est un (k,q)-arbre de partitions, nous pouvons en conclure que T_α est q-ramifié.

Soit x un sommet de $cp(T_\alpha)$ qui n'apparait pas dans T_α^r et soit e une arête de T_α qui n'apparait pas dans T_α^r. Par construction nous avons que x est un sommet de $cp(T_\beta)$ qui n'apparait pas dans T_β^r et que e est une arête de T_β qui n'apparait pas dans T_β^r. Et, comme T_β est un (k,q)-arbre de partitions et que la Procédure d'Introduction de Type 3 ne change pas les largeurs suivant ϕ associées à x et e dans T_α, nous avons que $\phi_{A_\alpha}(\mathcal{P}_x) \leq k$ et $\phi_{A_\alpha}(\mathcal{P}_e) \leq k$, où $\mathcal{P}_x$ est la partition de A_α induite par x et $\mathcal{P}_e$ est la bipartition de A_α induite par e. De plus, pour chaque sommet x de $cp(T_\alpha^r)$ et chaque arête e de T_α^r, nous avons $\ell_\alpha(x) \leq k$ et $\ell_\alpha(e) \leq k$. En faisant appel à la deuxième règle de compatibilité entre ϕ et $(\mathcal{T}, \mathcal{X})$, nous avons que $\phi_{A_\alpha}(\mathcal{P}_x) \leq k$ et $\phi_{A_\alpha}(\mathcal{P}_e) \leq k$, où $\mathcal{P}_x$ est la partition de A_α induite par x et $\mathcal{P}_e$ est la bipartition de A_α induite par e. Par conséquent, la largeur suivant ϕ de T_α est au plus k.

En conclusion, T_α est un (k,q)-arbre de partitions de A_α admettant C_δ comme caractéristique. □

Corollaire 5.80. *Soit α un nœud d'introduction. La Procédure d'Introduction garantit que, pour chaque (k,q)-caractéristique C_δ de FSC_α, il existe un (k,q)-arbre de partition de A_α dont la caractérisation suivant X_α est C_δ.*

Démonstration. Il nous suffit d'adapter la preuve du Lemme 5.79 aux Procédures d'Introductions de Types 1 et 2. Cela se fait assez simplement étant donné le peu de modifications que nous apportons. □

Lemme 5.81. *Soit α un nœud d'introduction. La Procédure d'Introduction garantit que, pour chaque (k,q)-arbre de partitions T_α de A_α, il existe un (k,q)-arbre de partitions $\check{T}_\alpha \preccurlyeq_\alpha T_\alpha$ tel que $\tau_\alpha(\check{T}_\alpha) \in FSC_\alpha$.*

Démonstration. Soit β le fils de α dans $\mathcal{T}$ et soit FSC_β l'ensemble complet de caractéristiques associé au nœud β. Nous appelons T_β la restriction de T_α à $A_\beta = A_\alpha \setminus \{\bullet\}$. En d'autre termes, si v est la feuille de T_α en bijection avec $\bullet$, alors T_β est obtenu en supprimant v de T_α et, dans le cas où r_α est le père de v, qu'il est de degré 2 et qu'il induit exactement une ramification, alors nous supprimons également r_α de T_α. Par définition, T_β est un (k,q)-arbre de partitions de A_β, il existe donc un (k,q)-arbre de partitions $\check{T}_\beta \preccurlyeq_\beta T_\beta$ tel que $\tau_\beta(\check{T}_\beta) \in FSC_\beta$. Nous montrons qu'il est alors possible de construire un (k,q)-arbre de partitions $\check{T}_\alpha$ à partir de $\check{T}_\beta$, tel que $\check{T}_\alpha \preccurlyeq_\alpha T_\alpha$ et que $\tau_\alpha(\check{T}_\alpha) \in FSC_\alpha$.

Nous appelons u le père de v dans T_α. D'après les Procédures d'Introduction de Types 1 et 3, u est un sommet de degré au moins 3 dans T_α^r. Dans le cas d'une Procédure d'Introduction de Type 2, u joue le rôle de racine dans T_α et nous appelons w son fils dans T_α^r qui, par construction est le père d'au moins deux sommets dans T_α^r.

Construction 1. Le sommet u est présent dans T_β^t, il est donc également présent dans $\check{T}_\beta^t$ et $\check{T}_\beta$. Nous construisons alors $\check{T}_\alpha$ en ajoutant une feuille $\check{v}$, en bijection avec $\bullet$, en temps que fils de u dans $\check{T}_\beta$. ■

Construction 2. Le sommet u n'est pas présent dans T_β^t et il est racine de T_α. Soit w le second fils de u dans T_α. Le sommet w est donc racine de T_β^t et donc de $\check{T}_\beta^t$ ainsi que de $\check{T}_\beta$. Nous construisons alors $\check{T}_\alpha$ en ajoutant un sommet $\check{u}$ en temps que père de w dans $\check{T}_\beta$, puis en ajoutant une feuille $\check{v}$, en bijection avec $\bullet$, en temps que fils de $\check{u}$. ■

Construction 3. Le sommet u n'est pas présent dans $T_\beta^t \cong \check{T}_\beta^t$ et il n'est pas racine de T_α. Le sommet u n'est donc pas le père d'une feuille de T_β et son degré dans T_β est égal à 2. Soient $P_\beta(s,u)$ et $P_\beta(u,w)$ les sous-chemins de T_β tels que sw est une arête de T_β^t. Ces deux sous-chemins forment alors une séparation suivant u de $P_\beta(s,w)$. D'après la Propriété 5.78 il existe une séparation de $\check{P}_\beta(s,w)$ suivant un sommet $\check{u}$ ou une arête $\check{s}\check{w}$ (dont la subdivision engendre le sommet $\check{u}$), telle que $\check{P}_\beta(s,\check{u}) \preccurlyeq P_\beta(s,u)$ et $\check{P}_\beta(\check{u},w) \preccurlyeq P_\beta(u,w)$. Nous construisons alors $\check{T}_\alpha$ en ajoutant un sommet $\check{v}$, en bijection avec $\bullet$, en temps que fils de $\check{u}$ dans $\check{T}_\beta$ (si $\check{u}$ n'existe pas nous le créons en subdivisant l'arête $\check{s}\check{w}$). ■

Notons que, par construction, $V(T_\alpha) \setminus V(T_\alpha^r) = V(T_\beta) \setminus V(T_\beta^r)$ et, pour chacune des trois constructions précédentes, nous avons $V(\check{T}_\alpha) \setminus V(\check{T}_\alpha^r) = V(\check{T}_\beta) \setminus V(\check{T}_\beta^r)$.

Lemme 5.81.1. *La Construction 1 garantit que $T_\alpha^t \cong \check{T}_\alpha^t$.*

La suppression de v dans T_α se traduit par la suppression de v dans T_α^t. Par conséquent, T_β^t est isomorphe à T_α^t privé de v. En parallèle, l'ajout de $\check{v}$ en temps que fils de u dans $\check{T}_\beta$ se traduit par l'ajout de $\check{v}$ en temps que fils de u dans $\check{T}_\beta^t$. Donc, $\check{T}_\alpha^t$ est isomorphe à $\check{T}_\beta^t$ auquel nous avons ajouter $\check{v}$ en temps que fils de u. L'isomorphisme de T_α^t et $\check{T}_\alpha^t$ apparît alors assez clairement.

Lemme 5.81.2. *La Construction 2 garantit que $T_\alpha^t \cong \check{T}_\alpha^t$.*

La suppression de u et v dans T_α se traduit par la suppression de u et v dans T_α^t. Par conséquent, T_β^t est isomorphe à T_α^t privé de u et v. En parallèle, l'ajout de $\check{u}$ en temps que père de w dans $\check{T}_\beta$ suivi de l'ajout de $\check{v}$ en temps que fils de $\check{u}$ se traduisent par l'ajout de $\check{u}$ en temps que père de w dans $\check{T}_\beta^t$ suivi de l'ajout de $\check{v}$ en temps que fils de $\check{u}$. Donc, $\check{T}_\alpha^t$ est isomorphe à $\check{T}_\beta^t$ auquel nous avons ajouter $\check{u}$ en temps que père de w puis $\check{v}$ en temps que fils de $\check{u}$. L'isomorphisme de T_α^t et $\check{T}_\alpha^t$ apparît alors assez clairement.

Lemme 5.81.3. *La Construction 3 garantit que* $T_\alpha^t \cong \check{T}_\alpha^t$.

La suppression de v dans T_α se traduit par la suppression de v puis le lissage de u dans T_α^t. Par conséquent, T_β^t est isomorphe à T_α^t privé de v suivi du lissage de u. En parallèle, la subdivision de sw dans $\check{T}_\beta$ suivie de l'ajout de $\check{v}$ en temps que fils de $\check{u}$ se traduisent par la subdivision de sw dans $\check{T}_\beta^t$ suivie de l'ajout de $\check{v}$ en temps que fils de $\check{u}$. Donc, $\check{T}_\alpha^t$ est isomorphe à $\check{T}_\beta^t$ dans lequel nous avons subdiviser l'arête sw puis ajouter le sommet $\check{v}$ en temps que fils de $\check{u}$. L'isomorphisme de T_α^t et $\check{T}_\alpha^t$ apparît alors assez clairement.

Lemme 5.81.4. *La Construction 1 garantit que* $\mathrm{dr}_{T_\alpha}(r_\alpha, r_\alpha^r) = \mathrm{dr}_{\check{T}_\alpha}(\check{r}_\alpha, \check{r}_\alpha^r)$.

La suppression de v dans T_α n'induit pas de changement de racine entre T_α et T_β, ni même entre T_α^t et T_β^t. De plus, l'ajout de $\check{v}$ dans $\check{T}_\beta$ n'induit pas non plus de changement de racine entre $\check{T}_\beta$ et $\check{T}_\alpha$, ni même entre $\check{T}_\beta^t$ et $\check{T}_\alpha^t$. Nous avons donc $\mathrm{dr}_{T_\alpha}(r_\alpha, r_\alpha^r) = \mathrm{dr}_{T_\beta}(r_\beta, r_\beta^r) = \mathrm{dr}_{\check{T}_\beta}(\check{r}_\beta, \check{r}_\beta^r) = \mathrm{dr}_{\check{T}_\alpha}(\check{r}_\alpha, \check{r}_\alpha^r)$.

Lemme 5.81.5. *La Construction 2 garantit que* $\mathrm{dr}_{T_\alpha}(r_\alpha, r_\alpha^r) = \mathrm{dr}_{\check{T}_\alpha}(\check{r}_\alpha, \check{r}_\alpha^r)$.

Par construction, nous avons $r_\alpha = r_\alpha^r$ et $\check{r}_\alpha = \check{r}_\alpha^r$ et, par conséquent $\mathrm{dr}_{T_\alpha}(r_\alpha, r_\alpha^r) = \mathrm{dr}_{\check{T}_\alpha}(\check{r}_\alpha, \check{r}_\alpha^r) = 0$.

Lemme 5.81.6. *La Construction 3 garantit que* $\mathrm{dr}_{T_\alpha}(r_\alpha, r_\alpha^r) = \mathrm{dr}_{\check{T}_\alpha}(\check{r}_\alpha, \check{r}_\alpha^r)$.

La suppression de v dans T_α n'induit pas de changement de racine entre T_α et T_β. De plus la suppression de v ainsi que le lissage de u dans T_α^t n'induisent pas de changement de racine entre T_α^t et T_β^t. Parallèlement, l'ajout de $\check{v}$ et la possible subdivision de $\check{s}\check{w}$ dans $\check{T}_\beta$ n'induisent pas non plus de changement de racine entre $\check{T}_\beta$ et $\check{T}_\alpha$. De même, l'ajout de $\check{v}$ et la subdivision de sw dans $\check{T}_\beta^t$ n'induisent pas de changement de racine entre $\check{T}_\beta^t$ et $\check{T}_\alpha^t$. Nous avons donc $\mathrm{dr}_{T_\alpha}(r_\alpha, r_\alpha^r) = \mathrm{dr}_{T_\beta}(r_\beta, r_\beta^r) = \mathrm{dr}_{\check{T}_\beta}(\check{r}_\beta, \check{r}_\beta^r) = \mathrm{dr}_{\check{T}_\alpha}(\check{r}_\alpha, \check{r}_\alpha^r)$.

Lemme 5.81.7. *La Construction 1 garantit que, pour tout sommet interne x de $T_\alpha^t \cong \check{T}_\alpha^t$, nous avons* $\mathrm{hr}_{T_\alpha}(x) = \mathrm{hr}_{\check{T}_\alpha}(x)$ *et* $\mathrm{nr}_{T_\alpha}(x) = \mathrm{nr}_{\check{T}_\alpha}(x)$.

La suppression de v dans T_α ne change ni la hauteur de ramification de x ni le nombre de ramifications de x entre T_α et T_β. De plus l'ajout de $\check{v}$ dans $\check{T}_\beta$ ne change ni la hauteur de ramification de x ni le nombre de ramifications de x entre $\check{T}_\beta$ et $\check{T}_\alpha$. Par conséquent, $\mathrm{hr}_{T_\alpha}(x) = \mathrm{hr}_{T_\beta}(x) = \mathrm{hr}_{\check{T}_\beta}(x) = \mathrm{hr}_{\check{T}_\alpha}(x)$ et $\mathrm{nr}_{T_\alpha}(x) = \mathrm{nr}_{T_\beta}(x) = \mathrm{nr}_{\check{T}_\beta}(x) = \mathrm{nr}_{\check{T}_\alpha}(x)$.

Lemme 5.81.8. *La Construction 2 garantit que, pour tout sommet interne x de $T_\alpha^t \cong \check{T}_\alpha^t$, nous avons* $\mathrm{hr}_{T_\alpha}(x) = \mathrm{hr}_{\check{T}_\alpha}(x)$ *et* $\mathrm{nr}_{T_\alpha}(x) = \mathrm{nr}_{\check{T}_\alpha}(x)$.

Si x est différent de u, la suppression de u et v dans T_α ne change ni la hauteur de ramification de x ni le nombre de ramifications de x entre T_α et T_β et, si $x = u$ alors $\mathrm{hr}_{T_\alpha}(x) = \mathrm{hr}_{T_\alpha}(w)$ et $\mathrm{nr}_{T_\alpha}(x) = 1$. De même si x est différent de $\check{u}$, l'ajout de $\check{u}$ et $\check{v}$ dans $\check{T}_\beta$ ne change ni la hauteur de ramification de x ni le nombre de ramifications de x entre $\check{T}_\beta$ et $\check{T}_\alpha$ et, si $x = \check{u}$ alors $\mathrm{hr}_{\check{T}_\alpha}(x) = \mathrm{hr}_{\check{T}_\alpha}(w)$ et $\mathrm{nr}_{\check{T}_\alpha}(x) = 1$. Il s'en suit que $\mathrm{hr}_{T_\alpha}(x) = \mathrm{hr}_{T_\beta}(x) = \mathrm{hr}_{\check{T}_\beta}(x) = \mathrm{hr}_{\check{T}_\alpha}(x)$ et $\mathrm{nr}_{T_\alpha}(x) = \mathrm{nr}_{T_\beta}(x) = \mathrm{nr}_{\check{T}_\beta}(x) = \mathrm{nr}_{\check{T}_\alpha}(x)$, pour $x \neq u$ ainsi que $\mathrm{hr}_{T_\alpha}(x) = \mathrm{hr}_{T_\beta}(w) = \mathrm{hr}_{\check{T}_\beta}(w) = \mathrm{hr}_{\check{T}_\alpha}(x)$ et $\mathrm{nr}_{T_\alpha}(x) = \mathrm{nr}_{\check{T}_\alpha}(x) = 1$ pour $x = u$.

Lemme 5.81.9. *La Construction 3 garantit que, pour tout sommet interne x de $T_\alpha^t \cong \check{T}_\alpha^t$, nous avons* $\mathrm{hr}_{T_\alpha}(x) = \mathrm{hr}_{\check{T}_\alpha}(x)$ *et* $\mathrm{nr}_{T_\alpha}(x) = \mathrm{nr}_{\check{T}_\alpha}(x)$.

Comme u n'est pas le père d'une feuille de T_β, l'arête sw de $T_\beta^t \cong \check{T}_\beta^t$ est telle que w n'est pas une feuille de $\check{T}_\beta$. Par conséquent, $\check{w}$ n'est pas non plus une feuille de $\check{T}_\beta$, ainsi la subdivision de $\check{s}\check{w}$ dans $\check{T}_\beta$ n'augmente pas le nombre de ramifications de $\check{s}$.

La suppression de v dans T_α ne change ni la hauteur de ramification de x ni le nombre de ramifications de x entre T_α et T_β. De plus la subdivision de $\check{s}\check{w}$ et l'ajout de $\check{v}$ ne changent ni la hauteur de ramification de x ni le nombre de ramifications de x entre $\check{T}_\beta$ et $\check{T}_\alpha$. Par conséquent, $\mathrm{hr}_{T_\alpha}(x) = \mathrm{hr}_{T_\beta}(x) = \mathrm{hr}_{\check{T}_\beta}(x) = \mathrm{hr}_{\check{T}_\alpha}(x)$ et $\mathrm{nr}_{T_\alpha}(x) = \mathrm{nr}_{T_\beta}(x) = \mathrm{nr}_{\check{T}_\beta}(x) = \mathrm{nr}_{\check{T}_\alpha}(x)$.

Lemme 5.81.10. *La Construction 1 garantit que, pour toute arête xy de $T_\alpha^t \cong \check{T}_\alpha^t$, nous avons $\check{P}_\alpha(x,y) \preccurlyeq P_\alpha(x,y)$.*

Si xy n'identifie pas l'arête uv de T_α^t, laquelle correspond à l'arête $u\check{v}$ de $\check{T}_\alpha^t$, nous avons $P_\alpha(x,y) = \lambda(P_\beta(x,y))$ et $\check{P}_\alpha(x,y) = \lambda(\check{P}_\beta(x,y))$, avec $\lambda(i) = I_\phi(i, \mathcal{P}_{xy}, \bullet)$ et $\mathcal{P}_{xy}$ la bipartition de A_α induite par xy. Grâce à la Propriété 5.75, nous pouvons écrire $\lambda(\check{P}_\beta(x,y)) \preccurlyeq \lambda(P_\beta(x,y))$, ce qui nous donne $\check{P}_\alpha(x,y) \preccurlyeq P_\alpha(x,y)$. Dans le cas où xy identifie l'arête uv de T_α^t ainsi que l'arête $u\check{v}$ de $\check{T}_\alpha^t$, les bipartitions $\mathcal{P}_{uv}$ et $\mathcal{P}_{u\check{v}}$ de A_α sont égales et, par conséquent, $\ell_\alpha(uv) = \ell_\alpha(u\check{v})$.

