Université de Caen Basse-Normandie
UFR de Sciences

École doctorale SIMEM

Réductions fines entre problèmes NP-complets

Linéarité, planarité, parcimonie, et minimalité logique

THÈSE

présentée et soutenue publiquement le 19 Décembre 2003

pour l'obtention du

Doctorat de l'université de Caen

Spécialité : Informatique
(arrêté du 25 Avril 2002)

par

M. Régis Barbanchon

Composition du jury

Rapporteurs :	Mme Nadia Creignou, Professeur, Université de la Méditerranée, Marseille
	M. Jacques Mazoyer, Professeur, LIP, Ecole Normale Supérieure de Lyon
	M. Leszek Pacholski, Professeur, Université de Wroclaw, Pologne
Autres membres :	M. Etienne Grandjean, Professeur, GREYC, Université de Caen (directeur)
	M. Vangelis Paschos, Professeur, LAMSADE, Université Paris-Dauphine
	M. Miklos Santha, Directeur de Recherche CNRS, LRI, Université Paris XI

GREYC – CNRS UMR 6072
Groupe de Recherches en Informatique, Image, Automatique et Instrumentation de Caen
Université de Caen campus II – Boulevard du Maréchal Juin 14032 Caen Cedex

Mis en page avec la classe thloria.

Remerciements

Ces travaux n'auraient jamais vu le jour sans l'efficace direction d'Etienne Grandjean. Les aspect logiques des résultats présentés dans ce mémoire sont pour une grande part le fruit de sa collaboration. Etienne est aussi un infatigable relecteur. A ces titres et bien d'autres, ces remerciements lui vont naturellement en premier.

Je remercie les rapporteurs, Nadia Creignou, Jacques Mazoyer et Leszek Pacholski de m'avoir fait l'honneur d'être relecteurs de cette thèse, rendue en retard et en deux fois, la version préliminaire étant largement incomplète.

Je remercie particulièrement Nadia Creignou, qui a piqué ma curiosité sur l'Hamiltonicité planaire il y a quatre ans. Ce problème est au centre d'une partie conséquente des résultats combinatoires présentés dans cette thèse.

Je remercie tous les doctorants de l'équipe algorithmique du GREYC avec qui j'ai partagé ces dernières années, en commençant comme il se doit par les filles : Aline, et euh... Aline donc... et les garçons : Jérémie, Bertrand, Phil, Ben, Bruno, Guillaume.

Je remercie particulièrement Bruno qui a patiemment relu ce mémoire pendant une semaine. Ses nombreuses remarques ont mené ce document à sa version finale, même si je n'ai pu tenir compte de toutes : Puisse donc le lecteur trouver joie à s'énerver sur mon utilisation quelque peu anglo-saxonne (il faut bien l'avouer) des points-virgules et des deux-points, et sur un style caractérisé par l'emploi – cela ne contribue pas à la clarté, j'en conviens – de phrases à rallonges parsemées deci-delà de digressions et de parenthèses.

Je remercie également Guillaume sur les (somme toute larges) épaules duquel je me suis un peu beaucoup appuyé pour la préparation des TP de réseaux la semaine où la première version de ce mémoire a été rendue dans ce qu'on pourrait qualifier d"'une certaine urgence" aux rapporteurs.

Je remercie ceux qui ont été mes comparses de bureau pendant ces quatre années, Jérémie et Bertrand, qui, malgré un caractère pouvant paraître bougon au premier abord, ne sont pas de si mauvais bougres.

Ma compassion va aux pauvres doctorants du bureau d'à côté, Ben et Phil, qui subissent le joug du maître des lieux Jean-Marie, et qui ont appris à craindre sa règle en bois trônant sur son bureau, perpétuelle menace. Courage les amis, l'histoire montre que les despotes tombent toujours.

Je remercie aussi la créature matheuse de l'autre aile, Marc (et ses galettes bretonnes) qui n'a jamais failli à sa visite quotidienne au bureau pendant les terribles chaleurs du dernier mois d'août, lorsque que j'y étais enfermé comme un moine à rédiger laborieusement cette thèse... en compagnie de la charmante Maraluz, qui profitait du PC de Jérémie déjà laissé vacant pour taper son mémoire.

Je n'oublie bien sûr pas Flo, Jasmine et Barbara.

Je remercie le petit malin qui a cru bon un jour de mettre sur sa page web que le japonais est une langue facile, sans orthographe, ni conjugaison compliquée, ni grammaire, non, vraiment, les doigts dans le nez. Deux ans après, je ne suis pas peu fier de clamer que "Rejisu no ronbun" signifie "la thèse de Régis".

Enfin, toutes mes pensées vont au remercié inconnu qui, chaque année, prend un plaisir certain à avancer de trois semaines les dates de qualifications, et qui, dans son infinie sagesse, les a fixées cette année un 26 Décembre.

Table des matières

Table des figures **v**

1 Introduction **1**

2 Préliminaires **7**

2.1 Problèmes combinatoires 8

2.2 Classes de problèmes et réductions 10

2.3 Terminologie pour nos réductions 16

2.4 Un problème bien pratique : $\frac{1}{3}$-Sat 19

3 Réductions parcimonieuses
ou la préservation de la structure des solutions **27**

3.1 Introduction 28

3.2 De la 3-colorabilité à la 3-colorabilité planaire 31

3.3 De la satisfaisabilité à la 3-colorabilité 52

3.4 Conclusion 56

4 Réductions linéaires
ou la préservation de la complexité des problèmes **63**

4.1 Introduction 64

4.2 De la satisfaisabilité planaire à l'Hamiltonicité planaire 66

4.3 De l'Hamiltonicité planaire à la satisfaisabilité planaire 80

4.4 Sat et les problèmes de cardinalité 89

5 Réductions planaires
ou la préservation de la géométrie des instances **93**

5.1 Introduction 94

5.2 L'équivalence des variantes non-planaires de Hamilton 94

5.3 L'équivalence des variantes de Plan-Hamilton 97

5.4 Variantes non-orientées de Plan-Hamilton à degré borné à 3 104

6 Logique et NP-complétude minimale
une caractérisation logique de la classe de SAT **107**
6.1 Introduction 108
6.2 Problèmes locaux et classe LIN-LOCAL 109
6.3 Un problème NP-complet sur des structures bijectives 113
6.4 Un seul prédicat unaire dans la signature et au second ordre 115
6.5 Elimination totale des prédicats unaires dans la signature 117
6.6 L'unique problème minimalement NP-complet 120
6.7 Equivalence parcimonieuse à SAT et NP-complétude minimale 125

Conclusion **129**

Bibliographie **133**

Table des figures

2.1 Schéma d'utilisation d'un crossover dans un problème de coloriage de sommets . 18
2.2 Réduction de Plan-$\frac{1}{3}$-Sat à Plan-Sat 20
2.3 Les gadgets d'inversion et de constantes dans Plan-$\frac{1}{3}$-Sat 20
2.4 Le gadget EQV-NOR dans Plan-$\frac{1}{3}$-Sat 21
2.5 Le gadget simulant la clause $x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4$ dans Plan-$\frac{1}{3}$-Sat 22
2.6 Notre crossover parcimonieux dans Plan-$\frac{1}{3}$-Sat 23
2.7 Le crossover parcimonieux de Lichtenstein dans Plan-Sat 23
2.8 Schéma de planarisation pour Sat et $\frac{1}{3}$-Sat 24
2.9 Réduction non planaire de 3-Col à $\frac{1}{3}$-Sat 24
2.10 Duplication du codage d'un sommet selon son degré 25
2.11 Inversion de l'ordre trigonométrique d'un slot 25

3.1 Transformation quadratique de 3-Col à Plan-3-Col 31
3.2 Propagation des couleurs dans le graphe-diamant 32
3.3 Les douze états locaux et les neuf configurations du graphe-diamant 33
3.4 Propagation des couleurs dans le graphe-diamant modifié par Hunt et al. 34
3.5 Les dix-huit états locaux et les neuf configurations du graphe-diamant modifié . 35
3.6 Les six configurations requises pour le crossover exclusif 38
3.7 Les quatre configurations primaires requises pour le crossover bicolore 38
3.8 Les trois configurations primaires requises pour le prisme 39
3.9 Schéma de construction d'un crossover non-restreint 40
3.10 Les trois configurations primaires requises pour les convertisseurs unidirectionnels 40
3.11 Schéma de construction d'un crossover bicolore 41
3.12 Les cinq config. primaires requises pour les deux convertisseurs bidirectionnels . 42
3.13 Un convertisseur sombre bidirectionnel planaire mais non-parcimonieux 43
3.14 La non-parcimonie du convertisseur est due aux identités sombres 43
3.15 Le convertisseur devient parcimonieux s'il est réinjecté dans lui même 44
3.16 Un crossover exclusif parcimonieux 45
3.17 Un convertisseur sombre bidirectionnel, planaire et parcimonieux 46
3.18 Un convertisseur sombre unidirectionnel, radiant, et parcimonieux 47
3.19 Un prisme radiant et parcimonieux 48
3.20 Un crossover bicolore radiant et parcimonieux 49
3.21 Un crossover non-restreint radiant et parcimonieux 50
3.22 Transformation quadratique et parcimonieuse de 3-Col à Plan-3-Col 51
3.23 Un duplicateur de paires 52
3.24 Une réduction non-parcimonieuse de Sat à 3-Col 53
3.25 Simulation d'une $\frac{1}{3}$-clause (x, y, z) par quatre Nae-3-clauses 54