Lemme 5.81.11. *La Construction 2 garantit que, pour toute arête xy de $T_\alpha^t \cong \check{T}_\alpha^t$, nous avons $\check{P}_\alpha(x,y) \preccurlyeq P_\alpha(x,y)$.*

Si xy n'identifie pas l'arête uw de T_α^t, laquelle correspond à l'arête $\check{u}w$ de $\check{T}_\alpha^t$, nous réutilisons les arguments du Lemme 5.81.10. Dans le cas où xy identifie l'arête uw de T_α^t ainsi que l'arête $\check{u}w$ de $\check{T}_\alpha^t$, les bipartitions $\mathcal{P}_{uw}$ et $\mathcal{P}_{\check{u}w}$ de A_α sont égales aux bipartitions $\mathcal{P}_{uv}$ et $\mathcal{P}_{u\check{v}}$ de A_α et, par conséquent, $\ell_\alpha(uw) = \ell_\alpha(\check{u}w)$.

Lemme 5.81.12. *La Construction 3 garantit que, pour toute arête xy de $T_\alpha^t \cong \check{T}_\alpha^t$, nous avons $\check{P}_\alpha(x,y) \preccurlyeq P_\alpha(x,y)$.*

Si xy n'identifie pas l'arête su ou uw de T_α^t, lesquelles correspondent aux arêtes $s\check{u}$ et $\check{u}w$ de $\check{T}_\alpha^t$, nous réutilisons les arguments du Lemme 5.81.10. Dans le cas où xy identifie les arêtes su de T_α^t et $s\check{u}$ de $\check{T}_\alpha^t$ ou les arêtes uw de T_α^t et $\check{u}w$ de $\check{T}_\alpha^t$, par construction, nous

avons $\check{P}_\beta(s,\check{u}) \preccurlyeq P_\beta(s,u)$ et $\check{P}_\beta(\check{u},w) \preccurlyeq P_\beta(u,w)$. Encore une fois grâce à la Propriété 5.75, nous obtenons $\check{P}_\alpha(x,y) \preccurlyeq P_\alpha(x,y)$.

Lemme 5.81.13. *La Construction 1 garantit que $\tau_\alpha(\check{T}_\alpha) \in FSC_\alpha$.*

L'insertion de $\check{v}$ en temps que fils de u dans $\check{T}_\beta$ revient à ajouter $\check{v}$ en temps que fils de u dans $\check{T}^c_\beta$, ce qui correspond à la Procédure d'Introduction de Type 1. Et, d'après, le Corollaire 5.80, nous avons $\tau_\alpha(\check{T}_\alpha) \in FSC_\alpha$.

Lemme 5.81.14. *La Construction 2 garantit que $\tau_\alpha(\check{T}_\alpha) \in FSC_\alpha$.*

L'insertion de $\check{u}$ en temps que père de w et $\check{v}$ dans $\check{T}_\beta$ revient à ajouter $\check{u}$ en temps que père de w et $\check{v}$ dans $\check{T}^c_\beta$, ce qui correspond à la Procédure d'Introduction de Type 2. Et, d'après, le Corollaire 5.80, nous avons $\tau_\alpha(\check{T}_\alpha) \in FSC_\alpha$.

Lemme 5.81.15. *La Construction 3 garantit que $\tau_\alpha(\check{T}_\alpha) \in FSC_\alpha$.*

Si le sommet $\check{u}$ est présent dans $\check{T}_\beta$, alors nous nous référons au Lemme 5.81.13. Sinon, la subdivision de l'arête $\check{s}\check{w}$, puis l'ajout de $\check{v}$ en temps que fils de $\check{u}$ dans $\check{T}_\beta$ revient à subdiviser une arête $\check{s}\check{w}$ de $\check{T}^c_\beta$ en $\check{s}\check{u}$ et $\check{u}\check{w}$, puis à ajouter $\check{v}$ en temps que fils de $\check{u}$, ce qui correspond à la Procédure d'Introduction de Type 3. Et, d'après, le Lemme 5.79, nous avons $\tau_\alpha(\check{T}_\alpha) \in FSC_\alpha$. □

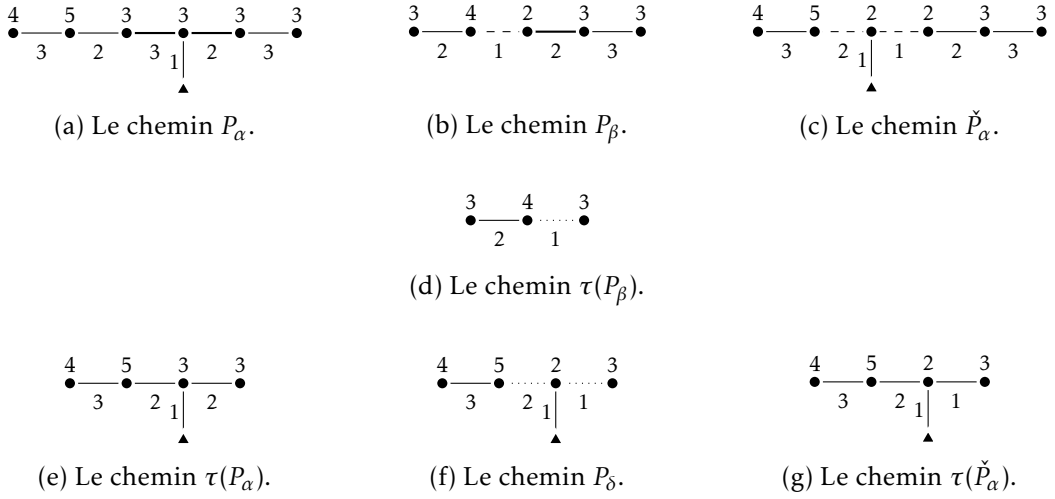
Corollaire 5.82. *Soit α un nœud d'introduction. La Procédure d'Introduction garantit que FSC_α est un ensemble complet de (k,q)-caractéristiques pour A_α restreint à X_α.*

Lemme 5.83. *La Procédure d'Introduction s'exécute en temps constant.*

Démonstration. Soit α un nœud d'introduction et soit β son fils dans $\mathcal{T}$. Par hypothèse, nous savons que FSC_β est un ensemble complet de (k,q)-caractéristiques pour A_β restreint à X_β. Et, par le Lemme 5.61, la taille d'une (k,q)-caractéristique C_β de FSC_β est majorée en fonction de k, q et $|X_\beta|$. Pour chaque (k,q)-caractéristique C_β le nombre d'insertions possibles de $\bullet$ est majoré en fonction de la taille de T^c_β. De plus, les définitions des différentes composantes de C_α se font en parcourant T^c_β. Ici encore la phase de *contraction des tronçons de T^c_α* s'exécute en temps constant. Finalement, l'énumération des (k,q)-caractéristiques de FSC_β est majorée en fonction de k, q et $|X_\beta|$, comme nous l'indique le Lemme 5.62. □

Maintenant que nous nous sommes fait une meilleur idée de comment fonctionne la modification d'une caractéristique, nous pouvons montrer une des raisons pour lesquelles il est nécessaire de recourir à une relation d'ordre sur les abres de partitions.

Exemple 5.84. Afin de rendre cet exemple plus parlant, nous nous appuyons sur la Figure 5.4. Nous reprenons alors un nœud d'introduction α avec son fils β dans $\mathcal{T}$. Ce que nous voulons montrer ici c'est qu'il existe un arbre de partitions T_α de A_α restreint à X_α, pour lesquel les Procédures de Caractérisation et d'Introduction que nous utilisons ne permettent pas de calculer une caractéristique de T_α à partir d'une caractéristique de T_β. À titre de comparaisons, nous présentons également un arbre $\check{T}_\alpha \prec_\alpha T_\alpha$.



(a) Le chemin P_α. (b) Le chemin P_β. (c) Le chemin $\check{P}_\alpha$.

(d) Le chemin $\tau(P_\beta)$.

(e) Le chemin $\tau(P_\alpha)$. (f) Le chemin P_δ. (g) Le chemin $\tau(\check{P}_\alpha)$.

FIGURE 5.4. : Illustration de l'Exemple 5.84 montrant la nécessité de recourir à une relation d'ordre sur les arbres de partitions.

Pour cela, nous nous concentrons sur un tronçon P_β de T_β et sa contraction $\tau(P_\beta)$. Nous insérons alors un nouvel élément dans T_β en utilisant la Procédures d'Introduction de Type 3, c'est-à-dire que nous subdivisons une arête de P_β et que nous ajoutons le nouvel élément en temps que fils du sommet ainsi créé (l'insertion de cet élément se traduit par une augmentation de 1 de la largeur des sommets et arêtes « qui ne sont pas à sa droite » sur le schéma). Nous opérons alors deux subdivisons différentes, et nous obtenons les chemins P_α (subdivison de l'arête en gras de P_β) et $\check{P}_\alpha$ (subdivison de l'arête en tirets). Ces deux subdivisons de P_β, bien qu'elles soient différentes, se traduisent par la subdivison de l'arête en pointillés de $\tau(P_\beta)$, laquelle est illustrée par le chemin P_δ.

Nous notons alors que les étiquettes du chemin P_δ sont les mêmes que celles du chemin $\tau(\check{P}_\alpha)$, alors que ce n'est pas le cas pour les étiquettes de la dernière arête et du dernier sommet interne de $\tau(P_\alpha)$.

5.4.4. Les nœuds de jonction

Définition 5.85 (Caractéristiques compatibles)**.** Soient A_β et A_γ deux ensembles et soit $X = A_\beta \cap A_\gamma \neq \emptyset$. Soit C_β une caractéristique de A_β restreint à X et soit C_γ une caractéristique de A_γ restreint à X. Nous disons alors que C_β et C_γ sont *compatibles* si $a_\beta = 0$ ou $a_\gamma = 0$ et que les troncs de T_β et T_γ restreint à X sont isomorphes.

Définition 5.86 (Fusion de chemins étiquetés)**.** Soient P et Q deux chemins étiquetés et soient $\bar{P} = (\bar{p}_0, \ldots, \bar{p}_n)$ et $\bar{Q} = (\bar{q}_0, \ldots, \bar{q}_n)$ deux extensions de P et Q de même longueur. Soit une fonction $\mu : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ strictement croissante. Nous appelons une *fusion de P et Q suivant μ* le chemin étiqueté $M = (\bar{m}_0, \ldots, \bar{m}_n)$ tel que, pour tout $0 \leq i \leq n$, nous avons $\ell(\bar{m}_i) = \mu(\ell(\bar{p}_i), \ell(\bar{q}_i))$ et, pour tout $0 < i \leq n$, nous avons $\ell(\bar{g}_i) = \mu(\ell(\bar{e}_i), \ell(\bar{f}_i))$, avec $\bar{g}_1, \ldots, \bar{g}_n$ les arêtes de M, $\bar{e}_1, \ldots, \bar{e}_n$ celles de $\bar{P}$ et $\bar{f}_1, \ldots, \bar{f}_n$ celles de $\bar{Q}$.

Procédure de Jonction. Soit α un nœud de jonction et soient β et γ ses deux fils dans $\mathcal{T}$. Nous rapellons que, par définition, nous avons $X_\alpha = X_\beta = X_\gamma$. Afin de calculer l'ensemble complet des (k,q)-caractéristiques de α, nous considérons que nous avons préalablement calculé les ensembles FSC_β et FSC_γ. Puis, pour chaque couple de (k,q)-caractéristiques compatibles (C_β, C_γ) de $FSC_\beta \times FSC_\gamma$, ainsi que pour chaque fusion possible de T^c_β et T^c_γ, nous calculons la caractéristique C_α associée et, si C_α est une (k,q)-caractéristique, alors nous l'ajoutons à FSC_α. De nouveau, C_α est une notation élégante pour $(T^c_\alpha, \ell_\alpha, S_\alpha, a_\alpha, d_\alpha, b_\alpha)$ et il en va de même pour C_β et C_γ.

▹ Définition de T^c_α. Soit T^c_α l'abre obtenu en lissant tous les sommets de degré 2 dans T^c_β. Pour chaque sommet de T^c_α, il existe donc un correspondant dans T^c_β ainsi que dans T^c_γ, étant donné que C_β et C_γ sont compatibles. Par conséquent, pour chaque arête xy de T^c_α, il existe un sous-chemin $P^c_\beta(x,y)$ dans T^c_β et un sous-chemin $P^c_\gamma(x,y)$ dans T^c_γ. Nous remplaçons alors, dans T^c_α, l'arête xy par une fusion $P^c_\alpha(x,y)$ de $P^c_\beta(x,y)$ et $P^c_\gamma(x,y)$ suivant une fonction μ, fonction que nous détaillons dans la phase de *définition de* ℓ_α.

▹ Définition de ℓ_α. Les étiquettes des sommets internes ainsi que des arêtes de T^c_α sont définies lors des différentes fusions. Plus précisement, lors du calcul de $P^c_\alpha(x,y)$, la fonction $\mu : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ prend en paramètres deux entiers i et j et retourne la valeur $J_\phi(i,j,\mathcal{Q}_{xy})$, avec $\mathcal{Q}_{xy}$ la bipartition de X_α induite par l'arête xy avant son remplacement par $P^c_\alpha(x,y)$ dans T^c_α. Notons que les sommets internes ainsi que les arêtes de $P^c_\alpha(x,y)$ induisent une bipartition de X_α égale à $\mathcal{Q}_{xy}$, et il en va de même pour $P^c_\beta(x,y)$ et $P^c_\gamma(x,y)$.

▹ Définition de S_α. Nous retrouvons dans S_α tous les éléments de S_β ainsi que ceux de S_γ, ce qui se traduit par $S_\alpha := S_\beta \cup S_\gamma$.

▹ Contraction des tronçons de T^c_α. Les différentes fusions redéfinissent totalement les tronçons de T^c_α par rapport à ceux que nous avions dans T^c_β et T^c_γ. Il nous faut par conséquent procéder aux contractions de l'ensemble des tronçons de T^c_α.

▹ Définition de a_α. Par hypothèse nous avons choisi C_γ telle que $a_\gamma = 0$. Les seules ramifications ancestrales oubliées proviennent donc uniquement de C_β. Nous définissons donc $a_\alpha := a_\beta$.

▹ Définition de d_α et b_α. Le calcul de d_α suit une procédure montante sur $cp(T^c_\alpha)$, de laquelle nous profitons pour définir b_α. Par conséquent, pour chaque feuille x de $cp(T^c_\alpha)$ nous définissons $d_\alpha(x) := \max\{d_\beta(x), d_\gamma(x)\} + \varepsilon(x)$, avec $\varepsilon(x)$ qui vaut 1 si $b_\beta(x) = b_\gamma(x) = 1$ et 0 sinon. Nous définissons également $b_\alpha(x) := \min\{2, b_\beta(x) + b_\gamma(x)\}$. Puis pour chaque sommet interne x de $cp(T^c_\alpha)$ dont tous les fils ont été traités, nous commençons par récolter la hauteur de ramification maximale parmis les fils de x, que nous notons $\nu(x)$, puis nous définissons $b_\alpha(x) := \max\{b_\beta(x), b_\gamma(x)\}$ et enfin $d_\alpha(x) := \max\{d_\beta(x), d_\gamma(x), \nu(x) + \varepsilon(x)\}$, avec $\varepsilon(x)$ qui vaut 1 si $b_\alpha(x) = 2$ et 0 sinon. ■

Définition 5.87 (Fusion respectueuse)**.** Soient P et Q deux chemins étiquetés et soient $\dot{P}$ et $\dot{Q}$ deux contractions de P et Q. Soit M une fusion de P et Q suivant une fonction μ et soit $\dot{M}$ une fusion de $\dot{P}$ et $\dot{Q}$ suivant μ. Nous disons alors que M *respecte* $\dot{M}$ si les six conditions suivantes sont vérifiées :

- si le sommet $\dot{m}$ de $\dot{M}$ est la fusion des sommets $\dot{p}$ et $\dot{q}$ de $\dot{P}$ et $\dot{Q}$, alors il existe un sommet m de M qui fusionne les origines de $\dot{p}$ et $\dot{q}$ dans P et Q ;

- si le sommet $\dot{m}$ de $\dot{M}$ est la fusion du sommet $\dot{p}$ de $\dot{P}$ et de l'arête $\dot{f}$ de $\dot{Q}$, alors il existe un sommet m de M qui fusionne les origines de $\dot{p}$ et $\dot{e}$ dans P et Q ;
- si le sommet $\dot{m}$ de $\dot{M}$ est la fusion de l'arête $\dot{e}$ de $\dot{P}$ et du sommet $\dot{q}$ de $\dot{Q}$, alors il existe un sommet m de M qui fusionne les origines de $\dot{e}$ et $\dot{q}$ dans P et Q ;
- si l'arête $\dot{g}$ de $\dot{M}$ est la fusion des arêtes $\dot{e}$ et $\dot{f}$ de $\dot{P}$ et $\dot{Q}$, alors il existe une arête g de M qui fusionne les origines de $\dot{e}$ et $\dot{f}$ dans P et Q ;
- si l'arête $\dot{g}$ de $\dot{M}$ est la fusion de l'arête $\dot{e}$ de $\dot{P}$ et du sommet $\dot{q}$ de $\dot{Q}$, alors il existe une arête g de M qui fusionne les origines de $\dot{e}$ et $\dot{q}$ dans P et Q ;
- si l'arête $\dot{g}$ de $\dot{M}$ est la fusion du sommet $\dot{p}$ de $\dot{P}$ et de l'arête $\dot{f}$ de $\dot{Q}$, alors il existe une arête g de M qui fusionne les origines de $\dot{p}$ et $\dot{f}$ dans P et Q.