3.26 Simulation d'une NAE-3-clause (x, y, z) avec 3-COL 54
3.27 Un simulateur parcimonieux et radiant de la $\frac{1}{3}$-clause (x, y, z) 55
3.28 La réduction parcimonieuse de PLAN-$\frac{1}{3}$-SAT à PLAN-3-COL 56
3.29 Comment obtenir des sommets de degré maximal 4 57
3.30 Planarisation du gadget de copie en préservant le degré 57
3.31 Exploitation d'un bon triplet dans un graphe 58
3.32 Trouver un bon triplet (v_1, v_2, v_3) dans un graphe triconnexe non-complet 59
3.33 Trouver un bon triplet (b_i, s, b_j) dans un graphe biconnexe non-triconnexe 60

4.1 Les simulateurs de $\frac{1}{3}$-clause et de $\frac{1}{2}$-clause 66
4.2 L'échelle XOR 68
4.3 La représentation du système $\{(a, c), (a, b, d), (c, d)\}$ 68
4.4 Le graphe obtenu après "expansion" des échelles XOR 69
4.5 La représentation d'une solution du système 69
4.6 Le cycle Hamiltonien associé à la solution 69
4.7 Le crossover décroisant deux échelles XOR 70
4.8 Résolution des croisements des échelles par pose de crossovers 71
4.9 Le nouveau cycle Hamiltonien planaire associé à la solution 71
4.10 Squelette obtenu pour un plongement planaire pour la formule $\varphi = C_1 \wedge ... \wedge C_6$ avec $C_1 = a \vee \neg d$, $C_2 = \neg a \vee c \vee d$, $C_3 = a \vee \neg b \vee \neg c$, $C_4 = \neg c \vee e \vee d$, $C_5 = b \vee e \vee c$, $C_6 = \neg b \vee \neg e$ 73
4.11 Pose des crossovers le long d'un arbre couvrant dans le dual 73
4.12 Ajout de la colonne vertébrale au squelette 73
4.13 Un cycle Hamiltonien représentant la valuation $c = e = faux, a = b = d = vrai$. 74
4.14 Le gadget fonctionnel AND 75
4.15 Les quatre configurations du gadget fonctionnel AND 76
4.16 Le gadget NOT 77
4.17 Le gadget OR 77
4.18 Le gadget de clause pour $\ell_1 \vee \cdots \vee \ell_5$ 78
4.19 Le jeu de configurations sur un flip-flap-flop 78
4.20 Les combinaison légales et illégales sur une série de flip-flap-flops 79
4.21 L'implémentation d'un flip-flap-flop 80
4.22 Les états locaux pour les configurations flip, flap, et flop 80
4.23 Idée 1 : Obtenir une collection de cycles simples disjoints 81
4.24 Idée 2 : Obtenir une structure d'arbres et de monocycles dans le graphe dual . . 81
4.25 Idée 3 : Empêcher l'apparition de monocycles dans le dual 82
4.26 Composantes cycliques et acycliques dans $\mathcal{D}$ 83
4.27 Explosion d'un sommet de degré $d = 6$ en une roue de vélo à d rayons 86
4.28 Plongement du système SAT dans le plan au niveau d'une face 87
4.29 Codage planaire, linéaire et parcimonieux de la contrainte $1/N(x_1, \cdots, x_d)$ . . . 88
4.30 Codage presque planaire d'une contrainte $2/N(x_1, \cdots, x_d)$ 88
4.31 Codage planaire, linéaire et parcimonieux d'une contrainte $2/N(x_1, \cdots, x_d)$ . . . 88
4.32 Réduction linéaire, parcimonieuse et planaire de $\frac{1}{3}$-SAT à VERTEX-COVER 90
4.33 Un schéma simple d'additionneur linéaire 90

5.1 Schéma de d'inter-réductions entre variantes de HAMILTON 94
5.2 Schéma simple de réduction de UHAM-CYCLE à DHAM-CYCLE 95
5.3 Schéma simple de réduction de UHAM-PATH à UHAM-CYCLE 96

5.4 Roue de vélo, version orientée . . . 96
5.5 Schéma simple de réduction de DHam-Cycle à UHam-Cycle . . . 97
5.6 Schéma d'inter-réductions entre variantes de Plan-Hamilton . . . 97
5.7 Notre schéma de réduction de Plan-UHam-Path à Plan-UHam-Cycle . . . 98
5.8 Notre schéma de réduction de Plan-DHam-Path à Plan-UHam-Cycle . . . 99
5.9 Le gadget pour la fonction XOR . . . 100
5.10 Le gadget pour la fonction ODD . . . 100
5.11 Le gadget pour la fonction MAJ . . . 101
5.12 Le gadget pour la fonction ADD . . . 101
5.13 Le petit chandelier simulant un sommet de degré 8 dans un graphe non-orienté . 102
5.14 Les modes de fonctionnement du petit chandelier . . . 102
5.15 Un état local typique pour le petit chandelier en mode 2 ou 3 . . . 103
5.16 Le grand chandelier simulant un sommet de degré 5 dans un graphe orienté . . . 103
5.17 Les modes de fonctionnement du grand chandelier . . . 104
5.18 Un état local typique pour le grand chandelier en mode 2 ou 3 . . . 104
5.19 Comment réduire le degré maximal du gadget AND à 3 . . . 105
5.20 Gadgets EST et CENTRAL en action . . . 105
5.21 Configurations des gadgets CENTRAL et EST . . . 106

6.1 Le gadget DUP . . . 111
6.2 Le gadget PSI . . . 111
6.3 Le gadget VAR . . . 112
6.4 Les gadgets VAR connectés ensemble . . . 112
6.5 Encodage de la clause $x \vee \neg y \vee z \vee t$ pour Plan-Π_1 . . . 114
6.6 Encodage de la clause $\neg x \vee y \vee z$ pour Plan-Π_2 . . . 115
6.7 Le gadget True pour le problème Π_{nand} . . . 118
6.8 Encodage d'une clause pour Plan-Π_{nand} . . . 119
6.9 Les monocycles ont une décomposition arborescente de largeur bornée . . . 123

1 Synthèse des réductions présentées dans ce mémoire . . . 130

1

Introduction

Cette thèse est consacrée à l'étude des problèmes combinatoires NP-complets les plus classiques – ceux qui arrivent en tête du catalogue de Garey & Johnson [33] – et se centre sur la classe des problèmes linéairement équivalents au problème le plus célèbre d'entre tous : SAT, i.e., la satisfaisabilité d'une formule de la logique propositionnelle. Sur cet aspect, nos travaux se situent dans le prolongement de ceux de N. Creignou dans sa Thèse "*Temps Linéaire et Problèmes NP-complets*" [19], ainsi que de ceux de A. Dewdney [25] et E. Grandjean.

L'intérêt des réductions en temps linéaire est d'établir un lien entre les problèmes qui est bien plus étroit que celui établi par les réductions en temps polynomial. En effet, si la réduction en temps polynomial a fait le succès de la classe de complexité NP[1] et celui de ses problèmes NP-complets[2] parce qu'elle a permis la définition d'une classe de complexité large, elle a le défaut de sa qualité en regroupant des problèmes de complexités très diverses pour le temps : Si l'on admet que le meilleur algorithme déterministe pour SAT prend un temps exponentiel $2^{O(n)}$ pour une formule propositionnelle de taille n, alors la réduction à SAT d'une instance de taille n d'un problème NP-complet en temps polynomial $O(n^k)$ laisse suggérer un temps $2^{O(n^k)}$ pour ce problème. Car la plupart du temps, la réduction en temps polynomial $O(n^k)$ est en fait une simple transformation d'instance résultant en une instance de taille également $O(n^k)$. Intuitivement, pour une classe de complexité bien définie pour le temps, un problème complet dans cette classe devrait représenter les problèmes les plus difficiles de la classe. Ce n'est pas le cas de SAT pour NP, pourtant l'archétype des problèmes NP-complets.

Les réductions linéaires ne connaissent pas ce problème : le temps consommé par la réduction étant $O(n)$, l'espace utilisé est nécessairement aussi $O(n)$, et en particulier, l'instance produite par une transformation linéaire est de taille $O(n)$. Si l'on a une réduction linéaire d'un problème A à un problème B, une complexité $f(n)$ pour B se transforme donc en une complexité $O(f(O(n)))$ pour A, qui est aussi $O(f(n))$ pour toutes les fonctions de complexité $f(n)$ communément rencontrées.

De même que la réduction en temps polynomial est l'outil naturel pour étudier la classe NP, la réduction en temps linéaire est associée à la classe NLIN[3] [45, 46, 48]. Si l'on veut obtenir une classe robuste pour le temps linéaire, il faut ici abandonner la machine de Turing (MT) comme modèle de calcul. Une modélisation du temps linéaire existe bien sur MT, et des choses intéressantes sont connues dans ce modèle – en particulier le résultat de séparation de Paul, Pippenger et al. [69], $\mathrm{DTIME}_{\mathrm{MT}}(n) \subsetneq \mathrm{NTIME}_{\mathrm{MT}}(n)$, l'analogue de la conjecture non résolue

[1] la classe des problèmes de décision décidables en temps polynomial sur machine de Turing non-déterministe.

[2] les problèmes dans NP auxquels tout problème dans NP est réductible en temps polynomial sur Machine de Turing déterministe.

[3] l'ensemble des problèmes décidables en temps linéaire non-déterministe.

pour le temps polynomial $\mathrm{P} \overset{?}{\subsetneq} \mathrm{NP}$ – mais d'une part, on ne sait rien calculer d'intéressant en temps linéaire déterministe sur MT, et d'autre part l'intuition que l'on se fait du temps linéaire déterministe n'est absolument pas réalisée par les machines de Turing. La taille de la donnée elle même n'est pas naturelle : l'alphabet d'une MT étant borné, coder un graphe à n sommets et m arêtes par liste de successeurs sur un ruban de la MT prend un espace de l'ordre de $(n+m)\log(m+n)$.

Ce manque de robustesse a amené certains auteurs qui ont voulu raffiner la réduction polynomiale à abandonner le temps linéaire pour un temps plus robuste sur MT : le temps quasi-linaire [77, 14, 49] $\mathrm{NQL}= \mathrm{NTIME}_{\mathrm{MT}}(n(\log n)^{O(1)})$, défini par Schnorr qui montre notamment que SAT est NQL-complet sous réductions dans $\mathrm{QL}= \mathrm{DTIME}_{\mathrm{MT}}(n(\log n)^{O(1)})$. D'autres auteurs, comme E. Grandjean [44, 45] et N. Creignou [19, 20], restant dans le cadre du temps linéaire sur MT avant de l'abandonner, ont montré qu'un certain nombre de problèmes sont linéairement inter-réductibles avec SAT sur MT si l'on autorise celle-ci à effectuer gratuitement un nombre fixé de tris.

Pourtant, le temps linéaire est une notion bien naturelle. Qu'a-t-on envie d'appeler temps linéaire sur un problème de graphe ? Clairement, un temps $O(n+m)$ et non pas $(n+m)\log(m+n)$. Quel problème s'attend-t-on à résoudre en temps linéaire déterministe suer un graphe ? Parcourir ce dernier en profondeur ou en largeur, déterminer les plus courts chemins d'un sommet donné à tous les autres pourvu que le coût des arêtes soit borné, décider s'il est biparti, décider s'il est connexe et sinon déterminer ses composantes maximalement connexes, décider s'il est biconnexe et sinon déterminer ses points d'articulation, calculer un arbre couvrant tous ses sommets, et s'il est planaire, on peut vouloir le trianguler ou calculer son graphe dual : toutes ces tâches se font communément en temps linéaire sur les architectures réelles, et l'on souhaite pouvoir les utiliser dans nos réductions.

En fait, décider de la triconnexité d'un graphe [52], décider de la planarité d'un graphe [53, 10], et si oui calculer son plongement dans le plan [67] et calculer tous les plus courts chemins à source fixée [51] sont également des tâches qui se font en temps linéaire, bien que loin d'être triviales. On sait donc faire beaucoup de choses intéressantes en temps linéaire sous réserve de pouvoir accéder en temps constant à une case mémoire par son adresse (e.g., accéder au k^{eme} élément d'un tableau et accéder au successeur d'un noeud dans une liste chaînée), ce que le modèle de calcul de la MT ne permet pas à cause de ses têtes de rubans ne pouvant se déplacer à chaque pas de calcul que d'une case à gauche ou à droite sur leurs rubans respectifs.

Il nous faut donc une RAM (Random Access Machine), et nous placer dans le cadre des classes $\mathrm{DTIME}_{\mathrm{RAM}}(n)$ et $\mathrm{NTIME}_{\mathrm{RAM}}(n)$, mais pour quel modèle de RAM ? Un modèle présentant une certaine robustesse vis-à-vis de la présentation de la donnée, et qui permet donc de trier une donnée en temps linéaire. Or, on ne sait pas trier n registres en temps significativement inférieur à $n \log n$ sur RAM dès lors qu'il n'y a pas de relation entre la capacité des registres et le nombre n de registres à trier. Certains auteurs ont tenté d'abattre cette barrière et prétendent atteindre le temps linéaire, mais au prix de modèles de calcul RAM irréalistes, dont la particularité la plus gênante est d'autoriser des opérations arithmétiques sur des nombres totalement non bornés en leur affectant un coût constant. Ces modèles exploitent cette particularité pour simuler du parallélisme dans les opérations bit à bit. Nous percevons cette particularité comme une faille dans le modèle de calcul, et y voyons la nécessité de toujours maintenir une borne sur les valeurs manipulées. Comment les borner ? Clairement si l'on veut pouvoir coder de façon aisée un graphe à n sommets et m arêtes en espace $s = O(n+m)$ comme dit plus haut, chaque case mémoire (registre) doit pouvoir stocker une parmi $O(s)$ valeurs, car on doit pouvoir stocker une adresse dans un registre. Dans la pratique algorithmique, et c'est le cas pour toutes les tâches accomplies

en temps linéaire et citées ci-dessus, on n'a jamais besoin de manipuler de valeurs au delà d'une borne linéaire en la taille de la donnée, et c'est donc dans le cadre de ce modèle de calcul RAM, défini et formalisé par E. Grandjean, que prendront place nos réductions linéaires.

Le temps linéaire déterministe, resp. non-déterministe, pour de telles RAM est noté DLIN, resp. NLIN. Toutes nos réductions linéaires seront donc par définition DLIN. Comme NP, la classe NLIN possède ses problèmes complets, dont RISA (Reduction of Incompletely Specified Automaton [42], présent dans le catalogue de Garey & Johnson [33]), mais à la fois peu naturels et peu nombreux. A l'inverse, NLIN a la particularité de contenir les 21 problèmes NP-complets cités par Karp dans son papier fondateur [59] sur la classe NP, et en particulier SAT. Mais SAT et ses problèmes équivalents sous réductions linéaires, n'ont quasiment aucune chance d'être NLIN-complets. En effet, tout comme son analogue polynomial NP, caractérisé par Fagin [28] comme l'ensemble des problèmes capturés par la logique existentielle du second ordre sur les relations, NLIN est caractérisée par une logique existentielle du second ordre sur les fonctions unaires. Du point de vue de la quantité de non-déterminisme utilisée pour une instance de taille n (c'est-à-dire répartie dans n registres contenant chacun un entier $< n$, donc en tout formée de $\Theta(n \log n)$ bits), cela signifie pour NLIN qu'il faut deviner $O(1)$ fonctions sur un domaine de taille $O(n)$, soit une quantité d'information de l'ordre de $n \log n$ bits comme pour la donnée, alors que pour SAT, il suffit de deviner $O(n)$ bits sur un domaine de taille 2 (booléen), soit une quantité d'information de l'ordre de n bits. Bien sûr, nous ne connaissons pas le statut de SAT dans NLIN : prouver qu'il est DLIN impliquerait P = NP et prouver qu'il est NLIN-complet (i.e., réduire linéairement un problème NLIN-complet à SAT) signifierait par la discussion précédente qu'on peut éliminer un grand nombre de pas non-déterministes en utilisant un nombre équivalent de pas déterministes, ce qui pourrait probablement être exploité pour montrer que P = NP. Enfin, prouver que SAT n'est pas NLIN-complet constituerait une preuve de séparation de DLIN $\subsetneq$ NLIN, analogue à $\mathrm{DTIME}_{\mathrm{MT}}(n) \subsetneq \mathrm{NTIME}_{\mathrm{MT}}(n)$ et probablement une piste pour la séparation de P et de NP.

Nous avons donc un problème phare, SAT, à la fois NP-complet et NLIN, qui n'est sans doute pas "maximalement" complet dans NP, c'est-à-dire non NLIN-complet. Par conséquent, nous construisons dans cette thèse une classe taillée spécifiquement à la mesure de SAT, la classe LIN-LOCAL, tout simplement définie comme la clôture de SAT sous réduction DLIN, incluse dans NLIN et où SAT est par définition complet sous réduction DLIN. La classe tire son nom du fait qu'on peut appliquer l'algorithme suivant, non-déterministe et à trois temps, à tout problème appartenant à LIN-LOCAL : Dans un premier temps, un nombre linéaire d'étapes déterministes calculent une structure fonctionnelle ; Dans un deuxième temps, pour chaque élément du domaine, on devine un nombre constant de bits ; Enfin, dans un troisième temps, pour chaque élément du domaine, on vérifie que les bits devinés pour l'élément et ses images satisfont une formule logique uniforme sur le domaine. La première étape prend un temps DLIN, et, si les autres étapes étaient effectuées en parallèle sur chaque élément du domaine, elles prendraient un temps constant.

Dans cette thèse, nous caractérisons logiquement la classe LIN-LOCAL par la clôture par réductions linéaires des problèmes sur structures bijectives vérifiant une formule existentielle monadique du second ordre (EMSO) sur ces structures. Nous exhibons un problème, que nous nommons $\Pi_{\min}$, LIN-LOCAL-complet (et donc NP-complet) sur structures bijectives et caractérisé par une formule EMSO qui est minimale sous plusieurs critères, dans la mesure où le renforcement d'un quelconque de ces critères rend le problème DLIN même si les autres critères sont relâchés : En particulier, le nombre de bijections de la structure est minimal, le préfixe EMSO de la formule est minimal, la longueur de sa partie sans quantificateur est minimale et porte sur un nombre minimal d'atomes, et sa mise en CNF est minimale en nombre de clauses de même que sa DNF en nombre d'anti-clauses. De plus, à quelques symétries naturelles près, la formule caractérisant $\Pi_{\min}$ est l'unique formule réunissant ces caractéristiques.

Le résultat précédent est obtenu par des transformations linéaires entre SAT et notre problème $\Pi_{\min}$. Ces réductions ont une particularité sympathique : elles préservent aussi la planarité, i.e., si l'instance à réduire se traduit par un graphe admettant un plongement dans le plan sans croisement d'arêtes, la transformation produit une instance admettant de même un plongement dans le plan sans croisement d'arêtes. Ceci a son importance car, si on ne connaît pas d'algorithme déterministe sous-exponentiel pour SAT dans le cas général, un algorithme déterministe en $2^{O(\sqrt{n})}$ pour SAT existe dès que le graphe biparti modélisant la relation d'occurrence des variables dans les clauses est planaire [64, 65, 74, 82]. Cet algorithme, qui suit une stratégie "diviser pour régner", exploite récursivement le fait qu'on peut toujours déconnecter un graphe planaire en deux parties de tailles relativement équilibrées en retirant un petit nombre de sommets, de l'ordre de $\sqrt{n}$. Cette plus basse complexité de SAT dans le plan (appelé alors PLAN-SAT) nous pousse a définir la classe LIN-PLAN-LOCAL, la classe des problèmes linéairement réductibles à PLAN-SAT. La préservation de la planarité par nos réductions nous permet de transférer nos résultats précédents sur cette classe, à savoir que PLAN-$\Pi_{\min}$ est LIN-PLAN-LOCAL-complet, et possède les mêmes caractéristiques de minimalité et d'unicité que $\Pi_{\min}$.