Le sommet m et l'arête g sont alors appelés les *origines* de $\dot{m}$ et $\dot{g}$ dans M.

Propriété 5.88. *Soient P et Q deux chemins étiquetés, et soient $\dot{P}$ et $\dot{Q}$ des contractions de P et Q respectivement. Alors, pour toute fusion $\dot{M}$ de $\dot{P}$ et $\dot{Q}$ suivant une fonction μ, il existe une fusion M de P et Q suivant μ qui respecte $\dot{M}$.*

Démonstration. Soient $\dot{P}^+ = (\dot{p}_0, \ldots, \dot{p}_{\dot{n}})$ et $\dot{Q}^+ = (\dot{q}_0, \ldots, \dot{q}_{\dot{n}})$ les extensions de $\dot{P}$ et $\dot{Q}$ relatives à la fusion $\dot{M}$. Nous commençons par construire l'extension P^+ de P qui est telle que si l'arête $\dot{e}$ de $\dot{P}$ a été subdivisée i fois pour obtenir $\dot{P}^+$, alors nous subdivision i fois l'origine de $\dot{e}$ dans P. De la même façon nous construisons l'extension Q^+ de Q. Par conséquent, nous pouvons considérer que chaque sommet de $\dot{P}^+$ à un sommet d'origine dans P^+, et que chaque sommet de $\dot{Q}^+$ à un sommet d'origine dans Q^+. Nous avons également que chaque arête de $\dot{P}^+$ à une arête d'origine dans P^+, et que chaque arête de $\dot{Q}^+$ à une arête d'origine dans Q^+.

Lors de la fusion de $\dot{P}$ et $\dot{Q}$, pour tout $0 \leq \dot{r} \leq \dot{n}$, le sommet $\dot{p}_{\dot{r}}$ de $\dot{P}^+$ est fusioné avec le sommet $\dot{q}_{\dot{r}}$ de $\dot{Q}^+$ et, pour tout $0 < \dot{r} \leq \dot{n}$, l'arête $\dot{e}_{\dot{r}}$ de $\dot{P}^+$ est fusionnée avec l'arête $\dot{f}_{\dot{r}}$ de $\dot{Q}^+$. Afin de faire de M une fusion de P et Q qui respecte $\dot{M}$, il nous suffit, pour tout $0 \leq \dot{r} \leq \dot{n}$, de fusioner l'origine de $\dot{p}_{\dot{r}}$ dans P^+ avec l'origine de $\dot{q}_{\dot{r}}$ dans Q^+ et, pour tout $0 < \dot{r} \leq \dot{n}$, de fusioner l'origine de $\dot{e}_{\dot{r}}$ dans P^+ avec l'origine de $\dot{f}_{\dot{r}}$ dans Q^+. Comme P^+ et Q^+ n'ont pas nécessairement la même longueur, nous allons devoir construire les extensions $\bar{P}^+ = (\bar{p}_0, \ldots, \bar{p}_{\bar{n}})$ et $\bar{Q}^+ = (\bar{q}_0, \ldots, \bar{q}_{\bar{n}})$ de P^+ et Q^+ (qui sont également des extensions de P et Q) telles que si $\bar{p}_{\bar{r}}$ représente l'origine de $\dot{p}_{\dot{r}}$ dans $\bar{P}^+$, alors $\bar{q}_{\bar{r}}$ représente l'originie de $\dot{q}_{\dot{r}}$ dans $\bar{Q}^+$ et, si $\bar{e}_{\bar{r}}$ représente l'originie de $\dot{e}_{\dot{r}}$ dans $\bar{P}^+$, alors $\bar{f}_{\bar{r}}$ représente l'origine de $\dot{f}_{\dot{r}}$ dans $\bar{Q}^+$.

La construction de $\bar{P}^+$ et $\bar{Q}^+$ se fait en suivant une méthode récursive. Nous commençons par aligner l'origine $\bar{p}_0$ de $\dot{p}_0$ dans $\bar{P}^+$ avec l'origine $\bar{q}_0$ de $\dot{q}_0$ dans $\bar{Q}^+$, ce qui ne cause pas trop de problème. Nous admettons ensuite que l'origine de $\dot{p}_{\dot{r}}$ dans $\bar{P}^+$ est alignée avec l'origine de $\dot{q}_{\dot{r}}$ dans $\bar{Q}^+$, c'est à dire que $\bar{p}_{\bar{r}}$ correspond à l'origine de $\dot{p}_{\dot{r}}$ dans $\bar{P}^+$ et que $\bar{q}_{\bar{r}}$ correspond à l'origine de $\dot{q}_{\dot{r}}$ dans $\bar{Q}^+$. Il nous faut alors aligner l'origine de $\dot{e}_{\dot{r}+1}$ avec l'origine de $\dot{f}_{\dot{r}+1}$ dans $\bar{P}^+$ et $\bar{Q}^+$ respectivement. Soit $\bar{e}_{\bar{r}+1+\varepsilon_{\bar{e}}}$ l'origine de $\dot{e}_{\dot{r}+1}$ dans $\bar{P}^+$ et soit $\bar{f}_{\bar{r}+1+\varepsilon_{\bar{f}}}$ l'origine de $\dot{f}_{\dot{r}+1}$ dans $\bar{Q}^+$. Si $\varepsilon_{\bar{e}} = \varepsilon_{\bar{f}}$, alors les arêtes $\bar{e}_{\bar{r}+1+\varepsilon_{\bar{e}}}$ et $\bar{f}_{\bar{r}+1+\varepsilon_{\bar{f}}}$ sont alignées. Sinon, si $\varepsilon_{\bar{e}} < \varepsilon_{\bar{f}}$, alors nous effectuons $\varepsilon_{\bar{f}} - \varepsilon_{\bar{e}}$ subdivisions de $\bar{e}_{\bar{r}+1+\varepsilon_{\bar{e}}}$, afin de l'aligner avec $\bar{f}_{\bar{r}+1+\varepsilon_{\bar{f}}}$. *A contrario*, si $\varepsilon_{\bar{e}} > \varepsilon_{\bar{f}}$, alors c'est $\bar{f}_{\bar{r}+1+\varepsilon_{\bar{f}}}$ que nous subdivisons $\varepsilon_{\bar{e}} - \varepsilon_{\bar{f}}$ fois, afin qu'elle soit alignée avec $\bar{e}_{\bar{r}+1+\varepsilon_{\bar{e}}}$. Nous pouvons maintenant admettre que

l'origine $\bar{e}_{\bar{s}}$ de $\dot{e}_{\dot{s}}$ dans $\bar{P}^+$ est alignée avec l'origine $\bar{f}_{\bar{s}}$ de $\dot{f}_{\dot{s}}$ dans $\bar{Q}^+$, et ainsi nous ateler à l'alignement des origines de $\dot{p}_{\dot{s}}$ et $\dot{q}_{\dot{s}}$ dans $\bar{P}^+$ et $\bar{Q}^+$ respectivement. Soit $\bar{p}_{\bar{s}+\varepsilon_{\bar{p}}}$ l'origine de $\dot{p}_{\dot{s}}$ dans $\bar{P}^+$ et soit $\bar{q}_{\bar{s}+\varepsilon_{\bar{q}}}$ l'origine de $\dot{q}_{\dot{s}}$ dans $\bar{Q}^+$. Si $\varepsilon_{\bar{p}} = \varepsilon_{\bar{q}}$, alors les sommets $\bar{p}_{\bar{s}+\varepsilon_{\bar{p}}}$ et $\bar{q}_{\bar{s}+\varepsilon_{\bar{q}}}$ sont alignés. Sinon, si $\varepsilon_{\bar{p}} < \varepsilon_{\bar{q}}$, alors nous effectuons $\varepsilon_{\bar{q}} - \varepsilon_{\bar{p}}$ subdivisions de $\bar{e}_{\bar{s}}$, afin d'aligner $\bar{p}_{\bar{s}+\varepsilon_{\bar{p}}}$ avec $\bar{q}_{\bar{s}+\varepsilon_{\bar{q}}}$. Dans le cas contraire, c'est-à-dire si $\varepsilon_{\bar{p}} > \varepsilon_{\bar{q}}$, alors c'est $\bar{q}_{\bar{s}}$ que nous subdivisons $\varepsilon_{\bar{p}} - \varepsilon_{\bar{q}}$ fois, afin d'aligner $\bar{q}_{\bar{s}+\varepsilon_{\bar{q}}}$ avec $\bar{p}_{\bar{s}+\varepsilon_{\bar{p}}}$. □

Exemple 5.89. Dans cet exemple nous détaillons le passage d'une fusion $\dot{M}$ de deux contractions $\dot{P}$ et $\dot{Q}$, relatives aux cehmins P et Q, à une fusion M de P et Q qui respecte $\dot{M}$. Pour cela nous nous ramenons à la FIGURE 5.5, dont la légende est présentée dans ce qui suit.

Pour commencer, la sous-figure (a) indique la correspondance entre P et la contraction $\dot{P}$. Les sommets en formes d'étoiles représentent les sommets de P présents dans $\dot{P}$, de même, les arêtes en gras dans P rerésentent les arêtes de P présentes dans $\dot{P}$. Finalement, les fonds gris indiquent la correspondance entre les arêtes de $\dot{P}$ et les sous-chemins de P. La sous-figure (b) indique la correspondance entre Q et la contraction $\dot{Q}$, en suivant la même logique.

Les sous-figures (c) et (d) mettent en exergue la correspondance entre, d'une part, les extensions de $\dot{P}$ et $\dot{Q}$ relatives à $\dot{M}$ et leur conséquences respectives au niveau de P et Q, d'autre part. Nous voyons alors apparaître des « petits » sommets qui symbolisent les sommets issus des différentes subdivisions d'arêtes nécessaires aux extensions ; des lignes pointillés dénotent la correspondance entre les extensions de P et $\dot{P}$ ainsi qu'entre celles de Q et $\dot{Q}$.

Nous passons enfin à la partie la plus importante illustrant le passage de $\dot{M}$ à M. Nous partons alors de $\dot{M}$ puis, en remontant, nous croisons les extensions $\dot{P}^+$ et $\dot{Q}^+$ de $\dot{P}$ et $\dot{Q}$ relatives. Viennent ensuite les extensions P^+ et Q^+ de P et Q « respectant » $\dot{P}^+$ et $\dot{Q}^+$, lesquelles se transforment en $\bar{P}^+$ et $\bar{Q}^+$ afin « d'aligner » P^+ et Q^+. Finalement, nous trouvons la fusion M, relative à P^+ et Q^+, et qui respecte $\dot{M}$. Dans cette dernière sous-figure, les flèches indiquent si un sommet de M (respectivement de $\dot{M}$) est originaire de P ou Q (respectivement de $\dot{P}$ ou $\dot{Q}$).

Propriété 5.90. *Soient P et Q deux chemins étiquetés, et soient $\dot{P}$ et $\dot{Q}$ des contractions de P et Q respectivement. Alors, pour toute fusion M de P et Q suivant une fonction μ, il existe une fusion $\dot{M}$ de $\dot{P}$ et $\dot{Q}$ suivant μ telle que M respecte $\dot{M}$.*

Démonstration. Nous commençons par créer l'extension $\dot{P}^+$ de $\dot{P}$ de façon à ce que $\dot{P}^+$ et P aient la même longueur. Pour cela, pour toute arête xy de $\dot{P}$ nous effectuons n subdivisions de xy, où n est la longueur de $P(x,y)$ moins 1, avec $P(x,y)$ le chemin reliant les origines de x et y dans P. De même, nous créons l'extension $\dot{Q}^+$ de $\dot{Q}$ de façon à ce que $\dot{Q}^+$ et Q aient la même longueur.

Nous considérons ensuite les extensions $P^\times$ et $Q^\times$ de P et Q relatives à M. Nous construisons alors l'extension $\dot{P}^\times$ de $\dot{P}^+$ en subdivisant chaque arête de $\dot{P}^+$ autant de fois que nous avons subdiviser son origine dans P en vue d'obtenir $P^\times$. De même, nous construisons l'extension $\dot{Q}^\times$ de $\dot{Q}^+$ en subdivisant chaque arête de $\dot{Q}^+$ autant de fois que

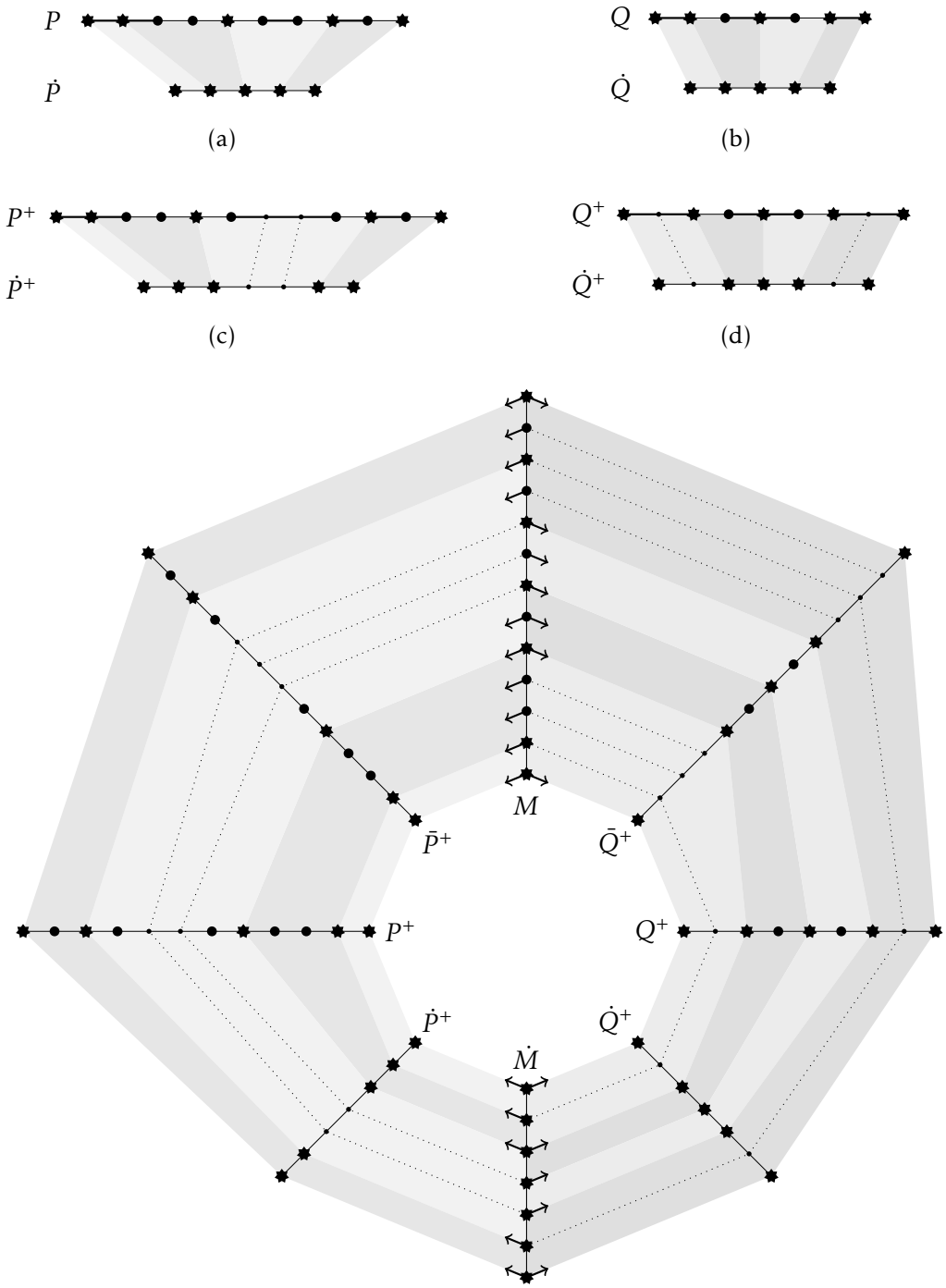


FIGURE 5.5. : Exemple d'une fusion M de deux chemins P et Q qui respecte une fusion $\dot{M}$ de deux compressions $\dot{P}$ et $\dot{Q}$ de P et Q (la description de ce schéma est donnée dans l'Exemple 5.89).

nous avons subdiviser son origine dans Q en vue d'obtenir $Q^\times$. Les chemins $\dot{P}^\times$ et $\dot{Q}^\times$ induisent donc la fusion $\dot{M}$ qui est telle que M respecte $\dot{M}$. □

Propriété 5.91. *Soient P et Q deux chemins étiquetés, et soient $\dot{P}$ et $\dot{Q}$ des contractions de P et Q respectivement. Soit $\dot{M} = (\dot{m}_0, \ldots, \dot{m}_{\dot{n}})$ une fusion de $\dot{P}$ et $\dot{Q}$ suivant une fonction μ, et soit $M = (m_0, \ldots, m_n)$ une fusion de P et Q suivant μ qui respecte $\dot{M}$. Alors, il existe une contraction entre un sommet $\dot{m}_{\dot{r}}$ et une arête $\dot{g}_{\dot{s}}$ de $\dot{M}$ si et seulement si il existe une contraction entre les origines respectives de $\dot{m}_{\dot{r}}$ et $\dot{g}_{\dot{s}}$ dans M.*

Démonstration. Soient $\dot{P}^+ = (\dot{p}_0, \ldots, \dot{p}_{\dot{n}})$ et $\dot{Q}^+ = (\dot{q}_0, \ldots, \dot{q}_{\dot{n}})$ les extensions de $\dot{P}$ et $\dot{Q}$ relatives à $\dot{M}$ et soient $P^+ = (p_0, \ldots, p_n)$ et $Q^+ = (q_0, \ldots, q_n)$ les extensions de P et Q relatives à M. Sans perte de généralité nous considérons que $\dot{r} < \dot{s}$. L'existance d'une contraction entre $\dot{m}_{\dot{r}}$ et $\dot{g}_{\dot{s}}$ se traduit alors par :

$$\begin{aligned} \ell(\dot{m}_{\dot{r}}) \leq \ell(\dot{m}_{\dot{t}}) \leq \ell(\dot{g}_{\dot{s}}), \quad \forall \dot{r} \leq \dot{t} < \dot{s}, \\ \ell(\dot{m}_{\dot{r}}) \leq \ell(\dot{g}_{\dot{t}}) \leq \ell(\dot{g}_{\dot{s}}), \quad \forall \dot{r} < \dot{t} \leq \dot{s}. \end{aligned}$$

Ce qui, par définition, est équivalent à :

$$\begin{aligned} \mu(\ell(\dot{p}_{\dot{r}}), \ell(\dot{q}_{\dot{r}})) \leq \mu(\ell(\dot{p}_{\dot{t}}), \ell(\dot{q}_{\dot{t}})) \leq \mu(\ell(\dot{e}_{\dot{s}}), \ell(\dot{f}_{\dot{s}})), \quad \forall \dot{r} \leq \dot{t} < \dot{s}, \\ \mu(\ell(\dot{p}_{\dot{r}}), \ell(\dot{q}_{\dot{r}})) \leq \mu(\ell(\dot{e}_{\dot{t}}), \ell(\dot{f}_{\dot{t}})) \leq \mu(\ell(\dot{e}_{\dot{s}}), \ell(\dot{f}_{\dot{s}})), \quad \forall \dot{r} < \dot{t} \leq \dot{s}. \end{aligned}$$

Et, comme μ est strictement croissante, nous obtenons :

$$\begin{aligned} \ell(\dot{p}_{\dot{r}}) \leq \ell(\dot{p}_{\dot{t}}) \leq \ell(\dot{e}_{\dot{s}}) \quad \text{et} \quad \ell(\dot{q}_{\dot{r}}) \leq \ell(\dot{q}_{\dot{t}}) \leq \ell(\dot{f}_{\dot{s}}), \quad \forall \dot{r} \leq \dot{t} < \dot{s}, \\ \ell(\dot{p}_{\dot{r}}) \leq \ell(\dot{e}_{\dot{t}}) \leq \ell(\dot{e}_{\dot{s}}) \quad \text{et} \quad \ell(\dot{q}_{\dot{r}}) \leq \ell(\dot{f}_{\dot{t}}) \leq \ell(\dot{f}_{\dot{s}}), \quad \forall \dot{r} < \dot{t} \leq \dot{s}. \end{aligned}$$

Comme M respecte $\dot{M}$, il est possible de définir le sommet p_r, à l'origine de $\dot{p}_{\dot{r}}$ dans P^+ ainsi que l'arête e_s, à l'origine de $\dot{e}_{\dot{s}}$ dans P^+. De même, nous définissons le sommet q_r comme étant l'origine de $\dot{q}_{\dot{r}}$ dans Q^+ ainsi que l'arête f_s comme étant l'origine de $\dot{f}_{\dot{s}}$ dans Q^+. En appliquant la Propriété 5.54 nous obtenons :

$$\begin{aligned} \ell(p_r) \leq \ell(p_t) \leq \ell(e_s) \quad \text{et} \quad \ell(q_r) \leq \ell(q_t) \leq \ell(f_s), \quad \forall r \leq t < s, \\ \ell(p_r) \leq \ell(e_t) \leq \ell(e_s) \quad \text{et} \quad \ell(q_r) \leq \ell(f_t) \leq \ell(f_s), \quad \forall r < t \leq s. \end{aligned}$$

Une fois de plus, comme μ est strictement croissante, nous avons :

$$\begin{aligned} \mu(\ell(p_r), \ell(q_r)) \leq \mu(\ell(p_t), \ell(q_t)) \leq \mu(\ell(e_s), \ell(f_s)), \quad \forall r \leq t < s, \\ \mu(\ell(p_r), \ell(q_r)) \leq \mu(\ell(e_t), \ell(f_t)) \leq \mu(\ell(e_s), \ell(f_s)), \quad \forall r < t \leq s. \end{aligned}$$

Finalement, par définition, nous obtenons :

$$\begin{aligned} \ell(m_r) \leq \ell(m_t) \leq \ell(g_s), \quad \forall r \leq t < s, \\ \ell(m_r) \leq \ell(g_t) \leq \ell(g_s), \quad \forall r < t \leq s, \end{aligned}$$

avec m_r l'origine de $\dot{m}_{\dot{r}}$ dans M et g_s l'origine de $\dot{g}_{\dot{s}}$ dans M.

La preuve de la réciproque se fait en suivant le même raisonnement. □

Remarque 5.92. Si m' est un sommet de M qui n'est pas présent dans $\dot{M}$, alors il existe un sommet m de M, à l'orgine du sommet $\dot{m}$ de $\dot{M}$, et une arête g de M, à l'orgine de l'arête $\dot{g}$ de $\dot{M}$, tels que $\dot{g}$ est incidente à $\dot{m}$ et que m' se situe entre m et g dans M. Nous pouvons donc considérer qu'il existe une contraction entre $\dot{m}$ et $\dot{g}$ dans $\dot{M}$ et, par la Propriété 5.90, nous savons qu'il existe une contraction entre m et g dans M. En d'autres termes, si m' est un sommet de M qui n'est pas présent dans $\dot{M}$, alors il existe une contraction qui supprime m' de M. De même, si g' est une arête de M qui n'est pas présente dans $\dot{M}$, alors il existe une contraction qui supprime g' de M.

Propriété 5.93. *Soient P, Q, $\check{P}$ et $\check{Q}$ quatre chemins étiquetés tels que $\check{P} \preccurlyeq P$ et $\check{Q} \preccurlyeq Q$. Soit M une fusion de P et Q suivant une fonction μ. Il existe alors une fusion $\check{M}$ de $\check{P}$ et $\check{Q}$ suivant μ telle que $\check{M} \preccurlyeq M$.*

Démonstration. Soient $\bar{P}$ et $\bar{Q}$ les extensions de P et Q relatives à M. Soient P^+ et $\check{P}^+$ les extensions de P et $\check{P}$ telles que $\check{P}^+ \leq P^+$. Soient $Q^\times$ et $\check{Q}^\times$ les extensions de Q et $\check{Q}$ telles que $\check{Q}^\times \leq Q^\times$. La preuve de cette propriété consiste à étendre les chemins $\bar{P}$, $\bar{Q}$, P^+, $\check{P}^+$, $Q^\times$ et $\check{Q}^\times$ de façons à ce que, au final, ils aient tous la même longueur et que les correspondances de sommets et d'arêtes entre $\bar{P}$ et $\bar{Q}$, P^+ et $\check{P}^+$ ainsi qu'entre $Q^\times$ et $\check{Q}^\times$ soient respectées. De cette façons, les extensions de $\check{P}^+$ et $\check{Q}^\times$ seront relatives à une fusion $\check{M}$ de $\check{P}$ et $\check{Q}$, respectant M, et telle que $\check{M} \preccurlyeq M$. Les nombreuses extensions que nous utilisons, ainsi que les relations qu'elles entretiennent, sont résumées en Figure 5.6.

La première étape de la construction de $\check{M}$ consiste à aligner les chemins $\bar{P}$ et P^+, le chemin ainsi créé est noté $\bar{P}^+$. En même temps, nous répercutons les changements apportés à $\bar{P}$ sur $\bar{Q}$, créant alors le chemin $\bar{Q}^+$. Nous répercutons également les changements apportés à P^+ sur $\check{P}^+$ afin d'obtenir le chemin $\check{P}^+$. Cet alignement se traduit par les extensions suivantes. Soit e_i une arête de P et soient ε^+ et $\bar{\varepsilon}$ le nombre d'extensions de e_i dans P^+ et $\bar{P}$. Si $\varepsilon^+ = \bar{\varepsilon}$, alors nous ne changeons rien. Dans le cas où $\varepsilon^+ < \bar{\varepsilon}$, alors nous choisissons e_j^+, une arête de P^+ ayant e_i pour origine dans P, et nous la subdivisons $\bar{\varepsilon} - \varepsilon^+$ fois. Cette extension de P^+ se répercute sur $\check{P}^+$ en subdivisant l'arête $\check{e}_j^+$ autant de fois que e_j^+. Maintenant, si $\bar{\varepsilon} < \varepsilon^+$, alors nous choisissons $\bar{e}_j$, une arête de $\bar{P}$ ayant e_i pour origine dans P, et nous la subdivisons $\varepsilon^+ - \bar{\varepsilon}$ fois. La répercussion de ces subdivisons se fait cette fois-ci sur l'arête $\bar{f}_j$ de $\bar{Q}$.

Les différentes extensions que nous venons de présenter sont telles que $\check{P}^+ \leq \bar{P}^+$ et que la fusion M^+ relative à $\bar{P}^+$ et $\bar{Q}^+$ est une extension de M. Ceci implique notamment que $\check{P}^+$, $\bar{P}^+$ et $\bar{Q}^+$ ont la même longeur.

Nous passons alors à la deuxième étape de la construction de $\check{M}$, laquelle consiste à aligner les chemins $\bar{Q}^+$ et $Q^\times$, le chemin ainsi créé est noté $\bar{Q}^\times$. Les changements apportés à $\bar{Q}^+$ sont alors répercutés sur $\bar{P}^+$ et $\check{P}^+$, donnant naissance aux chemins $\bar{P}^\times$ et $\check{P}^\times$. De plus, les changements apportés à $Q^\times$ sont répercutés sur $\check{Q}^\times$, afin d'obtenir le chemin $\check{Q}^\times$. Nous réutilisons la méthode d'alignement précédente.

Il en découle alors que $\bar{P}^\times$, $\bar{Q}^\times$, $\check{P}^\times$ et $\check{Q}^\times$ ont la même longueur. Soit $M^\times$ la fusion relative à $\bar{P}^\times$ et $\bar{Q}^\times$ et soit $\check{M}$ la fusion relative à $\check{P}^\times$ et $\check{Q}^\times$. Comme $\check{P}^\times \leq \bar{P}^\times$ et $\check{Q}^\times \leq \bar{Q}^\times$ et que μ est strictement croissante, nous avons que $\check{M} \leq M^\times$ et donc que $\check{M} \preccurlyeq M$, car $M^\times$ est une extension de M. □

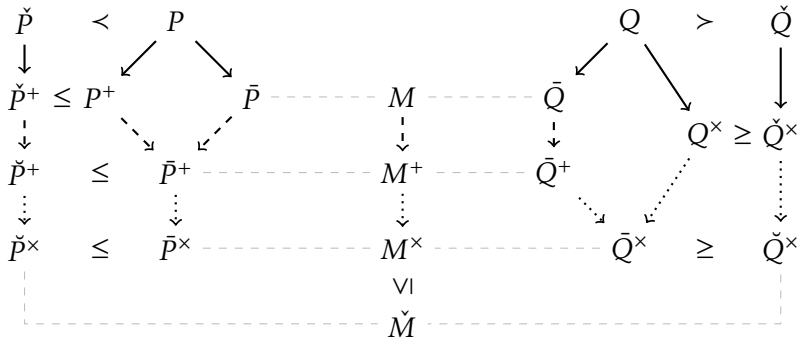


FIGURE 5.6. : Extensions utilisées lors de la construction d'une fusion $\check{M}$ de $\check{P}$ et $\check{Q}$ respectant une fusion M de P et Q. Les flèches pleines indiquent les extensions « usuelles ». Les flèches en tirets représentent l'alignement de P^+ et $\bar{P}$. Les flèches en pointillés représentent l'alignement de $Q^\times$ et $\bar{Q}^+$.

Lemme 5.94. *Soit α un nœud de jonction. La Procédure de Jonction garantit que, pour chaque (k,q)-caractéristique C_δ de FSC_α, il existe un (k,q)-arbre de partitions T_α de A_α tel que $C_\delta = \tau_\alpha(T_\alpha)$.*

Démonstration. Soient β et γ les deux fils de α dans $\mathcal{T}$ et soient C_β et C_γ les (k,q)-caractéristiques respectives de A_β restreint à X_β et A_γ restreint à X_γ à partir desquelles nous avons calculé C_δ. Par hypothèse, FSC_β est un ensemble complet de (k,q)-caractéristiques pour A_β restreint à X_β, il existe donc un (k,q)-arbre de partitions T_β de A_β tel que $C_\beta = \tau_\beta(T_\beta)$. De même, FSC_γ est un ensemble complet de (k,q)-caractéristiques pour A_γ restreint à X_γ, il existe donc un (k,q)-arbre de partitions T_γ de A_γ tel que $C_\gamma = \tau_\gamma(T_\gamma)$. Nous construisons alors un arbre T_α en appliquant la *Fusion de T_β et T_γ* présentée ci-desssous. Ce que nous cherchons à démontrer c'est que $C_\delta = \tau_\alpha(T_\alpha)$. Pour cela, nous considérons $C_\alpha = \tau_\alpha(T_\alpha)$ et nous montrons que $C_\delta = C_\alpha$.

Comme à l'accoutumée, T_α^r, T_β^r et T_γ^r représentent les restrictions respectives de T_α, T_β et T_γ à $X_\alpha = X_\beta = X_\gamma$. De plus, la notation C_α est utilisée pour représenter le 6-uplet $(T_\alpha^c, \ell_\alpha, S_\alpha, a_\alpha, d_\alpha, b_\alpha)$, et il en va de même pour C_β, C_γ et C_δ.

Fusion de T_β et T_γ. Soit T^s l'abre obtenu en lissant tous les sommets de degré 2 dans T_β^c. Comme C_β et C_γ sont compatibles, T^s est également le résultat du lissage des sommets de degré 2 dans T_γ^c. De plus, chaque sommet de T^s a un représentant dans T_β ainsi que dans T_γ. La première étape de construction de T_α consiste à fusionner chaque représentant d'un sommet de T^s dans T_β avec son correspondant dans T_γ. À ce moment précis, T_α n'est pas un arbre à proprement dit, en effet, pour chaque arête xy de T^s, il existe un sous-chemin $P_\beta(x,y)$ de T_α issu de T_β ainsi qu'un sous-chemin $P_\gamma(x,y)$ de T_α issu de T_γ. La seconde étape consiste alors à fusionner $P_\beta(x,y)$ avec $P_\gamma(x,y)$. Pour cela, nous rappelons que le sous-chemin $P_\delta^c(x,y)$ de T_δ^c a été obtenu par une fusion M^c des sous-chemins $P_\beta^c(x,y)$ et $P_\gamma^c(x,y)$ de T_β^c et T_γ^c respectivement. Et donc, en utilisant la Propriété 5.88, nous savons comment construire une fusion de $P_\beta(x,y)$ et $P_\gamma(x,y)$ respectant M^c. ■

Lemme 5.94.1. *Les ensembles S_α et S_δ sont égaux.*

Si x est un sommet de ramifications dans T_β, alors il le reste dans T_α. De même, si x est le père d'une feuille dans T_β^r, alors il est toujours le père de cette même feuille dans T_α^r. Par conséquent, nous avons $S_\beta \subseteq S_\alpha$ et, avec le même raisonnement, nous obtenons $S_\gamma \subseteq S_\alpha$. En combinant ces deux résultats nous obtenons $S_\beta \cup S_\gamma \subseteq S_\alpha$.

Afin de montrer que $S_\alpha = S_\delta$ il nous faut prouver que $S_\alpha \subseteq S_\beta \cup S_\gamma$. Pour cela, nous remarquons que toutes les feuilles de T_α^r apparaissent dans S_β ou dans S_γ, par conséquent, tout sommet x qui est le père d'une feuille dans T_α^r est le père d'une feuille dans T_β^r ou dans T_γ^r. En ce qui concerne les sommets de ramifications de T_α^r il existe plusieurs cas de figure. Soit x un sommet de ramifications de T_α^r et soient y et z les sommets issus [4] de T_β^r et T_γ^r respectivement qui ont été fusionné afin d'obtenir x. Si y à au moins deux ramifications dans T_β^r, alors y appartient à S_β. De même, si z à au moins deux ramifications dans T_γ^r, alors z appartient à S_γ. Il nous reste donc les quatre cas où y et z ont au plus une ramification chacun. Dans les trois cas où l'un de ces deux sommets au moins n'admet aucune ramification, alors le sommet x à au plus une ramification et il n'appartient alors pas à S_α (à moins qu'il ne soit le pére d'une feuille bien évidemment). Le quatrième cas est celui où y et z admettent tous les deux exactement une ramification. Si y et z ne sont pas des feuilles de $cp(T_\beta^r)$ et $cp(T_\gamma^r)$, alors leurs fils respectifs ont été fusionnés afin de donner le fils de x dans T_α^r, et x appartient à S_α si et seulement si il est le père d'une feuille. Par contre, si y et z sont des feuilles de $cp(T_\beta^r)$ et $cp(T_\gamma^r)$, alors leurs fils respectifs qui ne sont pas présents dans T_β^r et T_γ^r ne sont alors pas fusionnés, x admet donc deux ramifications. Le sommet x est alors un sommet de ramifications alors que ni y ni z n'en sont un, cependant ces deux derniers sommets sont, par la Procédure de Caractérisation, les pères de feuilles dans T_β^r et T_γ^r. Ils appartiennent donc à S_β et S_γ.

En conclusion, nous avons $S_\alpha = S_\beta \cup S_\gamma$ ce qui correspond à la valeur affectée à S_δ lors de la Procédure de Jonction.

Lemme 5.94.2. *Les arbres T_α^c et T_δ^c sont isomorphes et les fonctions ℓ_α et ℓ_δ sont égales.*

Nous reprenons ici l'arbre T^s introduit lors de la Fusion de T_β et T_γ. Par construction, T^s est l'arbre que nous obtenons en lissant les sommets de degré 2 dans T_α^r et T_δ^c. Soit xy une arête de T^s. Nous définissons alors $P_\beta^r(x,y)$ et $P_\gamma^r(x,y)$ les sous-chemins de T_β^r et T_γ^r reliant x à y, ainsi que M_α la fusion de $P_\beta^r(x,y)$ et $P_\gamma^r(x,y)$ calculée lors de la Fusion de T_β et T_γ. En parallèle, nous définissons $P_\beta^c(x,y)$ et $P_\gamma^c(x,y)$ les sous-chemins de T_β^c et T_γ^c reliant x à y, ainsi que M_δ la fusion de $P_\beta^c(x,y)$ et $P_\gamma^c(x,y)$ calculée par la Procédure de Jonction. Par construction, nous savons que M_α respecte M_δ. En combinant ce résultat avec le fait que $S_\alpha = S_\delta$, nous pouvons écrire $M_\alpha = M_\alpha^1 \odot \cdots \odot M_\alpha^n$ et $M_\delta = M_\delta^1 \odot \cdots \odot M_\delta^n$, où chaque M_α^i représente un tronçon de T_α^c avant contraction et, chaque M_δ^i représente le tronçon correspondant dans T_δ^c, avant contraction également. En rappelant la Propriété 5.91, nous obtenons $\tau(M_\alpha^i) = \tau(M_\delta^i)$, pour tout $1 \leq i \leq n$.