Par équivalence linéaire à SAT et PLAN-SAT respectivement, nous établirons la LIN-LOCAL-complétude du problème VERTEX-COVER, et la LIN-PLAN-LOCAL-complétude d'un grand nombre de variantes du problème de l'Hamiltonicité planaire (PLAN-HAMILTON). Ces problèmes ne semblent pas locaux à première vue car, après la phase non-déterministe consistant à deviner $O(n)$ bits, la phase de vérification n'est pas seulement constituée de $O(n)$ vérifications locales, indépendantes, et en temps constant pour chaque bit deviné, mais demande en plus un test prenant un temps séquentiel linéaire sur l'ensemble des bits devinés : Pour VERTEX-COVER, il faut calculer la cardinalité de la couverture devinée afin qu'elle n'excède pas la borne exigée pour la couverture, et pour l'Hamiltonicité, il faut vérifier que l'ensemble des arêtes devinées comme faisant partie de la solution est connexe. Ce que montre la LIN-LOCALITE de ces problèmes est que la phase de vérification DLIN située après la phase non-déterministe devinant la solution peut en quelque sorte être transférée *avant* cette phase. En ce qui concerne PLAN-HAMILTON, ce résultat tient beaucoup à l'aspect planaire du problème, et nous ne savons pas généraliser ce résultat au cas non planaire, ou même simplement au cas d'une surface de genre supérieur à 0 comme le tore.

Il manque une propriété sympathique à notre réduction de SAT à $\Pi_{\min}$: la parcimonie [60], i.e., la préservation du nombre de solutions de l'instance d'entrée dans l'instance de sortie. La parcimonie est une propriété désirable pour une réduction dans la mesure où elle permet de transférer la difficulté du comptage des solutions entre les problèmes, i.e., si un problème A se réduit parcimonieusement à un problème B, alors compter le nombre de solutions d'une instance de B est au moins aussi difficile que de compter celui d'une instance de A. Dans la pratique, les réductions parcimonieuses établissent en fait une bijection entre les solutions des instances d'entrée et de sortie, et l'on peut calculer facilement une solution pour l'instance d'entrée en fonction d'une solution pour l'instance de sortie. La structure des solutions étant préservée, la complexité de l'énumération l'est aussi. La parcimonie permet aussi de préserver la complexité des problèmes qui vont au delà de la décision dans le cadre NP (l'existence d'une solution facilement vérifiable) et demandent s'il existe un nombre fixé de solutions, et en particulier s'il existe une solution unique : De tels problèmes posent à la fois la question de l'existence d'une solution (question dans NP) et de la non-existence de deux solutions distinctes (question dans co-NP), et appartiennent par définition à la classe DP.

Le problème de la satisfaisabilité par un assignement unique (UNIQUE-SAT) et sa variante planaire (UNIQUE-PLAN-SAT) admettant une forme de DP-complétude, il est intéressant d'obtenir des équivalences parcimonieuses à ces problèmes afin de transférer cette difficulté par les

réductions. Nous montrerons dans cette thèse que le problème de la 3-colorabilité des graphes (3-Col) et sa variante planaire (Plan-3-Col), problèmes connus comme étant linéairement équivalents à Sat, sont parcimonieusement équivalents à Sat et Plan-Sat lorsque l'on compte les 3-colorations isomorphes par permutation de couleurs comme formant une seule et même solution. Ce résultat unifie les preuves existantes de NP-complétude et de difficulté de comptage pour la 3-colorabilité [22] et la 3-colorabilité planaire [54, 55], et montre également une forme de DP-complétude pour Unique-3-Col et Unique-Plan-3-Col.

Bien que nous ne connaissions pas de transformation parcimonieuse de Sat (resp. Plan-Sat) à $\Pi_{\min}$ (resp. Plan-$\Pi_{\min}$), nous montrons cependant que notre réduction peut être rendue faiblement parcimonieuse, i.e., un lien fonctionnel est préservé entre le nombre de solutions de l'instance à réduire et de l'instance réduite, ce qui suffit à préserver les résultats de difficulté de comptage, mais pas les autres comme celui de l'énumération. Nous montrons également qu'un sur-problème de $\Pi_{\min}$ (resp. de Plan-$\Pi_{\min}$), i.e., un problème qui raffine sa formule EMSO, possède dans un cadre plus restreint des propriétés de minimalité et d'unicité analogues à celles de $\Pi_{\min}$ (resp. Plan-$\Pi_{\min}$) mais cette fois pour l'équivalence linéaire et parcimonieuse à Sat (resp. Plan-Sat).

Ce mémoire s'organise en quatre chapitres principaux. Les trois premiers sont centrés sur les aspects combinatoires de cette thèse : réductions parcimonieuses, réductions linéaires, et réduction planaires. Les réductions exposées dans cette thèse étant souvent à la fois linéaires, planaires et parcimonieuses, elles ont été réparties dans les trois chapitres combinatoires en fonction de la propriété (parcimonie, linéarité, planarité) qui nous a semblé la plus difficile à établir :

Le chapitre 3, consacré à la parcimonie, regroupe les réductions pour lesquelles l'aspect parcimonieux a été le point de blocage. Nous y établissons tout d'abord que 3-Col se réduit parcimonieusement à Plan-3-Col en temps quadratique, puis nous réutilisons les outils créés lors de cette réduction pour établir l'équivalence parcimonieuse (et linéaire) de 3-Col et Sat d'une part, et de Plan-3-Col et Plan-3-Sat d'autre part. Les résultats de ce chapitre sont à paraître dans un numéro spécial [5] du journal TCS (Theoretical Computer Science).

Le chapitre 4, consacré à la linéarité, regroupe les réductions pour lesquelles l'aspect linéaire a été le point de blocage. Nous y établissons l'équivalence linéaire (et parcimonieuse) de Plan-Sat avec une variante particulière de Plan-Hamilton : l'existence d'un cycle Hamiltonien dans un graphe non orienté. L'équivalence linéaire (et parcimonieuse) de Sat et du problème de cardinalité Vertex-Cover y est également démontrée. Ces résultats sont apparus la première fois dans [3, 4] et ont été publiés dans les actes [7] de la conférence CSL 2002 (Computer Science Logic).

Le chapitre 5, consacré à la préservation de la planarité, regroupe les réductions pour lesquelles l'aspect planaire a été le point de blocage. Ce chapitre étend le résultat précédent en le généralisant pour un grand nombre de variantes de Plan-Hamilton : cycles Hamiltoniens, chemins Hamiltoniens à extrémités fixes ou libres, dans un graphe orienté, non orienté, non orienté de degré 3. Ces résultats sont apparus la première fois dans [3].

Enfin, le chapitre 6 aborde les aspects logiques du mémoire : la classe LIN-LOCAL et l'obtention d'une formule minimale décrivant un problème NP-complet. La définition de LIN-LOCAL est apparue pour première fois dans [6] ainsi que dans les actes [7] de la conférence CSL 2002. La formule minimale décrivant un problème NP-complet fait l'objet d'un article [8] accepté à la conférence STACS 2004 (Symposium on Theoretical Aspect of Computer Science).

2

Préliminaires

Sommaire

2.1 Problèmes combinatoires **8**
2.2 Classes de problèmes et réductions **10**
2.3 Terminologie pour nos réductions **16**
2.4 Un problème bien pratique : $\frac{1}{3}$-Sat **19**

2.1 Problèmes combinatoires

Les problèmes abordés dans cette thèse sont les problèmes combinatoires suivants, pour la plupart des problèmes de décision, i.e., des problèmes pour lesquels on accepte ou on rejette une instance. Les problèmes de décision ci-dessous sont tous issus de problèmes de recherche, i.e., des problèmes pour lesquels on cherche en plus à produire en sortie une solution à la donnée entrée lorsque celle-ci est acceptée :

Problème SAT (satisfaisabilité)
Entrée : Une formule $\varphi(V, L)$ de la logique propositionnelle en CNF, constituée d'une liste de clauses L de longueurs quelconques, construites sur l'ensemble de variables V.
Question : Existe-t-il un assignement I sur l'ensemble V des variables tel que chaque clause dans L contienne au moins un littéral évalué à vrai par I ?

Problème 3-SAT
Définition : La restriction de SAT aux instances contenant uniquement des 3-clauses (clauses à 3 littéraux).

Voici un problème bien pratique et connexe à la Satisfaisabilité que l'on va voir plus en détail dans ces préliminaires et qui va nous servir dans tous les chapitres de cette thèse.

Problème $\frac{1}{3}$-SAT ("exactly one among three satisfiability") [76]
Entrée : Une formule $\varphi(V, L)$ de la logique propositionnelle en CNF, constituée d'une liste L de 3-clauses monotones positives construites sur l'ensemble de variables V. Les 3-clauses sont ici appelées $\frac{1}{3}$-clauses.
Question : Existe-t-il un assignement I sur l'ensemble V des variables tel que chaque clause dans L contienne exactement une variable évaluée à vrai par I, les deux autres étant évaluées à faux ?