4. Les sommets y et z peuvent être le résultats de subdivisons d'arêtes dans T_β^r et T_γ^r, auquel cas leur nombre de ramifications est égale à 1.

Nous venons donc de montrer que chaque tronçon P_α^c de T_α^c correspond à un tronçon P_δ^c de T_δ^c. De plus, P_α^c et P_δ^c sont composés de la même séquence de sommets, et les étiquettes attribuées à chaque sommet x et chaque arête e de P_α^c sont respectivement égales aux étiquettes attribuées aux correspondants de x et e dans P_δ^c. Par conséquent, les arbres T_α^c et T_δ^c sont isomorphes et les fonctions ℓ_α et ℓ_δ sont égales.

Lemme 5.94.3. *Les entiers a_α et a_δ sont égaux.*

Comme C_β et C_γ sont compatibles, nous avons que $a_\beta = 0$ ou $a_\gamma = 0$. Sans perte de généralité, nous supposons que $a_\gamma = 0$, en d'autres termes la racine de T_γ est présente dans T_γ^c. Les seuls sommets de ramifications ancestraux oubliés dans T_α^c proviennent donc de T_β. Nous avons alors $\mathrm{dr}_{T_\alpha}(r_\alpha, r_\alpha^r) = \mathrm{dr}_{T_\beta}(r_\beta, r_\beta^r)$, ce qui se traduit par $a_\alpha = a_\beta$. Finalement, par la Procédure de Jonction, nous avons $a_\delta = a_\beta$, et donc $a_\alpha = a_\delta$.

Lemme 5.94.4. *les fonctions b_α et b_δ sont égales.*

En comparant l'étude menée dans le Lemme 5.94.1 avec les calculs présentés dans la Procédure de Jonction, nous pouvons conclure que $b_\alpha(x) = b_\delta(x)$, pour tout sommet interne x de $T_\alpha^c \cong T_\delta^c$.

Lemme 5.94.5. *les fonctions d_α et d_δ sont égales.*

Ici nous considérons $\bar{T}_\beta$ et $\bar{T}_\gamma$, qui sont les images des arbres T_β et T_γ auxquels nous avons appliquer les extensions qui nous ont permi leur fusion. De cette manière, si nous considérons x, un sommet de T_α, nous avons que x est également un sommet de $\bar{T}_\beta$ ainsi que de $\bar{T}_\gamma$. Notons que si x est un sommet issu d'une extension dans $\bar{T}_\beta$ (c'est-à-dire que x n'apparait pas dans T_β), alors la hauteur de ramifications de x dans $\bar{T}_\beta$ est égale à celle du plus proche descendant de x qui n'est pas issu d'une extension ; le même raisonnement s'applique si x est un sommet issu d'une extension dans $\bar{T}_\gamma$.

Une autre remarque avant de rentrer dans le vif du sujet est que, si x n'est pas un sommet de ramifications dans $\bar{T}_\beta$, alors la hauteur de ramifications de x dans $\bar{T}_\beta$ ne tient pas compte du fait que x puisse être un sommet de ramifications dans T_α (ce qui est le cas si, par exemple, x est un sommet de ramifications dans $\bar{T}_\gamma$). Nous définissons alors $\varepsilon_\beta(x)$ qui vaut 1 si x n'est pas un sommet de ramifications dans $\bar{T}_\beta$ et qu'il en est un dans T_α et 0 sinon (x n'est pas un sommet de ramifications de T_α ou il en est déjà un dans $\bar{T}_\beta$). Nous définissons $\varepsilon_\gamma(x)$ de façon similaire.

Soit x un sommet de $cp(T_\alpha^r)$ et soit y un fils de x dans $cp(T_\alpha)$. Si y est un sommet de $V(\bar{T}_\beta) \setminus V(\bar{T}_\gamma)$, alors nous disons que y est un β-fils de x. Sinon si y est un sommet de $V(\bar{T}_\gamma) \setminus V(\bar{T}_\beta)$, alors nous disons que y est un γ-fils de x. Enfin, si y est un sommet de $V(\bar{T}_\beta) \cap V(\bar{T}_\gamma)$, alors nous disons que y est un (β, γ)-fils de x.

Dans le cas où les tous les fils de x dans $cp(T_\alpha)$ sont soit des β-fils soit des γ-fils, il est assez évident de voir que $\mathrm{hr}_{T_\alpha}(x) = \max\{\mathrm{hr}_{\bar{T}_\beta}(x) + \varepsilon_\beta(x), \mathrm{hr}_{\bar{T}_\gamma}(x) + \varepsilon_\gamma(x)\}$, ce qui, en terme de caractéristique, se traduit par $d_\alpha(x) = \max\{d_\beta(x) + \varepsilon_\beta(x), d_\gamma(x) + \varepsilon_\gamma(x)\}$. Dire que x n'admet pas de (β, γ)-fils est équivalent à dire que x est une feuille de $cp(T_\alpha^r)$.

Par contre, si x est un sommet interne de $cp(T_\alpha^r)$, le fait que, lors de la fusion, nous ayons pu, par exemple, placer un sommet de ramifications provenant de T_β entre deux sommets de ramifications provenant de T_γ ne nous permet pas d'exprimer $\mathrm{hr}_{T_\alpha}(x)$ en

fonction de $\mathrm{hr}_{\bar{T}_\beta}(x)+\varepsilon_\beta(x)$ et $\mathrm{hr}_{\bar{T}_\gamma}(x)+\varepsilon_\gamma(x)$. Cependant, nous pouvons calculer la hauteur de ramifications de x en fonction de celles de ses fils, en admettant, bien enendu, que nous ayons préalablement calculer la hauteur de ramifications de ces derniers. Comme nous avons déjà traiter toutes les feuilles de $cp(T_\alpha^r)$, nous pouvons utiliser un calcul montant sur $cp(T_\alpha^r)$ et ainsi obtenir :

$$\mathrm{hr}_{T_\alpha}(x) = \max_{y\in N^+(x)} \left\{ \begin{array}{ll} \mathrm{hr}_{\bar{T}_\beta}(y) & \text{si } y \in N_\beta^+(x) \\ \mathrm{hr}_{\bar{T}_\gamma}(y) & \text{si } y \in N_\gamma^+(x) \\ \mathrm{hr}_{T_\alpha}(y) & \text{si } y \in N_{\beta,\gamma}^+(x) \end{array} \right\} + \varepsilon(x),$$

avec $\varepsilon(x)$ qui vaut 1 si x est un sommet de ramifications de T_α et 0 sinon. Notons que $\max_{y\in N_\beta^+(x)}\{\mathrm{hr}_{\bar{T}_\beta}(y)\} + \varepsilon(x)$ à la même valeur que $\mathrm{hr}_{\bar{T}_\beta}(x) + \varepsilon_\beta(x) = d_\beta(x) + \varepsilon_\beta(x)$ et que $\max_{y\in N_\gamma^+(x)}\{\mathrm{hr}_{\bar{T}_\gamma}(y)\} + \varepsilon(x)$ à la même valeur que $\mathrm{hr}_{\bar{T}_\gamma}(x) + \varepsilon_\gamma(x) = d_\gamma(x) + \varepsilon_\gamma(x)$. Et, en notant $\nu(x) = \max_{y\in N_{\beta,\gamma}^+(x)}\{\mathrm{hr}_{T_\alpha}(y)\}$, nous obtenons :

$$\mathrm{hr}_{T_\alpha}(x) = \max\{d_\beta(x) + \varepsilon_\beta(x), d_\gamma(x) + \varepsilon_\gamma(x), \nu(x) + \varepsilon(x)\}.$$

Finalement, en observant la Procédure de Jonction, nous arrivons à la conclusion que les fonctions d_α et d_δ sont égales.

Lemme 5.94.6. *La largeur q-ramifiée suivant ϕ de T_α est au plus k.*

La hauteur de ramification de r_α peut être définie par $\mathrm{hr}_{T_\alpha}(r_\alpha) = \mathrm{dr}_{T_\alpha}(r_\alpha, r_\alpha^c) + \mathrm{hr}_{T_\alpha}(r_\alpha^c)$. Par la Procédure de Caractérisation, nous avons que $\mathrm{hr}_{T_\alpha}(r_\alpha) = a_\alpha + d_\alpha(r_\alpha^c)$. D'après les Lemmes 5.94.3 et 5.94.5, nous pouvons écrire $\mathrm{hr}_{T_\alpha}(r_\alpha) = a_\delta + d_\delta(r_\alpha^c)$ et, comme C_δ est une (k,q)-caractéristique, il en découle que $\mathrm{hr}_{T_\alpha}(r_\alpha) \leq q$. De plus, pour tout sous-chemin P de T_α joignant r_α à une feuille de T_α, et tel que r_α^c n'est pas un sommet de P, nous savons que P est un sous-chemin de T_β. Étant donné que T_β est un (k,q)-arbre de partitions, nous pouvons en conclure que T_α est q-ramifié.

Soit x un sommet de $cp(T_\alpha)$ qui n'apparait pas dans T_α^r et soit e une arête de T_α qui n'apparait pas dans T_α^r. Par construction nous avons que x est soit un sommet de $cp(T_\beta)$ qui n'apparait pas dans T_β^r soit un sommet de $cp(T_\gamma)$ qui n'apparait pas dans T_γ^r et que e est soit une arête de T_β qui n'apparait pas dans T_β^r soit une arête de T_γ qui n'apparait pas dans T_γ^r. Et, comme T_β et T_γ sont des (k,q)-arbres de partitions et que la Procédure de Jonction ne change pas les largeurs suivant ϕ associées à x et e dans T_α, nous avons que $\phi_{A_\alpha}(\mathcal{P}_x) \leq k$ et $\phi_{A_\alpha}(\mathcal{P}_e) \leq k$, où $\mathcal{P}_x$ est la partition de A_α induite par x et $\mathcal{P}_e$ est la bipartition de A_α induite par e. De plus, pour chaque sommet x de $cp(T_\alpha^r)$ et chaque arête e de T_α^r, nous avons $\ell_\alpha(x) \leq k$ et $\ell_\alpha(e) \leq k$. En faisant appel à la deuxième règle de compatibilité entre ϕ et $(\mathcal{T},\mathcal{X})$, nous avons que $\phi_{A_\alpha}(\mathcal{P}_x) \leq k$ et $\phi_{A_\alpha}(\mathcal{P}_e) \leq k$, où $\mathcal{P}_x$ est la partition de A_α induite par x et $\mathcal{P}_e$ est la bipartition de A_α induite par e. Par conséquent, la largeur suivant ϕ de T_α est au plus k.

En conclusion, T_α est un (k,q)-arbre de partitions de A_α admettant C_δ comme caractéristique. □

Lemme 5.95. *Soit α un nœud de jonction. La Procédure de Jonction garantit que, pour chaque (k,q)-arbre de partitions de A_α, il existe un (k,q)-arbre de partitions $\check{T}_\alpha \preccurlyeq_\alpha T_\alpha$ tel que $\tau_\alpha(\check{T}_\alpha) \in FSC_\alpha$.*

Démonstration. Soient β et γ les fils de α dans $\mathcal{T}$ et soient FSC_β et FSC_γ les ensembles complets de caractéristiques associés aux nœuds β et γ. Nous appelons T_β et T_γ les restrictions de T_α à A_β et A_γ respectivement. Par définition, T_β est un (k,q)-arbre de partitions de A_β, il existe donc un (k,q)-arbre de partitions $\check{T}_\beta \preccurlyeq_\beta T_\beta$ tel que $\tau_\beta(\check{T}_\beta) \in FSC_\beta$. De même, il existe un (k,q)-arbre de partitions $\check{T}_\gamma \preccurlyeq_\gamma T_\gamma$ tel que $\tau_\gamma(\check{T}_\gamma) \in FSC_\gamma$. Nous montrons qu'il est alors possible de construire un (k,q)-arbre de partitions $\check{T}_\alpha$ à partir de $\check{T}_\beta$ et $\check{T}_\gamma$, tel que $\check{T}_\alpha \preccurlyeq_\alpha T_\alpha$ et que $\tau_\alpha(\check{T}_\alpha) \in FSC_\alpha$.

Construction de $\check{T}_\alpha$. Rappelons que, par définition, nous avons $A_\beta \cap A_\gamma = X_\alpha$ et que $X_\alpha = X_\beta = X_\gamma$. Par conséquent, les restrictions de T_β et T_γ à X_α, respectivement notées T^r_β et T^r_γ, sont isomorphes. De plus, par construction, les squelettes [5] T^s_β et T^s_γ sont également isomorphes. Et, grâce aux définitions de $\preccurlyeq_\beta$ et $\preccurlyeq_\gamma$, nous obtenons $\check{T}^t_\beta \cong T^t_\beta$ et $T^t_\gamma \cong \check{T}^t_\gamma$, ce qui implique $\check{T}^s_\beta \cong T^s_\beta \cong T^s_\gamma \cong \check{T}^s_\gamma$. Comme lors des fusions précédentes, nous commençons la constructions de $\check{T}_\alpha$ en fusionnant les sommets de $\check{T}_\beta$ et $\check{T}_\gamma$ qui apparaissent dans $\check{T}^s_\beta \cong \check{T}^s_\gamma$. Puis pour chaque arête xy de $T^s_\beta \cong T^s_\gamma \cong \check{T}^s_\beta \cong \check{T}^s_\gamma$ nous définissons les chemins $P_\beta(x,y)$, $P_\gamma(x,y)$, $\check{P}_\beta(x,y)$ et $\check{P}_\gamma(x,y)$ sous-jacents à xy dans T_β, T_γ, $\check{T}_\beta$ et $\check{T}_\gamma$ respectivement. Enfin, en utilisant la construction de la Propriété 5.93, nous fusionnons $\check{P}_\beta(x,y)$ et $\check{P}_\gamma(x,y)$ dans $\check{T}_\alpha$ en nous basant sur la fusion de $P_\beta(x,y)$ et $P_\gamma(x,y)$ dans T_α. ■

Lemme 5.95.1. *La Construction de $\check{T}_\alpha$ garantit que $T^t_\alpha \cong \check{T}^t_\alpha$.*

Soit xy une arête de $T^s_\alpha \cong \check{T}^s_\alpha$ et soient $P_\alpha(x,y)$, $P_\beta(x,y)$, $P_\gamma(x,y)$, $\check{P}_\alpha(x,y)$, $\check{P}_\beta(x,y)$ et $\check{P}_\gamma(x,y)$ les chemins sous-jacents dans T_α, T_β, T_γ, $\check{T}_\alpha$, $\check{T}_\beta$ et $\check{T}_\gamma$ respectivement. Comme indiqué dans la preuve de la Propriété 5.93, les correspondances entres les sommets et les arêtes de $P_\beta(x,y)$ et $P_\gamma(x,y)$ dans la fusion $P_\alpha(x,y)$ sont conservées lors de la fusion de $\check{P}_\beta(x,y)$ et $\check{P}_\gamma(x,y)$ en $\check{P}_\alpha(x,y)$. De plus, le fait que $\check{T}^t_\beta \cong T^t_\beta$ et que $T^t_\gamma \cong \check{T}^t_\gamma$, implique que tous les sommets de T^t_α ont un correspondant dans $\check{T}^t_\alpha$. Par conséquent nous avons $T^t_\alpha \cong \check{T}^t_\alpha$.

Lemme 5.95.2. *La Construction de $\check{T}_\alpha$ garantit que* $\mathrm{dr}_{T_\alpha}(r_\alpha, r^r_\alpha) = \mathrm{dr}_{\check{T}_\alpha}(\check{r}_\alpha, \check{r}^r_\alpha)$.

Dans un premier temps, si r_α est présent dans T^r_α, alors il est évident que $\mathrm{dr}_{T_\alpha}(r_\alpha, r^r_\alpha) = \mathrm{dr}_{\check{T}_\alpha}(\check{r}_\alpha, \check{r}^r_\alpha) = 0$. Nous considérons alors que r_α n'est pas un sommet de T^r_α. Par construction le sommet r_α ne peut pas être présent à la fois dans T_β et dans T_γ et, sans perte de généralité, nous supposons que r_α est représenté dans T_β, ainsi que dans $\check{T}_\beta$. Nous pouvons alors écrire $\mathrm{dr}_{T_\alpha}(r_\alpha, r^r_\alpha) = \mathrm{dr}_{T_\beta}(r_\beta, r^r_\beta)$. De plus, lors de la Construction de $\check{T}_\alpha$, le chemin $P_\beta(r_\beta, r^r_\beta)$ ne souffre d'aucune modification. Enfin, le fait que $\mathrm{dr}_{T_\beta}(r_\beta, r^r_\beta) = \mathrm{dr}_{\check{T}_\beta}(\check{r}_\beta, \check{r}^r_\beta)$ nous indique que $\mathrm{dr}_{T_\alpha}(r_\alpha, r^r_\alpha) = \mathrm{dr}_{\check{T}_\alpha}(\check{r}_\alpha, \check{r}^r_\alpha)$.

5. Le squelette T^s_β est obtenu par un lissage des sommets de degré 2 dans T^r_β (ou dans T^t_β, ce qui revient strictement au même).

Lemme 5.95.3. *La Construction de $\check{T}_\alpha$ garantit que nous avons* $\mathrm{nr}_{T_\alpha}(x) = \mathrm{nr}_{\check{T}_\alpha}(x)$, *pour tout sommet interne x de $T^t_\alpha \cong \check{T}^t_\alpha$.*

Nous rappelons que $cp(T^t_\alpha) \cong cp(T^t_\beta) \cong cp(T^t_\gamma)$ et que $cp(\check{T}^t_\alpha) \cong cp(\check{T}^t_\beta) \cong cp(\check{T}^t_\gamma)$. Si x est une feuille de $cp(T^t_\alpha)$ alors, par construction, nous avons $\mathrm{nr}_{T_\alpha}(x) = \mathrm{nr}_{T_\beta}(x) + \mathrm{nr}_{T_\gamma}(x)$ et, si x n'est pas une feuille de $cp(T^t_\alpha)$, alors $\mathrm{nr}_{T_\alpha}(x) = \mathrm{nr}_{T_\beta}(x) + \mathrm{nr}_{T_\gamma}(x) - 1$. En parallèle, si x est une feuille de $cp(\check{T}^t_\alpha)$ alors, nous avons $\mathrm{nr}_{\check{T}_\alpha}(x) = \mathrm{nr}_{\check{T}_\beta}(x) + \mathrm{nr}_{\check{T}_\gamma}(x)$ et, si x n'est pas une feuille de $cp(\check{T}^t_\alpha)$, alors $\mathrm{nr}_{\check{T}_\alpha}(x) = \mathrm{nr}_{\check{T}_\beta}(x) + \mathrm{nr}_{\check{T}_\gamma}(x) - 1$. Comme, par définition, $\mathrm{nr}_{T_\beta}(x) = \mathrm{nr}_{\check{T}_\beta}(x)$ et $\mathrm{nr}_{T_\gamma}(x) = \mathrm{nr}_{\check{T}_\gamma}(x)$, nous pouvons écrire $\mathrm{nr}_{T_\alpha}(x) = \mathrm{nr}_{\check{T}_\alpha}(x)$.

Lemme 5.95.4. *La Construction de $\check{T}_\alpha$ garantit que nous avons* $\mathrm{hr}_{T_\alpha}(x) = \mathrm{hr}_{\check{T}_\alpha}(x)$, *pour tout sommet interne x de $T^t_\alpha \cong \check{T}^t_\alpha$.*

Ici encore $cp(T^t_\alpha) \cong cp(T^t_\beta) \cong cp(T^t_\gamma)$ et $cp(\check{T}^t_\alpha) \cong cp(\check{T}^t_\beta) \cong cp(\check{T}^t_\gamma)$. Si x est un sommet interne de $cp(T^t_\alpha)$ alors, par construction, $\mathrm{hr}_{T_\alpha}(x) = \max\{\mathrm{hr}_{T_\beta}(x) + \varepsilon_\beta, \mathrm{hr}_{T_\gamma}(x) + \varepsilon_\gamma\}$, avec ε_β qui vaut 1 si $\mathrm{nr}_{T_\beta}(x) \geq 2$ et 0 sinon, et ε_γ qui vaut 1 si $\mathrm{nr}_{T_\gamma}(x) \geq 2$ et 0 sinon. En parallèle, si x est un sommet interne de $cp(\check{T}^t_\alpha)$ alors, nous avons $\mathrm{hr}_{\check{T}_\alpha}(x) = \max\{\mathrm{hr}_{\check{T}_\beta}(x) + \check{\varepsilon}_\beta, \mathrm{hr}_{\check{T}_\gamma}(x) + \check{\varepsilon}_\gamma\}$, avec $\check{\varepsilon}_\beta$ qui vaut 1 si $\mathrm{nr}_{\check{T}_\beta}(x) \geq 2$ et 0 sinon, et $\check{\varepsilon}_\gamma$ qui vaut 1 si $\mathrm{nr}_{\check{T}_\gamma}(x) \geq 2$ et 0 sinon. D'après le Lemme 5.95.3, nous savons que $\varepsilon_\beta = \check{\varepsilon}_\beta$ et que $\varepsilon_\gamma = \check{\varepsilon}_\gamma$. Comme, par définition, $\mathrm{hr}_{T_\beta}(x) = \mathrm{hr}_{\check{T}_\beta}(x)$ et $\mathrm{hr}_{T_\gamma}(x) = \mathrm{hr}_{\check{T}_\gamma}(x)$, nous pouvons écrire $\mathrm{hr}_{T_\alpha}(x) = \mathrm{hr}_{\check{T}_\alpha}(x)$.

Lemme 5.95.5. *La Construction de $\check{T}_\alpha$ garantit que nous avons $\check{P}_\alpha(x,y) \preccurlyeq P_\alpha(x,y)$, pour toute arête xy de $T^t_\alpha \cong \check{T}^t_\alpha$.*

Ceci découle directement du fait que la Construction de $\check{T}_\alpha$ utilise la fusion définie par la Propriété 5.93.

Lemme 5.95.6. *La Construction de $\check{T}_\alpha$ garantit que $\tau_\alpha(\check{T}_\alpha) \in FSC_\alpha$.*

Soit xy une arête de $\check{T}^s_\alpha \cong \check{T}^s_\beta \cong \check{T}^s_\gamma$ et soient $\check{P}_\alpha(x,y)$, $\check{P}_\beta(x,y)$ et $\check{P}_\gamma(x,y)$ les chemins sous-jacents à xy dans $\check{T}_\alpha$, $\check{T}_\beta$ et $\check{T}_\gamma$ respectivement. D'après la Propriété 5.90, il existe une fusion $\dot{P}_\alpha(x,y)$ de $\tau(\check{P}_\beta(x,y))$ et $\tau(\check{P}_\gamma(x,y))$ telle que $\check{P}_\alpha(x,y)$ respecte $\dot{P}_\alpha(x,y)$. Comme $\tau_\beta(\check{T}_\beta) \in FSC_\beta$ et $\tau_\gamma(\check{T}_\gamma) \in FSC_\gamma$ ainsi que grâce au Lemme 5.94, nous arrivons à la conclusion que $\tau_\alpha(\check{T}_\alpha) \in FSC_\alpha$. □

Corollaire 5.96. *Soit α un nœud de jonction. La Procédure de Jonction garantit que FSC_α est un ensemble complet de (k,q)-caractéristiques pour A_α restreint à X_α.*

Propriété 5.97. *Soit P un chemin de longueur r et Q un chemin de longueur s et soit M une fusion de P et Q. Alors la longueur de M est au plus $r+s-1$.*

Démonstration. Soient $\bar{P}$ et $\bar{Q}$ les extensions de P et Q relativent à M. Soit m_i un sommet de M tel que les sommets p_i et q_i correspondants dans $\bar{P}$ et $\bar{Q}$ sont tous deux issus de subdivisons d'arêtes dans P et Q. Alors la fusion de p_i et q_i produit le même résultat que la fusion des arêtes e_i et f_i ainsi que celle des arêtes e_{i+1} et f_{i+1} dans M. Par conséquent, pour tout sommet m_i de M, nous pouvons assumer que p_i est un sommet de P ou que q_i

est un sommet de Q. Par hypothèse, P contient $r-1$ sommets internes et Q en contient $s-1$. Par conséquent, M contient au plus $r+s-2$ sommets internes. La longueur de M peut donc être majorée par $r+s-1$. □

Lemme 5.98. *La Procédure de Jonction s'exécute en temps constant.*

Démonstration. Soit α un nœud de jonction de $\mathcal{T}$ et soient β et γ les fils de α dans $\mathcal{T}$. Par hypothèse FSC_β et FSC_γ sont des ensembles complets de (k,q)-caractéristiques pour A_β et A_γ restreints à $X_\beta = X_\gamma$. Soient $C_\beta \in FSC_\beta$ et $C_\gamma \in FSC_\gamma$ deux (k,q)-caractéristiques compatibles, et soit C_α le résultat d'une exécution de la Procédure de Jonction sur C_β et C_γ. Par le Lemme 5.61, les tailles de C_β et C_γ sont majorées en fonction de k, q et $|X_\alpha|$. Plus précisément, la taille des arbres T^c_β et T^c_γ sont majorées en fonction de k, q et $|X_\alpha|$. En utilisant le Lemme 5.97, nous pouvons majorer la taille de T^c_α avant contraction en fonction de k, q et $|X_\alpha|$. Les définitions des autres composantes de C_α se font en parcourant les arbres T^c_α, T^c_β et T^c_γ, elles nécessitent donc un temps de calcul constant. La phase de *contraction des tronçons de* T^c_α est ici encore exécutable en temps constant.

Comme la taille d'une fusion est majorée, le nombre de fusions différentes que nous pouvons générer est lui aussi majoré. De plus, le Lemme 5.62 nous indique que le nombre de (k,q)-caractéristiques de FSC_β et FSC_γ est majoré en fonction de k, q et $|X_\alpha|$, il en découle que le nombre de couples de $FSC_\beta \times FSC_\gamma$ est également majoré. Par conséquent, le nombre de caractéristiques étudiées par la Procédure de Jonction est majoré en fonction de k, q et $|X_\alpha|$, tout comme la taille desdites caractéristiques. Cette procédure s'execute donc en temps constant. □

C'est ainsi que s'achève la longue preuve du Théorème 5.30. En effet, les Corollaires 5.72, 5.82 et 5.96 ainsi que les Lemmes 5.65, 5.66, 5.73, 5.83 et 5.98 nous indiquent que, pour chacun des nœuds de $\mathcal{T}$, nous pouvons calculer un ensemble complet de (k,q)-caractéristiques dans un laps de temps que nous pouvons majorer en fonction de k, q et l, où l est la largeur (suposée constante) de la jolie décomposition $(\mathcal{T},\mathcal{X})$ de A. Il nous suffit alors de rappeler que nous utilisons une procédure montante sur $\mathcal{T}$ pour conclure que le problème de reconnaissance du Théorème 5.30 est solvable en temps $\mathcal{O}(f(k,q,l)\cdot|A|)$, avec f une fonction calculable.

5.4.5. Retour sur les largeurs de graphes

Démonstration de la Propriété 5.38

Maintenant que nous avons vu la preuve du Théorème 5.30, nous pouvons revenir sur la démonstration de la Propriété 5.38, laquelle disait : « pour toute fonction de partition λ, compatible avec une jolie décomposition $(\mathcal{D},\mathcal{Y})$ d'un ensemble A, la fonction de partition $max\lambda$ qui associe $\max_{1\le i\le n}\lambda(A_i, A\setminus A_i)$ à chaque partition $(A_1,\dots,A_n)$ de A est également compatible avec $(\mathcal{D},\mathcal{Y})$ ».

Démonstration de la Propriété 5.38. Le fait que λ soit compatible avec $(\mathcal{D},\mathcal{Y})$ implique l'existance des fonctions I_λ et J_λ telles que pour toute partition $\mathcal{P}$ de A :

$$\lambda_{A_\alpha}(\mathcal{P}) = I_\lambda(\lambda_{A_\beta}(\mathcal{P}\cap A_\beta), \mathcal{P}\cap X_\alpha, \bullet),$$

pour tout nœud d'introduction α de $\mathcal{D}$, avec β le fils de α et $\bullet$ l'élément introduit, et :

$$\lambda_{A_\alpha}(\mathcal{P}) = J_\lambda(\lambda_{A_\beta}(\mathcal{P} \cap A_\beta), \lambda_{A_\gamma}(\mathcal{P} \cap A_\gamma), \mathcal{P} \cap X_\alpha),$$

pour tout nœud de jonction α de $\mathcal{D}$, avec β et γ ses fils.

Pour montrer que $max\lambda$ est également compatible avec $(\mathcal{D}, \mathcal{Y})$, nous devons montrer l'existance des fonctions $I_{max\lambda}$ et $J_{max\lambda}$.

Clairement, si $\mathcal{P} = \{P_1, P_2\}$ est une bipartition de A, alors $max\lambda(\mathcal{P}) = \lambda(\mathcal{P})$, étant donné que $\lambda(\{P_1, P_2\}) = \lambda(\{P_2, P_1\})$. Par conséquent, pour chaque bipartition $\mathcal{P}$ de A, la fonction $I_{max\lambda}$ est égale à la fonction I_λ et la fonction $J_{max\lambda}$ est égale à la fonction J_λ.

La fonction $I_{max\lambda}$. Soit α un nœud d'introduction de $\mathcal{D}$, soit β le fils de α et soit $\bullet$ l'élément introduit. Nous considérons alors T_α, l'arbre de partition de A_α, ainsi que T_α^c l'arbre de caractérisation de T_α suivant X_α. Pour chaque arête e de T_α^c, nous définissons les bipartitions $\mathcal{Q}_e$ et $\mathcal{P}_e$ de X_α et A_α respectivement induites par e. De même, pour chaque sommet interne v de T_α^c, nous définissons les partitions $\mathcal{Q}_v$ et $\mathcal{P}_v$ de X_α et A_α respectivement induites par v.

Étant donné que les arêtes de T_α^c induisent des bipartitions, lors de la phase de *définition de ℓ_α*, il nous suffit d'effectuer l'opération $\ell_\alpha(e) := I_\lambda(\ell_\beta(e), \mathcal{P}_e \cap X_\alpha, \bullet)$, pour toute arête e de T_α^c. Et donc, après exécution de cette instruction, nous obtenons :

$$\ell_\alpha(e) = I_\lambda\Big(\ell_\beta(e), \mathcal{Q}_e, \bullet\Big) = I_{max\lambda}\Big(\ell_\beta(e), \mathcal{Q}_e, \bullet\Big) = max\lambda(\mathcal{P}_e).$$

Puis, pour chaque sommet interne v de T_α^c, nous définissons E_v^c, l'ensemble des arêtes de T_α^c qui sont incidentes à v, ainsi que E_v, l'ensemble des arêtes de T_α qui sont incidentes à v. Par hypothèse d'induction, nous supposons que $\ell_\beta(v) = max\lambda(\mathcal{P}_v \cap A_\beta)$. De par le fait que T_α soit un arbre de partitions de A_α, nous avons que $\mathcal{P}_v = \{P_1, \ldots, P_n\}$, avec $n = |E_v|$ et, d'après la définition de $max\lambda$, nous avons $\max \lambda(\mathcal{P}_v) = \max_{1 \le i \le n} \lambda(\{P_i, A_\alpha \setminus P_i\}) = \max_{e \in E_v} \ell_\alpha(e)$. Par conséquent, nous définissons :

$$\ell_\alpha(v) = I_{max\lambda}\Big(\ell_\beta(v), \mathcal{Q}_v, \bullet\Big) = \max\left\{\ell_\beta(v), \max_{e \in E_v^c} \ell_\alpha(e)\right\}.$$

Comme le degré de T_α est majoré en fonction de l (la largeur de $(\mathcal{D}, \mathcal{Y})$), alors le calcule de $I_{max\lambda}$ s'effectue en temps constant.

La fonction $J_{max\lambda}$. Soit α un nœud de jonction de $\mathcal{D}$ et soient β et γ les fils de α. Nous considérons alors T_α, l'arbre de partitions de A_α, obtenu arpès fusion des arbres de partitions T_β et T_γ de A_β et A_γ Nous considérons également T_α^c, l'arbre de caractérisation de T_α suivant X_α, obtenu après une fusion (respectant la précédente) des arbres de caractérisation T_β^c et T_γ^c de T_β et T_γ. De nouveau, pour chaque arête e de T_α^c, nous définissons les bipartitions $\mathcal{Q}_e$ et $\mathcal{P}_e$ de X_α et A_α respectivement induites par e. De même, pour chaque sommet interne v de T_α^c, nous définissons les partitions $\mathcal{Q}_v$ et $\mathcal{P}_v$ de X_α et A_α respectivement induites par v.

Par hypothèse d'induction, nous supposons que, pour chaque arête e et chaque sommet interne v de T_β^c, nous avons $\ell_\beta(e) = max\lambda(\mathcal{P}_e \cap A_\beta)$ et $\ell_\beta(v) = max\lambda(\mathcal{P}_v \cap A_\beta)$. De même,

pour chaque arête e et chaque sommet interne v de T^c_γ, nous avons $\ell_\gamma(e) = max\lambda(\mathcal{P} \cap A_\gamma)$ et $\ell_\beta(v) = max\lambda(\mathcal{P}_v \cap A_\gamma)$. Soit $P^c_\beta(x,y)$ le sous-chemin de T^c_β qui a été fusionné avec le sous-chemin $P^c_\gamma(x,y)$ de T^c_γ en vu d'obtenir le sous-chemin $P^c_\alpha(x,y)$ de T^c_α (avant contraction), et soient $\bar{P}^c_\beta(x,y)$ et $\bar{P}^c_\gamma(x,y)$ les extensions utilisées pour la fusion.

Nous définissons alors $\ell_\alpha(e) := J_\lambda(\ell_\beta(e), \ell_\gamma(e), \mathcal{Q}_e)$, pour chaque arête e de $P^c_\alpha(x,y)$. Comme e définit une bipartition, nous avons $\ell_\alpha(e) = max\lambda(\mathcal{P}_e)$. Puis, pour chaque sommet v de $P^c_\alpha(x,y)$, nous définissons :

$$\ell_\alpha(v) = J_{max\lambda}(\ell_\beta(v), \ell_\gamma(v), \mathcal{Q}_v) = \max\left\{\ell_\beta(v), \ell_\gamma(v), \max_{e \in E^c_v(\alpha)} \ell_\alpha(e)\right\},$$

et, par hypothèse d'induction, nous obtenons :

$$\ell_\alpha(v) = \max\left\{\begin{array}{l} \max\limits_{e \in E_v(\beta)} max\lambda(\mathcal{P}_e \cap A_\beta) \\ \max\limits_{e \in E_v(\gamma)} max\lambda(\mathcal{P}_e \cap A_\gamma) \\ \max\limits_{e \in E^c_v(\alpha)} \ell_\alpha(e) \end{array}\right\} = max\lambda(\mathcal{P}_v).$$

Encore une fois, la borne supérieure sur le degré de T_α implique que le calcul de $J_{max\lambda}$ s'effectue en temps constant. □

Adaptation de l'algorithme aux différentes largeurs

Lors de l'utilisation de l'algorithme que nous venons de décrire afin de décider de la majoration d'une largeur donnée, il nous faut adapter l'algorithme aux propriétés structurelles de la largeur considérée. C'est-à-dire qu'il nous faut vérifier que chaque caractéristique présente dans FSC_α, pour tout nœud α de la jolie décomposition, caractérisent des arbres de décompositions respectant la largeur de graphe associée. Par exemple, les arbres de partitions pour la largeur de branche sont tels que leurs sommets internes sont de degré 3, par conséquents, les caractéristiques référencées par FSC_α doivent être ratachées à des arbres de partitions dont le degré des sommets internes est 3. Nous montrons alors les divers changements à appliquer à notre algorithme, et ce pour chacune des largeurs définies en Section 5.1.1. La Procédure de Départ est donc modifiée de façon à respecter les propriétés structurelles de la largeur en cours de calcul. Nous nous penchons ensuite sur les modifications apportées aux Procédures d'Introduction et de Jonction en fonction des largeurs considérées.