Problème NAE-3-SAT ("not all equal satisfiability")
Entrée : Une formule $\varphi(V, L)$ de la logique propositionnelle en CNF, constituée d'une liste L de 3-clauses monotones construites sur l'ensemble de variables V. Les 3-clauses sont ici appelées NAE-3-clauses.
Question : Existe-t-il un assignement I sur l'ensemble V des variables tel que chaque clause dans L contienne au moins une variable évaluée à vrai et une variable évaluée à faux par I ?

Le problème de la 3-colorabilité est l'objet de la totalité de notre chapitre 3 qui étudie ses liens forts avec SAT.

Problème 3-COL (3-colorabilité)
Entrée : Un graphe $G(V, E)$ non orienté.
Question : Existe-t-il un 3-coloriage des sommets de G, i.e., une fonction $C : V \longrightarrow \{white, gray, black\}$ telle que quels que soient $x, y \in V$, si $(x, y) \in E$ alors $C(x) \neq C(y)$?

Problème VERTEX-COVER
Entrée : Un graphe $G(V, E)$ non orienté, et un entier $k \leq |V|$.
Question : Existe-t-il un sous-ensemble C de V de cardinalité $|C| \leq k$, tel que toute arête

$(x, y) \in E$ soit couverte par au moins un sommet, i.e., $x \in C$ ou $y \in C$?

Les problèmes Hamiltoniens seront l'objet des chapitres 4 et 5, où un lien fort avec SAT sera établi dans le cas des restrictions planaires de ces problèmes.

Problème UHAM-CYCLE ("undirected Hamiltonian Cycle")
Entrée : Un graphe $G(V, E)$ non orienté.
Question : Existe-t-il un sous-ensemble de E constituant un cycle Hamiltonien C dans G, i.e., un cycle visitant une fois et une seule chaque sommet de V ?

Problème UHAM-PATH ("undirected Hamiltonian Path with free ends")
Entrée : Un graphe $G(V, E)$ non orienté.
Question : Existe-t-il un sous-ensemble de E constituant un chemin Hamiltonien C dans G, i.e., un chemin d'extrémités quelconques visitant une fois et une seule chaque sommet de V ?

Problème UHAM-PATH(x, y) ("undirected Hamiltonian Path with fixed ends")
Entrée : Un graphe $G(V, E)$ non orienté et deux sommets fixés x et $y \in V$.
Question : Existe-t-il un sous-ensemble de E constituant un chemin Hamiltonien C de x à y dans G, i.e., un chemin de x à y visitant une fois et une seule chaque sommet de V ?

Problèmes DHAM-CYCLE, DHAM-PATH, DHAM-PATH(x, y)
Définitions : Les variantes respectives de UHAM-CYCLE, UHAM-PATH, UHAM-PATH(x, y) sur les graphes orientés ("digraphes"). En ce qui concerne DHAM-PATH(x, y), il sera pratique de considérer que x et y peuvent être tous deux indifféremment points de départ ou points d'arrivée d'un chemin Hamiltonien orienté. On notera UHAM-PATH$(\overrightarrow{x, y})$ la variante pour laquelle on exige que x soit le point de départ et y le point d'arrivée.

Problème HAMILTON ("Hamiltonicité")
Définition : Terme générique pour l'Hamiltonicité et désignant plus particulièrement une variante quelconque parmi UHAM-CYCLE, UHAM-PATH, UHAM-PATH(x, y), DHAM-CYCLE, DHAM-PATH, DHAM-PATH(x, y).

Il est utile de considérer les problèmes connexes à la Satisfaisabilité comme des problèmes de graphes. La raison est qu'on peut alors parler d'instances planaires pour ces problèmes.

Définition 2.1 (Graphe de formule) *Soit $\varphi(V, L)$ une formule de la logique propositionnelle en CNF constituée d'une liste L de clauses construites sur un ensemble de variables V. Le* graphe de formule *G_ϕ est le graphe biparti $(V \cup L, E)$ où E est la relation d'occurrence des variables dans les clauses : $E = \{(v, c) : v \in V, c \in L, c$ contient une occurrence de $v\}$. Les arêtes (v, c) de E sont étiquetées par le signe du littéral sous lequel la variable v apparaît dans e. Une formule φ est dite* planaire *ssi G_φ est lui-même planaire.*

L'avantage des graphes planaires est qu'ils peuvent faire l'objet d'une stratégie de diviser-pour-régner, ce qui permet souvent, même pour les problèmes difficiles, de faire décroître la complexité du problème par rapport à ce qu'on sait faire dans le cas général.

Problème PLAN-Π
Définition : La restriction aux instances planaires d'un problème Π sur les graphes. Une instance planaire est toujours donnée par sa carte planaire, i.e., le plongement du graphe dans le

plan est donné en indiquant pour chaque sommet l'ordre de ses voisins dans un sens fixé, e.g., le sens trigonométrique.

Les problèmes sur formules sont aussi vues comme des problème de coloriage sur leur graphe de formule. En particulier, les instances de PLAN-SAT, PLAN-$\frac{1}{3}$-SAT et PLAN-NAE-3-SAT, sont des formules planaires suivant la définition 2.1.

Convention 2.2 *Dans toutes les figures représentant des graphes de formules, les sommets de forme carrée représentent les sommets de clauses et les sommets de forme circulaire représentent les variables. L'étiquette de l'arête représentant le signe de l'occurrence est parfois représentée (par le symbole ¬) lorsque celle-ci est négative. Les assignements sont représentés par un coloriage des sommets de variables : en blanc lorsque la variable est mise à vrai, en noir lorsque la variable est mise à faux.*

Problème UNIQUE-Π
Entrée : Une instance I d'un problème de recherche Π (ou du problème de décision associé).
Question : L'instance I a-t-elle une solution unique (ou un témoin unique) pour Π ?

Typiquement, UNIQUE-SAT demande si une formule donnée est satisfaite par un assignement unique, UNIQUE-HAMILTON demande si un graphe donné admet un unique cycle (ou chemin) Hamiltonien, UNIQUE-VERTEX-COVER demande si un graphe admet une unique couverture de cardinalité inférieure à un entier donné, etc.

Problème #Π
Entrée : Une instance I d'un problème de recherche Π (ou du problème de décision associé).
Sortie : Combien I a-t-elle de solutions (ou de témoins) pour Π ?

Typiquement, #SAT demande le nombre d'assignements satisfaisant une formule donnée, #HAMILTON demande le nombre de cycles (ou de chemins) Hamiltoniens dans un graphe donné, etc.

2.2 Classes de problèmes et réductions

classe P : Un sous-langage Π (ou problème de décision Π) d'un langage donné L (ensemble des instances de Π) appartient à la classe P, si une MT déterministe peut, pour tout mot (ou instance) w de L, décider l'appartenance de w à Π en temps polynomial $O(|w|^{O(1)})$. Par abus de langage, on dit également qu'une fonction est dans P si elle est calculable en temps polynomial sur MT déterministe.

Karp-réduction ou transformation polynomiale : Soient Π_1 et Π_2 deux problèmes de décision sur les ensembles d'instances respectifs L_1 et L_2. Une Karp-réduction [59] (ou transformation polynomiale) de Π_1 à Π_2 est une fonction $R : L_1 \longrightarrow L_2$ dans P, telle que pour toute instance w_1 de L_1, $w_2 = R(w_1) \in \Pi_2$ ssi $w_1 \in \Pi_1$.

Classe NP : Un sous-langage Π (ou problème de décision Π) d'un langage donné L (ensemble des instances de Π) appartient à la classe NP, s'il existe un programme N sur MT non-déterministe pour lequel :

- pour tout mot w de Π, N admet un calcul acceptant sur l'entrée w en temps polynomial $O(|w|^{O(1)})$, et
- pour tout mot w de $L \setminus \Pi$, N n'admet aucun calcul acceptant.

De façon équivalente, Π est dans NP s'il existe un programme D sur MT déterministe tel que pour tout $w \in \Pi$, il existe un témoin T_w de taille polynomiale, permettant à D de vérifier que $w \in \Pi$ sur l'entrée (w, T_w) en temps polynomial, i.e, D accepte l'entrée (w, T_w) en temps polynomial, et pour tout $w \notin \Pi$ et tout mot T de taille polynomiale, D rejette l'entrée (w, T).

On appelle généralement *solutions* pour un problème combinatoire Π dans NP, les témoins naturels pour Π, i.e., les solutions pour le problème de recherche sous-jacent au problème de décision, e.g., une solution pour une instance SAT est un assignement satisfaisant la formule donnée en entrée, une solution pour HAMILTON est un cycle ou un chemin Hamiltonien dans le graphe d'entrée, une solution pour 3-COL est un 3-coloriage du graphe d'entrée, etc.

Problème NP-complet : Un problème est *C-complet* pour une classe de complexité C et un type de réduction R s'il est lui-même dans la classe C et est *C-dur*. Un problème Π est C-dur si pour tout problème π dans C, il existe une réduction de type R de π à Π. Les problèmes NP-complets utilisent la transformation polynomiale.

Le problème NP-complet fondamental est SAT (théorème de Cook [15]). La réduction universelle de Cook transforme tout programme N sur MT non-déterministe qui tourne en temps $T(n)$ en une instance de SAT de taille $O(T(n)^2)$ qui simule les $T(n)$ étapes de calcul de N. Avec un peu de soin, on peut faire en sorte que cette réduction universelle préserve la structure des solutions, i.e., que l'instance SAT d'arrivée ait autant de solutions que N a de calculs acceptants.