Pour la largeur arborescente, nous n'apportons aucun changement et, pour la largeur de chemin, nous vérifions que nous conservons un arbre de partitions 0-ramifié tout le long de la procédure montante.

Nous nous plaçons ensuite dans le cadre de la largeur arborescente spéciale. La Procédure d'Introduction est modifiée comme ceci. Soit xy l'arête introduite et soit T^c_x le sous-arbre minimum de T^c_α dont les arêtes contenu dans ses feuilles sont incidentes à x ; nous définissons de même T^c_y. Nous ajoutons alors la caractéristique C_α dans FSC_α seulement si T^c_x et T^c_y sont des chenilles, conformément à la définition de la largeur arborescente spéciale. La Procédure de Jonction reste, quant-à elle, inchangée.

En ce qui concerne les autres largeurs qui sont, à savoir, les largeurs de branche, linéaire, de coupe et de découpe, nous modifions la Procédure d'Introduction ainsi que la Procédure de Jonction comme il suit. Nous commençons par la Procédure de Jonction, durant laquelle, étant donnés deux chemins P_β^c et P_γ^c que nous souhaitons fusionner ensemble, nous forçons les sommets internes des P_β^c à être « fusionnés » avec des arêtes de P_γ^c, et inversement, les sommets internes de P_γ^c sont alors nécessairement « fusionnés » avec des arêtes de P_β^c. Pour le Procédure d'Introduction la restriction est un peu moins évidente. En effet, il nous faut différencier la racine r_β^c de T_β^c du reste de ses sommets internes. Si l'insertion du nouvel élément ne ce fait pas au niveau de r_β^c, alors nous interdisons tout simple les Procédures d'Introductions de Types 1 et 2 (seul la subdivision d'arête est autorisée). Vient ensuite le cas de la racine r_β^c, laquelle peut exceptionnellement être de degré 2, auquel cas nous autorisons les Procédures d'Introductions de Types 1 et 2 à travailler avec. Par contre, si la racine est de degré 3, alors nous la considérons comme n'importe quel autre sommet internes, et l'utilisation des Procédures d'Introductions de Types 1 et 2 sont alors prohibées.

CHAPITRE

6

CONCLUSION

Cette thèse nous a permis d'explorer un ensemble de largeurs de graphes sous différentes formes : arbres de décompositions, grammaires de graphes, arbres de super-groupes et fonctions de partition sous-modulaires. Au cours de ces travaux de recherche, nous avons soumi ces largeurs à diverses comparaisons, faisant apparaître un certain nombre de similitudes et de disparités.

Notre aventure a débuté dans le Chapitre 3 avec une étude comparative de la largeur arborescente et de la largeur de clique. Le premier résultat que nous avons obtenu nous a permis d'établir que la largeur de clique d'un graphe scindé est majorée par une fonction exponentielle de la largeur arborescente de l'hyper-graphe avec lequel il est en bijection ; cette fonction devient polynomiale si nous fixons une borne supérieure sur la taille des hyper-arêtes. Puis nous avons appliquer une première restriction sur les hyper-graphes afin de nous concentrer sur le cas des graphes, pour lequels le comportement de la largeur arborescente est équivalent (à quelques optimisations près) à la largeur de clique des graphes de scissions associés. La seconde restriction nous a amené à considérer les graphes de scissions des graphes complets. Nous avons alors été en mesure de déterminer avec exactitude la largeur de clique de ces graphes, améliorant ainsi un résultat de Makowsky et Rotics. L'obtention de ce résultat est en grande partie due à l'utilisation de la notion d'arbres de super-groupes, qui semble être un outil efficace pour la détermination, encore difficile de nos jours, de minorants sur la largeur de clique des graphes. Enfin nous avons étendu une partie de ces résultats aux graphes d'incidence, aussi bien dans le cas non-orienté que dans le cas orienté.

La démonstration que nous avons utilisée pour notre premier résultat, portant sur l'ensemble des hyper-graphes et de leurs graphes de scissions, est une adaptation de la

preuve d'un résultat étonnamment proche de Corneil et Rotics. Cette forte ressemblance nous amène à considérer l'existance d'une démonstration visant, d'une part, à généraliser ces deux résultats et, d'autre part, à prendre en compte un sous-ensemble de graphes bipartis plus large que les graphes d'incidence. Il semblerait de plus qu'une technique de ré-écriture de termes soit particulièrement bien adaptée à l'écriture d'une telle démonstration, ce qui améliorerait grandement sa compréhension, comme nous avons pu nous en rendre compte lorsque nous avons traiter le cas des graphes.

En ce qui concerne les graphes de scissions des graphes, l'établissement d'un algorithme polynomial décidant de la valeur exacte de la largeur de clique d'un graphe scindé en fonction de la largeur arborescente du graphe sous-jacent, nous donnerait une preuve immédiate de la NP-complétude du problème de décision lié à la largeur de clique des graphes scindés, qui se trouve être un problème encore ouvert. Une idée d'algorithme consiste à caractériser les quatre classes de graphes scindés $\mathcal{C}_1,\ldots,\mathcal{C}_4$, définies par $\mathcal{C}_i = \{G : \mathrm{cw}(\mathcal{S}(G)) = \mathrm{tw}(G) + i\}$, puis à vérifier qu'il existe un algorithme polynomial testant l'appartenance ou non d'un graphe à chacune de ces classes (le cas des classes $\mathcal{C}_1$ et $\mathcal{C}_2$ semble ne pas être trop compliqué).

Il y a aussi des conséquences concernant la vérification de propriétés logiques des graphes à l'aide d'automates. En effet les Théorèmes 2.14 et 2.15 qui nous donnent une majoration exponentielle de la largeur de clique d'un graphe en fonction de sa largeur arborescente rend difficile la construction d'automates de taille raisonable pour la largeur de clique. Le fait que la largeur de clique des graphes d'incidence soit majorée de façon linéaire en fonction de la largeur arborescente du graphes sous-jacent (laquelle est un majorant de la largeur arborescente du graphe d'incidence considéré) permet la construction d'automates de taille plus raisonnable. Cette propriété s'étend également, dans une certaine mesure, aux graphes bipartis dont nous majorons le degré des sommets de l'une des deux partitions.

Dans le Chapitre 4, nous étudions la largeur arborescente spéciale qui est, comme nous l'avons vu, une largeur inspirée à la fois de la largeur arborescente et de la largeur de clique. Nous obtenons alors des résultats de complexité pour diverses classes de graphes ainsi que des propriétés structurelles pour certains graphes de largeur arborescente spéciale majorée par une constante. Nous montrons notamment qu'il existe une caractérisation par mineurs interdits des graphes de largeur arborescente spéciale au plus 2, alors que les classes de graphes de largeur arborescente spéciale au plus k, pour tout $k \geq 3$, ne sont pas stables par minoration.

Les résultats de complexité que nous obtenons semblent montrer que la largeur arborescente spéciale à un comportement qui est souvent plus proche de la largeur de chemin que de la largeur arborescente « classique », comme par exemple pour les graphes étoilés et par conséquent les graphes triangulés. Il existe cependant des graphes dont le comportement est plus proche de la largeur arborescente « classique » que de la largeur de chemin, comme en témoignent les graphes de chemins enracinés. Dans notre étude, nous avons également amorcé une preuve de la NP-complétude supposée du problème de décision lié à la largeur arborescente spéciale appliqué aux graphes

bipartis et bipartis triangulés, à l'aide d'une technique rapprochant une fois de plus la largeur arborescente spéciale des graphes considérés de leur largeur de chemin.

Une autre classe de graphes qui a retenu notre attention est la classe des graphes k-étoilés, pour laquelle Gustedt [30] nous fourni un algorithme polynomial résolvant le problème de décision lié à la largeur de chemin. Bien que l'étude de cette classe de graphe nous ait conduit à la conclusion que la largeur arborescente spéciale d'un graphe k-étoilé est relativement proche de sa largeur de chemin, nous n'avons pas été en mesure d'adapter l'algorithme susmentionné au problème de décision lié à la largeur arborescente spéciale. Il serait assez surprenant que ce problème soit NP-complet, alors que les problèmes de décisions liés aux largeur arborescente « classique » et de chemin sont polynomiaux pour cette classe de graphes.

Dans le cadre des graphes de largeur arborescente majorée par une constante, nous avons montré que la largeur arborescente spéciale des graphes de Halin peut, elle aussi, être majorée par une constante, ce qui, à part dans le cas des forêts, ne semble pas être le cas des autres classes de graphes de largeur arborescente majorée présentes dans la Table 4.1. En effet, en examinant un graphe planaire extérieur obtenu à partir du graphe complet K_3 puis en ajoutant, pour chaque arête de la face extérieure et de façon récursive, un sommet que nous connectons aux extrémités de cette arête, nous nous sommes convaincu que la largeur arborescente spéciale du graphe ainsi obtenu est égale à sa largeur de chemin. Et, comme les graphes planaires extérieurs forment une sous-classe des graphes séries-parallèles, mais aussi des graphes planaires k-extérieurs, il semblerait que la largeur arborescente spéciale de ces deux classes ne puisse pas être majorée par une constante.

L'étude que nous avons menée a surtout mis en avant une comparaison des largeurs arborescente et de chemin avec la largeur arborescente spéciale. Une comparaison entre la largeur de clique et la largeur arborescente spéciale est à envisager, bien que nous n'ayons pas la même abondance de résultats pour la largeur de clique (comparativement à la largeur arborescente).

Le Chapitre 5 présente un méta-algorithme linéaire à paramètre fixé permettant de décider de la largeur (définie par une fonction de partition sous-modulaire) d'un ensemble. Ce méta-algorithme s'applique particulièrement bien au cas des graphes, en considérant des fonctions de partitions de leurs ensembles de sommets ou d'arêtes. Nous obtenons alors un méta-algorithme regroupant les problèmes de reconnaissance liés aux largeurs arborescente q-ramifiées (englobant la largeur arborescente et la largeur de chemin) et arborescente spéciale ainsi qu'aux largeurs linéaire, de branche, de coupe et de découpe, et ce en exprimant ces largeurs sous forme de fonctions de partition sous-modulaires. Ceci nous permet d'énoncer les premiers algorithmes linéaires à paramètre fixé pour la largeur arborescente q-ramifiée ainsi que pour la largeur arborescente spéciale (en parallèle avec [7], pour cette dernière) et de généraliser les résultats de [5, 9, 10, 49].

La largeur de rang (*rank-width*), définie par Oum et Seymour [41], est elle aussi exprimable à l'aide d'une fonction de partition sous-modulaire [1], il serait alors intéressant

de voir si cette fonction remplie les contraintes nécessaires à l'utilisation de notre méta-algorithme, definissant ainsi un algorithme linéaire à paramêtre fixé pour la largeur de rang. Une question supplémentaire consiste à savoir quelles sont les autres largeurs de graphes pouvant entrer dans le cadre des fonctions de partitions sous-modulaires.

Le méta-algorithme que nous présentons ne résoud que le problème de décision. En effet, il ne nous permet pas de construire un arbre de partitions de largeur au plus k, dans le cas d'une réponse positive. Dans un travail en cours il apparait que la technique utilisée par Bodlaender et Kloks [5] afin de rendre leur algorithme de décision constructif peut être adaptée à notre problématique.

ANNEXE

A

DIVISIONS DE SOMMETS

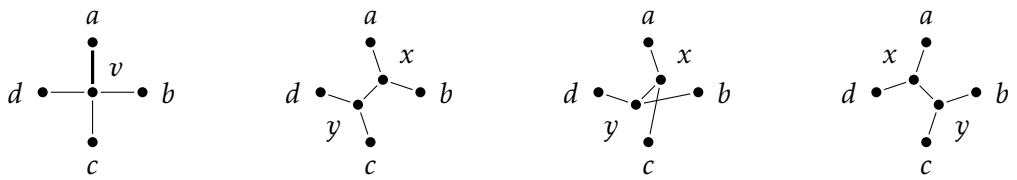
Dans cette annexe, nous discutons plus en détail des mineurs topologiques que nous utilisons dans la preuve du Lemme 4.31. Notre étude porte sur les familles des mineurs topologiques minimaux correspondant aux différentes divisions de sommets qu'il est possible d'opérer sur les graphes S_3, F_1, F_2 et F_3 (voir Figure 4.3).

Diviser un sommet de degré 4. Les divisions que nous utilisons ici ont pour but de réduire le nombre de sommets de degré 4 des graphes considérés, afin de pouvoir utiliser notre argument sur les mineurs de degré au plus 3. Par conséquent, nous laissons de côté les divisions qui consistent à remplacer un sommet de degré 4 par un autre sommet de degré 4 et un sommet de degré 2. Cette opération est équivalente à une subdivision d'arête et est totalement inutile dans la recherche de mineurs toplogiques minimaux. Les divisions sur lesquelles nous nous attardons sont donc celles qui divisent un sommet de degré 4 en deux sommets de degré 3.

Soit un sommet v et ses quatre voisins a, b, c et d. Si v se divise en x et y, nous observons trois types de divisions :

1. a et b sont voisins de x et c et d sont voisins de y ;
2. a et c sont voisins de x et b et d sont voisins de y ;
3. a et d sont voisins de x et b et c sont voisins de y.

Afin de répertorier uniformément ces trois types de divisions, nous devons fixer un plongement des graphes que nous voulons diviser. Puis, pour chacun des sommets de degré 4, nous choisissons une *arête de référence*, qui est celle reliant le sommet v au sommet a. Ensuite, nous tournons autour de v dans le sens anti-trigonométrique et nous indexons les voisins de v que nous rencontrons avec b, puis c et enfin d. Une fois l'indexation terminée nous sommes en mesure de recenser les trois types de divisions



(a) Arête de référence. (b) Division de type 1. (c) Division de type 2. (d) Division de type 3.

Figure A.1. : Les trois types de divisions d'un sommet de degré 4.

de chaque sommet de degré 4. Une illustration de l'utilisation de l'arête de référence ainsi que des trois divisions qui en résultent se trouve en Figure A.1.

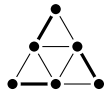
Remarque A.1. Si H est un mineur de G alors H est aussi un mineur de chacun des graphes obtenus par division d'un sommet de G.

Les divisions dans S_3. Premièrement, si nous appliquons, à un seul sommet de degré 4 de S_3, une division de type 2 ou 3, alors nous obtenons un graphe admettant un K_4 comme mineur (voir Figures A.3a et A.3b). La Remarque A.1 nous permet de ne considérer que les divisions de type 1. Il nous reste donc à considérer les quatre graphes illustrés par les Figures A.2a-d, qui ont tous les quatre une largeur arborescente spéciale égale à 3.

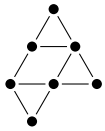
Les divisions dans F_1. Si nous divisons le sommet d'articulation de F_1 en utilisant une division de type 1 ou 2, alors nous obtenons le graphe illustré par la Figure A.3c qui admet le graphe θ_2 comme mineur. Il reste donc la division de type 3 (Figure A.2j) qui produit un graphe de largeur arborescente spéciale 3 ainsi que le graphe F_1 non divisé (Figure A.2i) lui aussi de largeur arborescente spéciale 3.

Les divisions dans F_2. Ici encore, si nous divisons le sommet d'articulation de F_2 avec une division de type 1 ou 2, nous faisons apparaître un mineur θ_2 (voir Figure A.3f). L'utilisation des divisions de type 2 et 3 pour l'autre sommet de degré 4 de F_2 forme des graphes avec un mineur K_4 (illustration en Figures A.3d et A.3e). Il nous reste donc les quatre combinaisons des Figures A.2e-h et formées par la division de type 3 pour le sommet d'articulation et la division de type 1 pour l'autre sommet.

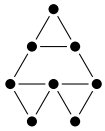
Les divisions dans F_3. Les divisions de type 1 et 2 du sommet d'articulation produisent maintenant un mineur S_3 (Figure A.3i). Les divisions de type 2 et 3 des deux autres sommets de degré 4 font apparaître un mineur K_4 (Figures A.3g et A.3h). Il nous reste finalement six configurations, toutes de largeur arborescente spéciale 3, obtenues grâce à la division de type 3 pour le sommet d'articulation et à la division de type 1 pour chacun des deux autres sommets. Ces six graphes sont illustrés Figures A.2k-p.



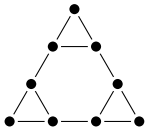
(a) S_3 : division 000.



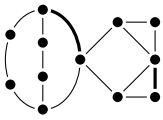
(b) S_3 : division 001.



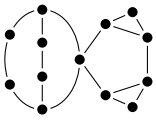
(c) S_3 : division 011.



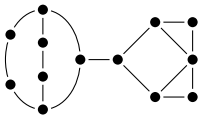
(d) S_3 : division 111.



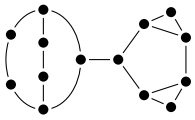
(e) F_2 : division 00.



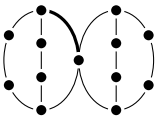
(f) F_2 : divison 01.



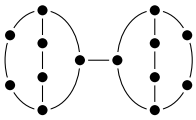
(g) F_2 : divisons 30.



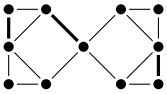
(h) F_2 : divison 31.



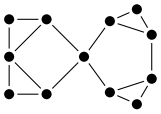
(i) F_1 : divison 0.



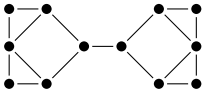
(j) F_1 : division 3.



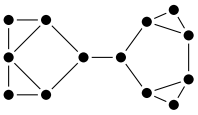
(k) F_3 : divison 000.



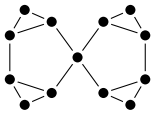
(l) F_3 : divison 001.



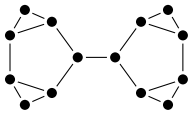
(m) F_3 : divison 030.



(n) F_3 : divison 031.

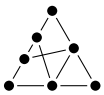


(o) F_3 : divison 101.

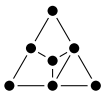


(p) F_3 : divison 131.

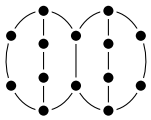
FIGURE A.2. : Liste des mineurs topologiques minimaux « utiles » des graphes S_3, F_1, F_2 et F_3. Les arêtes en gras désignent les arêtes de références.



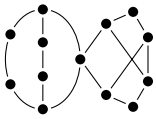
(a) S_3 : division 002.



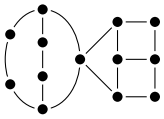
(b) S_3 : division 003.



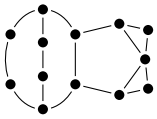
(c) F_1 : division 1 ou 2.



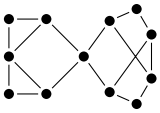
(d) F_2 : divison 02.



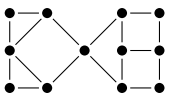
(e) F_2 : divison 03.



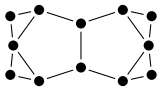
(f) F_2 : division 10 ou 20.



(g) F_3 : divison 002.



(h) F_3 : divison 003.



(i) F_3 : divison 010 ou 020.

FIGURE A.3. : Liste de certains mineurs topologiques minimaux « inutiles » des graphes S_3, F_1, F_2 et F_3.

BIBLIOGRAPHIE

[1] Omid Amini, Frédéric Mazoit, Nicolas Nisse, and Stéphan Thomassé. Submodular partition functions. *Discrete Mathematics*, 309(20):6000–6008, 2009.

[2] Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987.

[3] Hans L. Bodlaender. A Tourist Guide through Treewidth. *Acta Cybern.*, 11(1-2):1–21, 1993.

[4] Hans L. Bodlaender. A Linear-Time Algorithm for Finding Tree-Decompositions of Small Treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[5] Hans L. Bodlaender and Ton Kloks. Efficient and Constructive Algorithms for the Pathwidth and Treewidth of Graphs. *J. Algorithms*, 21(2):358–402, 1996.

[6] Hans L. Bodlaender, Ton Kloks, and Dieter Kratsch. Treewidth and Pathwidth of Permutation Graphs. In Andrzej Lingas, Rolf G. Karlsson, and Svante Carlsson, editors, *ICALP*, volume 700 of *Lecture Notes in Computer Science*, pages 114–125. Springer, 1993.

[7] Hans L. Bodlaender, Stefan Kratsch, and Vincent J. C. Kreuzen. Fixed-Parameter Tractability and Characterizations of Small Special Treewidth. In Andreas Brandstädt, Klaus Jansen, and Rüdiger Reischuk, editors, *WG*, volume 8165 of *Lecture Notes in Computer Science*, pages 88–99. Springer, 2013.

[8] Hans L. Bodlaender and Rolf H. Möhring. The Pathwidth and Treewidth of Cographs. In John R. Gilbert and Rolf G. Karlsson, editors, *SWAT*, volume 447 of *Lecture Notes in Computer Science*, pages 301–309. Springer, 1990.

[9] Hans L. Bodlaender and Dimitrios M. Thilikos. Constructive Linear Time Algorithms for Branchwidth. In Pierpaolo Degano, Roberto Gorrieri, and Alberto

Marchetti-Spaccamela, editors, *Automata, Languages and Programming, 24th International Colloquium, ICALP'97, Bologna, Italy, 7-11 July 1997, Proceedings*, volume 1256 of *Lecture Notes in Computer Science*, pages 627–637. Springer, 1997.

[10] Hans L. Bodlaender and Dimitrios M. Thilikos. Computing Small Search Numbers in Linear Time. In Rodney G. Downey, Michael R. Fellows, and Frank K. H. A. Dehne, editors, *Parameterized and Exact Computation, First International Workshop, IWPEC 2004, Bergen, Norway, September 14-17, 2004, Proceedings*, volume 3162 of *Lecture Notes in Computer Science*, pages 37–48. Springer, 2004.

[11] Arthur Cayley. A theorem on trees. *Quart. J. Pure Appl. Math.*, 23:376–378, 1889.

[12] Maria Chudnovsky, Neil Robertson, Paul D. Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of mathematics*, 164(1):51–229, 2006.

[13] Derek G. Corneil, Michel Habib, Jean-Marc Lanlignel, Bruce A. Reed, and Udi Rotics. Polynomial-time recognition of clique-width ≤ 3 graphs. *Discrete Applied Mathematics*, 160(6):834–865, 2012.

[14] Derek G. Corneil and Udi Rotics. On the Relationship Between Clique-Width and Treewidth. *SIAM J. Comput.*, 34(4):825–847, 2005.

[15] Bruno Courcelle. The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs. *Inf. Comput.*, 85(1):12–75, 1990.

[16] Bruno Courcelle. The Monadic Second-Order Logic of Graphs. III. Tree-decompositions, minor and complexity issues. *ITA*, 26:257–286, 1992.

[17] Bruno Courcelle. Special tree-width and the verification of monadic second-order graph properties. In Kamal Lodaya and Meena Mahajan, editors, *FSTTCS*, volume 8 of *LIPIcs*, pages 13–29. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.

[18] Bruno Courcelle. On the model-checking of monadic second-order formulas with edge set quantifications. *Discrete Applied Mathematics*, 160(6):866–887, 2012.

[19] Bruno Courcelle. Clique-width and edge contraction. *Inf. Process. Lett.*, 114(1-2):42–44, 2014.

[20] Bruno Courcelle and Irène Durand. Fly-Automata, Their Properties and Applications. In Béatrice Bouchou-Markhoff, Pascal Caron, Jean-Marc Champarnaud, and Denis Maurel, editors, *Implementation and Application of Automata - 16th International Conference, CIAA 2011, Blois, France, July 13-16, 2011. Proceedings*, volume 6807 of *Lecture Notes in Computer Science*, pages 264–272. Springer, 2011.

[21] Bruno Courcelle, Pinar Heggernes, Daniel Meister, Charis Papadopoulos, and Udi Rotics. A characterisation of clique-width through nested partitions. Submitted to DAM, 2014.

[22] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear Time Solvable Optimization Problems on Graphs of Bounded Clique-Width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

[23] Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.

[24] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[25] Paul F. Dietz. Intersection graph algorithms. Technical report, Cornell University, 1984.

[26] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.

[27] Wolfgang Espelage, Frank Gurski, and Egon Wanke. Deciding Clique-Width for Graphs of Bounded Tree-Width. *J. Graph Algorithms Appl.*, 7(2):141–180, 2003.

[28] Michael R. Fellows, Frances A. Rosamond, Udi Rotics, and Stefan Szeider. Clique-Width is NP-Complete. *SIAM J. Discrete Math.*, 23(2):909–939, 2009.

[29] Fănică Gavril. Algorithms for Minimum Coloring, Maximum Clique, Minimum Covering by Cliques, and Maximum Independent Set of a Chordal Graph. *SIAM J. Comput.*, 1(2):180–187, 1972.

[30] Jens Gustedt. On the Pathwidth of Chordal Graphs. *Discrete Applied Mathematics*, 45(3):233–248, 1993.

[31] Michel Habib and Rolf H. Möhring. Treewidth of cocomparability graphs and a new order-theoretic parameter. *Order*, 11(1):47–60, 1994.

[32] Pinar Heggernes, Daniel Meister, and Udi Rotics. Computing the Clique-Width of Large Path Powers in Linear Time via a New Characterisation of Clique-Width. In Alexander S. Kulikov and Nikolay K. Vereshchagin, editors, *CSR*, volume 6651 of *Lecture Notes in Computer Science*, pages 233–246. Springer, 2011.

[33] Petr Hlinený, Sang-il Oum, Detlef Seese, and Georg Gottlob. Width Parameters Beyond Tree-width and their Applications. *Comput. J.*, 51(3):326–362, 2008.

[34] Ton Kloks and Hans L. Bodlaender. Approximating Treewidth and Pathwidth of some Classes of Perfect Graphs. In Toshihide Ibaraki, Yasuyoshi Inagaki, Kazuo Iwama, Takao Nishizeki, and Masafumi Yamashita, editors, *ISAAC*, volume 650 of *Lecture Notes in Computer Science*, pages 116–125. Springer, 1992.

[35] Ton Kloks, Hans L. Bodlaender, Haiko Müller, and Dieter Kratsch. Computing Treewidth and Minimum Fill-In: All You Need are the Minimal Separators. In Thomas Lengauer, editor, *ESA*, volume 726 of *Lecture Notes in Computer Science*, pages 260–271. Springer, 1993.

[36] Cornelis G. Lekkerkerker and J. C. Boland. Representation of a finite graph by a set of intervals on the real line. *Fundamenta Mathematicae*, 51:45–64, 1962.

[37] Johann A. Makowsky and Udi Rotics. On the Clique-Width of Graphs with Few P_4's. *Int. J. Found. Comput. Sci.*, 10(3):329–348, 1999.

[38] Daniel Meister. Treewidth and minimum fill-in on permutation graphs in linear time. *Theor. Comput. Sci.*, 411(40-42):3685–3700, 2010.

[39] Rodica Mihai and Ioan Todinca. Pathwidth is NP-Hard for Weighted Trees. In Xiaotie Deng, John E. Hopcroft, and Jinyun Xue, editors, *FAW*, volume 5598 of *Lecture Notes in Computer Science*, pages 181–195. Springer, 2009.

[40] Burkhard Monien and Ivan H. Sudborough. Min Cut is NP-Complete for Edge Weighted Trees. *Theor. Comput. Sci.*, 58:209–229, 1988.

[41] Sang-il Oum and Paul D. Seymour. Approximating clique-width and branch-width. *J. Comb. Theory, Ser. B*, 96(4):514–528, 2006.

[42] Neil Robertson and Paul D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *J. Comb. Theory, Ser. B*, 52(2):153–190, 1991.

[43] Neil Robertson and Paul D. Seymour. Graph Minors .XIII. The Disjoint Paths Problem. *J. Comb. Theory, Ser. B*, 63(1):65–110, 1995.

[44] Neil Robertson and Paul D. Seymour. Graph Minors. XX. Wagner's conjecture. *J. Comb. Theory, Ser. B*, 92(2):325–357, 2004.

[45] Petra Scheffler. *Die Baumweite von Graphen als ein Mass für die Kompliziertheit algorithmischer Probleme*. Report (Karl-Weierstrass-Institut für Mathematik). Akademie der Wissenschaften der DDR, Karl-Weierstrass-Institut für Mathematik, 1989.

[46] Paul D. Seymour and Robin Thomas. Call Routing and the Ratcatcher. *Combinatorica*, 14(2):217–241, 1994.

[47] Karol Suchan and Ioan Todinca. Pathwidth of Circular-Arc Graphs. In Andreas Brandstädt, Dieter Kratsch, and Haiko Müller, editors, *WG*, volume 4769 of *Lecture Notes in Computer Science*, pages 258–269. Springer, 2007.

[48] Dimitrios M. Thilikos. Algorithms and obstructions for linear-width and related search parameters. *Discrete Applied Mathematics*, 105(1-3):239–271, 2000.

[49] Dimitrios M. Thilikos, Maria J. Serna, and Hans L. Bodlaender. Constructive Linear Time Algorithms for Small Cutwidth and Carving-Width. In D. T. Lee and Shang-Hua Teng, editors, *Algorithms and Computation, 11th International Conference, ISAAC 2000, Taipei, Taiwan, December 18-20, 2000, Proceedings*, volume 1969 of *Lecture Notes in Computer Science*, pages 192–203. Springer, 2000.

[50] Jan van Leeuwen. Graph Algorithms. In *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity (A)*, pages 525–631. 1990.

INDEX

A

algorithme constructif 14
ancêtre 7
arbre 7
  enraciné 7
  planaire 51
  q-ramifié 61
arbre de découpe 61
arbre de partitions 63
  q-ramifié 63
  restriction 64
  tronc 85
arc 5
arête 5

C

caractéristique
  compatible 106
  d'un arbre de partition 82
  d'un ensemble 84
  (k,q)-caractéristique 84
chaîne 7
chemin 7
chemin étiqueté 79
  concaténation 87
  contraction 79
  extension 81
  fusion 106
    respectueuse 107
  séparation
    suivant un sommet 95
    suivant une arête 95
clique 7
complément 44
complexité
  paramétrée 14
    à paramètre fixé 14
connexité 7
contraction d'arête 6
corps d'un arbre 63

D

décomposition
  arborescente 8
    jolie 8
    q-ramifiée 61
    spéciale 40
  en branches 61
  en chemin 11
degré 5
  entrant 6
  sortant 6

descendant 7
division de sommet 53

E

ensemble complet
  d'arbres de partitions 73
  de (k,q)-caractéristiques 85
expression
  arborescente 11
  en clique 7
    k-expression 8
    spéciale 41

F

feuille 7
fils 7
fonction de partition 63
  close par sous-ensemble 63
  compatible 68
  monotone 63

G

graphe 5
  biparti 48
  biparti triangulé 49
  complet 7
  de chemins enracinés 50
  de comparabilité 48
  de co-comparabilité 48
  de Halin 51
    ouvert 52
  de permutation 45
  de scission 18
    partiel 24, 28
  d'incidence 35
    partiel 35
  d'intervalles 44
  orienté 5
  scindé 18
  triangulé 47
  vide 7

H

hyper-arête 18
hyper-graphe 18
  multi-hyper-graphe 18

I

incidence 5, 56

J

jolie décomposition 67

K

k-arbre 11
  hyper-k-arbre 18
  partiel 11

L

largeur
  arborescente 8
    spéciale 40
  de branches 61
  de chemin 11
  de clique 7
    spéciale 41
  de coupe 62
  de découpe 62
  linéaire 62
  suivant ϕ 63
lissage de sommet 6

M

mineur 6
  topologique 6

N

nœud
  de départ 8, 67
  de jonction 8, 41, 67
  d'introduction 8, 67
  d'oubli 8, 67

P

père 7
problème
  de décision 14
  de k-reconnaissance 14
  de reconnaissance 14
  d'optimisation 14

R

racine 7

S

sommet 5
  central 18
  périphérique 18
source 11
sous-graphe 6
  induit 6
stable 7
super-groupe 19
  arbre de super-groupes 20
  compatible 19
  hétérogène 32
  homogène 32
  non-singleton 30
  partition en super-groupes 19
  singleton 30

T

tronçon 82

V

voisinage 5, 56
  entrant 6
  sortant 6

Graphes et décompositions

Tom Bouvier

Dans cette thèse nous étudions divers largeurs de graphes gravitant autour des deux largeurs les plus étudiées que sont la largeur arborescente et la largeur de clique. L'importance de ces deux paramètres de graphes est en partie dû aux résultats initiés par Courcelle, qui statuent de l'existance de deux méta-algorithmes polynomiaux à paramètres fixés résolvant les problèmes exprimables à l'aide de la logique monadique du second ordre, si nous considérons les classes de graphes dont la largeur arborescente ou la largeur de clique peuvent être majorées par une constante ; dans le cas général un certain nombre de ces problèmes sont NP-complets.

Les premiers résultats que nous obtenons sont établis en définissant une bijection entre les hyper-graphes et les graphes scindés, laquel nous permet d'exprimer la largeur de clique d'un graphe scindé en fonction de la largeur arborescente de l'hyper-graphe avec lequel il est en bijection. Nous obtenons alors une majoration exponentielle de la largeur de clique, qui n'est pas sans rappeler un résultat de Corneil et Rotics, qui expose un majorant de la largeur de clique d'un graphe en fonction de sa largeur arborescente. Puis, nous restreignons les graphes scindés auxquels nous nous interessons, afin de ne considérer que ceux qui sont en bijection avec les graphes. Le résultat qui en découle est une approximation absolue de la largeur arborescente d'un graphe. Nous avons également été capable de donner une approximation absolue de la largeur de clique d'un graphe scindé en fonction de la largeur arborescente du graphe sous-jacent, faisant alors apparaitre un lien fort entre ces deux paramètres. Une partie des résultats que nous obtenons s'étendent aux graphes d'incidences, avec des conséquences intéressantes concernant la vérification de propriétés logiques à l'aide des automates qui définissent la largeur de clique.

La deuxième partie de la thèse étudie une nouvelle largeur de graphe à mi-chemin entre la largeur arborescente et la largeur de clique, et récemment introduite par Courcelle sous le nom de largeur arborescente spéciale. Nous présentons alors différents résultats de complexité concernant le problème de décision associé à ce nouveau paramètre en fonction d'un certain nombre de classes de graphes. À titre de comparaison, nous rappelons les résultats existant pour les problèmes de décision associé à la largeurs arborescente « classique », à la largeur de chemin ainsi qu'à la largeur de clique. Il apparait alors que le problème de décision associé à la largeur arborescente spéciale est un problème NP-complet dans le cas général. Nous ajoutons à ces résultats de complexité, une caractérisation par mineurs interdits des graphes de largeur arborescente spéciale au plus 2, à l'aide d'une famille de six graphes. Nous montrons également que les classes de graphes de largeur arborescente au plus k, pour tout $k \geq 3$, ne sont pas closes par minoration.

Dans la troisième et dernière partie de cette thèse, nous présentons un méta-algorithme de complexité linéaire à paramètre fixé qui résout les problèmes de reconaissance associés à un certain nombre de largeurs de graphes comprenant, entre autres, les largeurs arborescentes « classique » et spéciale, la largeur de chemin ainsi que les largeurs de branches (*branch-width*) et de découpe (*carving-width*) et leurs versions linéaires. Pour la plus part de ces largeurs il existait déjà de tels algorithmes, tous émanant d'une première solution concernant les largeurs arborescente et de chemin, et présentée par Bodlaender et Kloks. La preuve de bon fonctionnement de ce méta-algorithme est compliquée et s'étend sur une soixantaine de pages.