Hormis les problèmes de type UNIQUE-Π et $\#\Pi$, tous les problèmes de décision cités dans la section précédente, $\frac{1}{3}$-SAT, NAE-3-SAT [76, 24], 3-COL, HAMILTON, VERTEX-COVER, sont également des problèmes classiquement NP-complets dans le cas non-planaire. Leur NP-complétude est généralement établie par Karp-réduction de SAT. Et à l'exception de PLAN-NAE-3-SAT qui est polynomial, presque tous restent NP-complets dans le plan, i.e., PLAN-SAT [62], PLAN-3-COL [33], PLAN-HAMILTON [34, 70, 58], PLAN-VERTEX-COVER restent NP-complets. La NP-complétude d'un problème PLAN-Π est généralement établie :

- soit par *planarisation*, i.e., par une Karp-réduction (générant le plus souvent une instance de taille quadratique) du problème général (non-planaire) Π lui-même, qui consiste à remplacer chaque croisement d'arêtes par un graphe planaire, appelé crossover, ayant la propriété de ne pas changer la décision du problème pour le graphe affecté. C'est le cas des problèmes PLAN-SAT, PLAN-$\frac{1}{3}$-SAT, et PLAN-3-COL. Il peut aussi arriver qu'on réduise un autre problème que Π : C'est le cas pour PLAN-HAMILTON dont la preuve classique de NP-complétude est une réduction directe de SAT.
- soit par *Karp-réduction planaire* (générant le plus souvent une instance de taille linéaire) d'un autre problème PLAN-Π', lui-même connu comme étant NP-complet. L'avantage de ces transformations est qu'elles sont souvent naturellement en temps linéaire : le principe est de s'appuyer sur le plongement planaire de l'instance de départ pour produire un plongement de l'instance d'arrivée. C'est le cas de la réduction de PLAN-SAT à PLAN-VERTEX-COVER. Bien souvent, les réductions de PLAN-Π' à PLAN-Π sont également valables pour réduire Π' à Π, ce qui permet de faire coup double (nous verrons que ce n'est pas toujours le cas dans le chapitre 4). Nous rechercherons particulièrement ce genre de réductions dans tous nos chapitres.

Les problèmes comme PLAN-SAT et PLAN-VERTEX-COVER ont des algorithmes sous-exponentiels en $2^{O(\sqrt{n})}$ qui exploitent récursivement le théorème du séparateur planaire de Lipton et Tarjan [64, 65, 74], selon lequel dans tout graphe G planaire, on peut trouver en temps $O(n)$ un ensemble de $O(\sqrt{n})$ sommets dont la suppression déconnecte le graphe en deux parties A et B selon un rapport de taille relativement équilibré (au pire $\frac{1}{3}$ contre $\frac{2}{3}$). L'algorithme sous-exponentiel pour PLAN-SAT consiste à calculer un tel séparateur, essayer tous les $O(\sqrt{n})$ assignements partiels en appliquant à chaque fois l'algorithme récursivement sur A et B séparément. Notons qu'à cause de tels algorithmes, il est difficile d'espérer une planarisation sous-quadratique de SAT ou VERTEX-COVER : ceci impliquerait en effet immédiatement l'existence d'un algorithme sous-exponentiel pour SAT et VERTEX-COVER dans le cas général (i.e., non planaire) [82].

Classe co-NP : Un sous-langage Π (ou problème de décision Π) d'un langage donné L (ensemble des instances de Π) appartient à la classe co-NP, si son complémentaire $L \setminus \Pi$ appartient à NP.
D'après la définition, on voit que les problèmes co-NP-complets sont les problèmes dont le complémentaire est lui-même NP-complet. En particulier, UNSAT, qui demande si une formule en CNF n'est pas satisfaisable, est un problème co-NP-complet.

Classe DP : Un langage Π appartient à DP [68] s'il existe deux langages $\Pi_1 \in$ NP, et $\Pi_2 \in$ co-NP, tel qu'un couple de mots (w_1, w_2) appartient à Π ssi $w_1 \in \Pi_1$ et $w_2 \in \Pi_2$. On notera alors $\Pi = (\Pi_1, \Pi_2)$.

DP contient naturellement NP $\cup$ co-NP. La DP-complétude classique est établie par transformation polynomiale. Tout problème (Π_1, Π_2), pour lequel Π_1 est NP-complet et Π_2 est co-NP-complet, est lui même DP-complet. Un exemple est le problème (SAT, UNSAT).

Parmi les problèmes appartenant à DP, il y a en particulier tous les problèmes de type UNIQUE-Π tels que Π est dans NP. Par exemple UNIQUE-SAT appartient à DP. En effet, demander si une formule F appartient à UNIQUE-SAT revient à demander si (F, F) appartient à (SAT, X) où X est le problème co-NP demandant s'il n'existe pas plus d'un assignement satisfaisant F (un témoin pour X est simplement une paire d'assignements satisfaisants). La même instance F étant utilisée pour les deux problèmes SAT et X, UNIQUE-SAT a peu de chance d'être DP-complet sous réduction déterministe polynomiale. Il existe cependant une notion de DP-complétude pour UNIQUE-SAT utilisant un autre type de réduction que la transformation déterministe polynomiale : la transformation polynomiale aléatoire. Dans une telle réduction d'un problème Π_1 à un problème Π_2, l'instance d'arrivée w_2 peut être produite aléatoirement et la réduction ne préserve pas nécessairement la décision de l'instance de départ w_1 :

- si $w_1 \notin \Pi_1$, la décision est totalement préservée, et alors on a nécessairement $w_2 \notin \Pi_2$;
- si $w_1 \in \Pi_1$, la décision est préservée avec une probabilité au moins $\frac{1}{P(n)}$, où n est la taille de l'instance de départ et $P(n)$ est un polynôme en n, et alors on a $w_2 \in \Pi_2$ avec une probabilité au moins $\frac{1}{P(n)}$.

L'intérêt de cette réduction est qu'un algorithme qui itère un nombre polynomial de fois cette réduction sur w_1 en résolvant l'instance produite w_2 jusqu'à ce que $w_2 \in P_2$ reste lui-même polynomial si le test $w_2 \in \Pi_2$ l'est aussi. Autrement dit, cette réduction préserve RP, la classe des problèmes de décision ayant un algorithme polynomial aléatoire. L.G. Valiant et V.V. Vazirani ont montré que (SAT,UNSAT) se réduit à UNIQUE-SAT sous réduction polynomiale aléatoire [87], ce qui montre que UNIQUE-SAT est DP-complet sous réduction polynomiale aléatoire. Cette notion de DP-complétude n'est pas essentielle dans cette thèse mais est mentionnée ici parce que

certains de nos résultats (sur 3-COL par exemple) permettent d'obtenir de nouveaux résultat de DP-complétude.

Réductions parcimonieuses : Une Karp-réduction R d'un problème Π_1 à un problème Π_2 est appelée *réduction parcimonieuse* si elle préserve le nombre de solutions, autrement dit si l'instance de départ w_1 possède $\#w_1$ solutions pour Π_1, alors $R(w_1)$ a aussi $\#w_1$ solutions pour Π_2.

Obtenir des réductions parcimonieuses est intéressant à plusieurs titres : D'une part, si SAT se réduit parcimonieusement à un problème Π, alors on sait que *Unique*-Π est aussi difficile que UNIQUE-SAT, DP-complet sous réductions polynomiales aléatoires. Par exemple, il est connu que SAT, PLAN-SAT, 3-SAT, PLAN-3-SAT, $\frac{1}{3}$-SAT, PLAN-SAT, HAMILTON et PLAN-HAMILTON sont parcimonieusement inter-réductibles. Par conséquent les versions UNIQUE-Π de ces problèmes Π sont également DP-complètes. Nous verrons certaines de ces réductions dans ces préliminaires ainsi que dans les chapitres centraux de la thèse et nous verrons qu'elles ne sont pas toutes optimales sous d'autres points de vue. D'autre part, les réductions parcimonieuses préservent naturellement la difficulté des problèmes de comptage de type $\#\Pi$:

Classe de comptage #P : Une fonction f à valeurs entières est de classe #P [60] s'il existe une MT non-déterministe M fonctionnant en temps polynomial telle que pour toute entrée w de M, $f(w)$ est le nombre de calculs acceptants de M sur w. De façon équivalente, f est associée à un problème Π de NP tel que pour toute donnée w, $f(w)$ est le nombre de solutions (ou témoins) de Π pour w. On notera alors $f = \#\Pi$.

La réduction pour la classe #P est la Turing-réduction, bien plus large que la Karp-réduction. C'est une réduction polynomiale avec la possibilité supplémentaire d'invoquer un nombre polynomial de fois un oracle dans NP en ne facturant qu'un temps polynomial déterministe pour chaque invocation. L'intérêt de la #P-complétude sous Turing-réduction tient au fait que si un problème #P-complet était prouvé être dans P, alors toute la classe #P le serait aussi. Par contre, on sait depuis Valiant [86] que certains problèmes $\#\Pi$ sont #P-complets alors que le problème de décision sous-jacent Π est lui-même dans P [72, 63, 84].

Le problème #P-complet de base est évidemment #SAT. Les équivalences parcimonieuses citées plus haut entre SAT, 3-SAT, $\frac{1}{3}$-SAT, HAMILTON, ainsi que leurs versions planaires montrent que tous les problèmes $\#\Pi$ associés à chacun de ces problèmes de décision Π sont aussi #P-complets. De façon assez intéressante, la #P-complétude de 3-COL, problème de comptage issu d'un problème NP-complet, a d'abord été obtenue dans la littérature par Linial [63] via une réduction parcimonieuse de #STABLE dans les graphes bipartis, un problème #P-complet dont le problème de décision sous-jacent est dans P et dont la #P-complétude a donc été obtenue par Turing-réduction. Ce n'est que plus tard que sa #P-complétude a été prouvée par transformation *faiblement parcimonieuse* de 3-COL à SAT [22].

Réductions faiblement parcimonieuses : Une réduction faiblement parcimonieuse d'un problème de recherche Π_1 à un problème de recherche Π_2 est une Karp-réduction R de Π_1 à Π_2 munie d'une fonction f de $\mathbb{N}$ dans $\mathbb{N}$ calculable en temps polynomial $O(|w_1|^{O(1)})$ et telle que pour toute instance w_1 de Π_1 et $w_2 = R(w_1)$, on a l'égalité $\#w_2 = f(\#w_1)$, où $\#w_i$ représente le nombre de solutions de Π_i pour l'instance w_i.

Les formes courantes de f sont $f(s) = a \times s$ ou $f(s) = a \times s \times b^n$, voire $f(s) = a \times s \times b^{n^c}$ pour a, b, c constants et n un paramètre dépendant de w_1. La deuxième preuve de #P-complétude de

#3-Col par réduction faiblement parcimonieuse de 3-Col à Sat utilise la deuxième forme : le nombre de solutions y est multiplié par un facteur exponentiel. Elle utilise plusieurs réductions intermédiaires dont une de $\frac{1}{3}$-Sat à Nae-3-Sat et qui est faiblement parcimonieuse de la première forme : le nombre de solutions y est multiplié par une constante (en fait doublé). En fait, cette dernière est optimale et devrait être considérée comme parcimonieuse puisque les solutions de Nae-3-Sat sont stables par inversion : La faible parcimonie de cette réduction n'est qu'un artefact de la convention de comptage des solutions de Nae-3-Sat. En comptant naturellement une solution et son inverse comme une et une seule solution, la réduction ne double plus le nombre de solutions.

Dans la littérature, la #P-complétude de #Plan-3-Col a d'abord été établie par Hunt et al. [55] via une réduction faiblement parcimonieuse de 3-Col de la troisième forme présentée ci-dessus : le nombre de solutions y est multiplié par un nombre aussi grand que l'exponentielle d'un carré du nombre de sommets du graphe initial. Comme on le verra dans le chapitre 3, cette réduction utilise une modification tardive du crossover classique pour Plan-3-Col. Ce crossover, à l'origine non-parcimonieux, permet de réduire 3-Col à Plan-3-Col, et ainsi de prouver la NP-complétude de Plan-3-Col. Une fois modifié, ce crossover devient faiblement parcimonieux (en dupliquant encore plus les solutions mais de façon contrôlée) et permet de prouver la #P-complétude de #Plan-3-Col.

Il reste que toutes les preuves de NP-complétude (pour 3-Col et Plan-3-Col) et de #P-complétude (pour #3-Col et #Plan-3-Col) présentées dans la littérature sont bien hétérogènes. De plus, aucune d'entre elles ne donne de résultats de DP-complétude pour Unique-3-Col et Unique-Plan-3-Col. Car si on applique à 3-Col la même idée de comptage des solutions que pour Nae-3-Sat, i.e., compter pour une et une seule solution une 3-coloration et ses colorations symétriques par permutation de couleurs, alors le problème de l'unicité d'une 3-coloration a du sens. Le chapitre 3 unifiera toutes ces preuves de NP-complétude et de #P-complétude autour de la 3-colorabilité par réduction parcimonieuse et planaire de Sat (en fait $\frac{1}{3}$-Sat) à 3-Col et prouvera par là même la DP-complétude de Unique-3-Col et Unique-Plan-3-Col. Les outils que nous construirons permettront également d'obtenir un crossover parcimonieux pour planariser quadratiquement et parcimonieusement 3-Col.

De façon moins significative, nous obtiendrons un résultat de DP-complétude pour Unique-Plan-Vertex-Cover par réduction parcimonieuse de Plan-$\frac{1}{3}$-Sat à Plan-Vertex-Cover. Ceci améliore un résultat de #P-complétude pour #Plan-Vertex-Cover obtenu par Hunt et al. par réduction faiblement parcimonieuse avec multiplication des solutions par un facteur exponentiel.

Mais nos réductions de $\frac{1}{3}$-Sat à 3-Col et de $\frac{1}{3}$-Sat à Vertex-Cover ne préservent pas seulement la planarité et le nombre de solutions : chacune produit également une instance de taille linéaire en la taille de l'instance d'entrée. On aimerait dire que la réduction n'est pas seulement polynomiale mais linéaire en la taille de la donnée et qu'elle préserve la complexité des problèmes. C'est bien le cas si l'on utilise pour nos réductions le modèle de calcul des machines RAM, un modèle bien adapté pour le temps linéaire. Nous donnons ici une définition succincte de cette machine. Nous n'aurons pas à rentrer dans les détails de ce modèle par la suite : il suffit de savoir que ce modèle permet de capturer la notion intuitive du temps linéaire, et qu'il est très robuste, c'est-à-dire largement indépendant des détails de sa définition.

Le Modèle RAM (Random Access Machine) : La RAM est un modèle de calcul où l'unité de mémoire est le registre, dans lequel on peut stocker une valeur entière. Les registres sont organisés séquentiellement dans la mémoire (que l'on peut voir comme un tableau de cases

$R[\cdot]$) et chacun a une adresse (l'indice de sa case dans le tableau $R[\cdot]$). Un programme tournant sur RAM peut lire ou écrire une valeur dans un registre à partir de son adresse en temps constant. Nous nous plaçons ici dans le modèle défini par E. Grandjean [46, 79, 48, 47]. Un programme sur RAM est une suite d'instructions, située dans une mémoire séparée $P[\cdot]$. Quatre registres sont particuliers : IP (pointeur d'instruction) qui indique l'adresse de l'instruction courante dans $P[\cdot]$ (ce registre est automatiquement incrémenté après l'exécution de chaque instruction sauf les instructions `if` et `halt`) et les deux registres accumulateurs A et B et un registre spécial N. Les instructions sont les suivantes :

`A= c ;` pour une constante $c \geq 0$	`B= A ;`
`A= A` $*$ `B ;` pour $* \in \{+, -, \times\}$	`R[A]= B ;`
`A= N ;`	`N= A ;`
`A= R [A] ;`	`if (A == B) IP=` c_1 `; else IP=` c_2 `;`
	`halt ;`

Dans le modèle de RAM défini par E. Grandjean pour *le temps linéaire*, le point important est que les registres $R[\cdot]$, N[4], A et B d'une RAM travaillant sur une entrée de taille n (en nombre de registres et non en bits) ne peuvent stocker à tout instant que des valeurs bornées en $O(n)$. En particulier, cette contrainte sur A implique que l'espace adressé est aussi $O(n)$. Si la RAM est non-déterministe, on ajoute l'instruction `guess`, qui devine une valeur entière et la stocke dans le registre A. Notons que cette instruction devine un entier $O(n)$ et non un bit ou une valeur $O(1)$ comme sur les MT non-déterministes.

La classe DLIN : Un programme est dit DLIN s'il tourne sur RAM déterministe en temps linéaire $O(n)$ pour toute entrée de taille n (en nombre de registres). Un problème de décision (resp. une fonction) est dit(e) DLIN s'il existe un programme DLIN le décidant (la calculant).

Nos réductions linéaires sont donc définies comme étant des réductions DLIN. Ces réductions préservent les fonctions de complexité $f(n)$ courantes : par exemple un temps exponentiel $f(n) = c^n$ est transformé en un temps $O(f(O(n))) = O(c^{O(n)}) = O(c^{an+b}) = O(c^b c^{an}) = O((c^a)^n) = O(d^n)$. La base de l'exponentielle change mais le temps reste exponentiel. Cette linéarité des réductions, combinée avec la parcimonie qui préserve en pratique une bijection simple entre les solutions de l'instance d'entrée w_1 et l'instance de sortie w_2 (de telle sorte qu'une solution pour w_1 est DLIN-calculable à partir de la solution correspondante pour w_2), permet également de préserver la complexité de l'énumération des solutions de Π_1 (i.e., le délai entre la génération de deux solutions successives).

Le fait que l'on obtienne des réductions linéaires entre les problèmes NP-complets Sat, 3-Col, $\frac{1}{3}$-Sat, n'est pas un hasard. En fait, ces problèmes n'utilisent qu'une quantité faiblement polynomiale de non-déterminisme puisque linéaire.

La classe NLIN : Un problème de décision appartient à la classe NLIN s'il est décidable en temps linéaire sur RAM non-déterministe, ou de façon équivalente s'il est vérifiable en temps linéaire sur RAM déterministe, avec des témoins de taille linéaire.

Sat, 3-Col et $\frac{1}{3}$-Sat, à l'instar des 21 problèmes NP-complets originaux étudiés par Karp [59], sont en fait NLIN puisque pour les décider, il suffit de deviner un assignement satisfaisant, ou

[4] Le registre spécial N contient au départ la taille n de l'entrée, et cette entrée et stockée dans les n premiers registres $R[0], \cdots, R[n-1]$. On adopte les mêmes conventions pour la sortie s'il y en a.

une coloration, solutions qui sont toutes de taille linéaire en la donnée et vérifiables en temps linéaire. Les solutions sont même de taille moins que linéaire puisqu'on devine n valeurs $O(1)$ (des bits ou des couleurs) alors qu'un non-déterminisme linéaire "plein" pourrait deviner jusqu'à n valeurs bornées chacune en $O(n)$. Ainsi la quantité de non-déterminisme utilisée par SAT, 3-COL et $\frac{1}{3}$-SAT est en fait légèrement sous-linéaire, de l'ordre de $\frac{n}{\log n}$ registres de valeurs $O(n)$. Pour cette raison, ils ont peu de chance d'être NLIN-complets sous réductions DLIN comme le problème RISA (Reduction of Incompletely Specified Automata), qui, lui, utilise pleinement le non-déterminisme linéaire et a été prouvé NLIN-complet par E. Grandjean [42].

Parmi les problèmes NP-complets dans NLIN mais n'utilisant qu'une quantité de non-déterminisme de l'ordre de $\frac{n}{\log n}$ registres, il y a aussi le problème HAMILTON. Si l'on sait réduire SAT à HAMILTON en temps linéaire, on ne sait pas faire l'inverse. La raison est qu'on ne sait pas maintenir une contrainte de connexité globale sur une donnée de taille n avec un système SAT de taille significativement moins large que $\Omega(n \log n)$, dans le cas général. A l'inverse, nous montrerons les équivalences linéaires (et parcimonieuses) des restrictions planaires de ces problèmes, PLAN-HAMILTON et PLAN-SAT (voir chapitres 4 et 5).

De façon moins significative, notons également l'équivalence linéaire (et parcimonieuse) de SAT et VERTEX-COVER. Ceci contredit l'intuition exposée dans [19] selon laquelle VERTEX-COVER n'est probablement par linéairement réductible à SAT, à cause de la contrainte globale de cardinalité, ce qui oblige SAT à simuler des additions en binaires pour vérifier cette condition, et ce faisant à utiliser un système de taille $n \log n$. En fait, une simple stratégie de diviser-pour-régner sur les additions nous permettra d'obtenir un schéma d'additionneur en arbre, de hauteur $\log n$ mais de taille $O(n)$ au total. De façon générale, nous montrons que tous les problèmes comme VERTEX-COVER qui combinent des contraintes locales plus une contrainte globale de cardinalité, tels MAX-SAT, DOMINATING-SET, etc., sont linéairement réductibles à SAT.

2.3 Terminologie pour nos réductions

Dans les sections suivantes, nous montrons comment $\frac{1}{3}$-SAT peut se révéler plus pratique que SAT pour établir des équivalences à SAT sous réductions linéaires, parcimonieuses et planaires. Nous profiterons des réductions qui y sont présentées pour illustrer la terminologie que nous introduisons ci-après : *gadgets, gadgets planaires, sommets distingués, états locaux, configurations, gadgets parcimonieux, arêtes distinguées et pendantes, arêtes grasses et maigres.*

Gadgets : Lorsqu'on veut réduire un problème Π_1 à un problème de graphe Π_2, une approche commune et modulaire est de créer des petits graphes dont la fonction est de simuler les différents objets élémentaires intervenant dans Π_1 : par exemple, si Π_1 est un problème de Satisfaisabilité, on cherchera des graphes simulant une clause, une variable, un littéral, ou encore un ensemble d'occurrences d'une même variable. Ces graphes sont appelés à être connectés ensemble pour produire un codage de l'instance d'entrée. Nous les appelons *gadgets*.

Etats locaux : Ce qui nous intéresse dans un gadget est l'ensemble des colorations légales dont il peut faire l'objet pour le problème Π_2 lorsqu'il est intégré comme sous-graphe propre du graphe constituant l'instance de sortie. Nous aurons deux types de gadgets dans cette thèse : les gadgets dont le coloriage porte sur les sommets, utilisés par exemple pour réduire un problème Π_1 à 3-COL, $\frac{1}{3}$-SAT, VERTEX-COVER, etc., et les gadgets dont le coloriage porte sur les arêtes, utilisés pour réduire Π_1 à toutes les variantes de HAMILTON. L'ensemble des colorations légales d'un

gadget G lorsque celui-ci est imaginé comme sous-graphe strict d'un autre graphe indéterminé est appelé *ensemble des états locaux*.

Sommets distingués et configurations : Prenons le cas d'un gadget dont le coloriage porte sur les sommets. Parmi ses sommets, certains constituent la "grille de lecture" de l'objet simulé par le gadget. Par exemple, si l'on simule une 3-clause $\ell_1 \vee \ell_2 \vee \ell_3$, on souhaitera que le gadget contienne trois sommets particuliers $v(\ell_1)$, $v(\ell_2)$, $v(\ell_3)$ reproduisant l'état des occurrences de cette clause, et on cherchera à établir une correspondance entre toutes les possibilités de satisfaire la clause et les possibilités de colorier ces sommets. Pour cette raison, nous les appellerons *sommets distingués*. Un état local (i.e., un coloriage légal du gadget entier) restreint aux sommets distingués sera appelé *configuration*.

Gadget Hamiltonien : Pour les réductions vers des problèmes de coloriages d'arêtes comme HAMILTON, nous n'avons plus de sommets distingués mais des *arêtes pendantes* et des *arêtes distinguées*. Les *arêtes pendantes* sont des *arêtes distinguées* auxquelles il manque le deuxième sommet. Ce sont les arêtes par lesquelles le gadget doit nécessairement être connecté au reste du graphe qui fournit le deuxième sommet de chaque arête pendante, et qui constitue l'instance complète que l'on cherche à construire. Notons que dans le cas de HAMILTON, un *état local* est un 2-coloriage d'arêtes (*grasses* ou *maigres*) tel que l'ensemble des arêtes grasses forme un nombre arbitraire de chemins qui relient chacun deux arêtes pendantes. Ces chemins sont disjoints deux à deux, et l'ensemble de ces chemins couvre tous les sommets du gadget.

Gadgets parcimonieux : Comme on souhaite toujours obtenir des réductions parcimonieuses, il est nécessaire que toute configuration détermine de façon unique la coloration du reste des sommets du gadget. Autrement dit, on exige une bijection entre les états locaux et les configurations. Un gadget ayant une telle propriété sera appelé *gadget parcimonieux*. Il arrive cependant que les choses prennent un tour un peu moins favorable : à chaque configuration correspond un nombre $c > 1$ d'états locaux, mais avec c identique pour toutes les configurations du gadget. Ce type de gadget est source de faible parcimonie pour la réduction construite avec au final multiplication des solutions par un facteur exponentiel (les états locaux pouvant varier indépendamment sur tous les gadgets en laissant les configurations inchangées). On utilisera le qualificatif *faiblement parcimonieux* pour ce genre de gadget. Enfin, si à chaque configuration correspond un nombre variable d'états locaux, le gadget est dit *non parcimonieux*. Ce type de gadget est source de non-parcimonie pour la réduction construite, chaque solution simulée étant dupliquée un nombre de fois dépendant de sa nature propre. Et comme on ne connaît pas la nature des solutions simulées, la connaissance du nombre de solutions de l'instance de départ ne permet pas d'établir le nombre de solutions dans l'instance d'arrivée.

Gadgets planaires : Lorsqu'on veut construire des réductions préservant la planarité des instances, il va de soit que les graphes constituant les gadgets doivent être planaires. Cela ne suffit cependant pas. Comme les sommets distingués sont les seuls à pouvoir être connectés au reste du graphe, tout sommet distingué doit se trouver sur la face externe du gadget. En ce qui concerne les gadgets Hamiltoniens, il en va de même pour les arêtes pendantes et plus généralement pour les arêtes distinguées (sur lesquelles peuvent éventuellement se greffer d'autres sommets par la suite). De plus, l'ordre des sommets distingués, des arêtes pendantes et des arêtes distinguées le long de la frontière de la face externe du gadget pour un ordre fixé (e.g., trigonométrique) doit impérativement être respecté pour sa connexion planaire au reste du graphe.

Crossovers : Dans le cadre d'une réduction à un problème planaire PLAN-Π_2, il est fréquent qu'il soit pratique de laisser certaines arêtes se croiser dans un premier temps. Ces croisements doivent ensuite être éliminés pour restaurer la planarité de l'instance construite. Ceci est généralement mis en oeuvre par la pose d'un dispositif, appelé *crossover*, au point de croisement. Un crossover est un gadget planaire qui, tout en éliminant un croisement d'arêtes dans un graphe G, laisse G invariant pour Π_2.

Lorsque Π_2 est un problème de coloriage de sommets, un crossover est un gadget à quatre sommets distingués x, y', x', y plongés dans cet ordre le long de sa face externe pour un sens fixé (e.g., trigonométrique), tel que l'ensemble des configurations $(C(x), C(y), C(x'), C(y'))$ soit tous les quadruplets (c_1, c_2, c_1, c_2) possibles pour tout couple de couleurs c_1,c_2 du problème Π_2. Intuitivement, x et y sont respectivement dupliqués en x' et y', et la résolution d'un croisement d'arête se fait alors selon le schéma de la Fig. 2.1. Pour les problèmes de coloriage d'arêtes comme HAMILTON, des schémas de crossovers analogues seront également utiles.

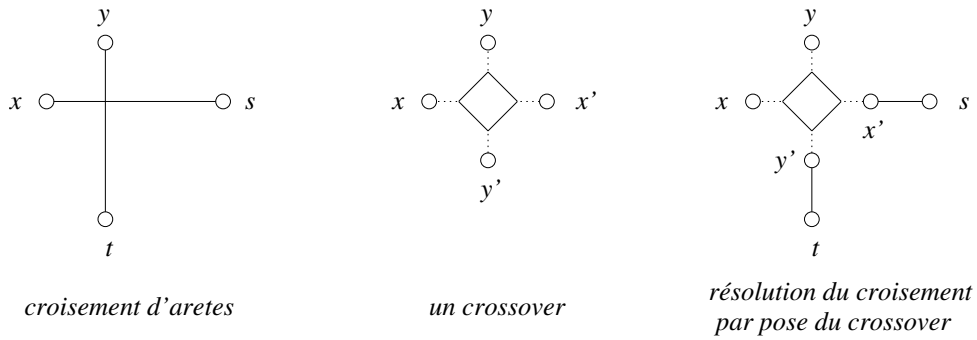


FIG. 2.1 – Schéma d'utilisation d'un crossover dans un problème de coloriage de sommets

Construire des crossovers parcimonieux directement est parfois difficile pour certains problèmes, notamment PLAN-3-COL. Lorsque l'on sait à l'avance que certains quadruplets (c_1, c_2, c_1, c_2) ne sont pas susceptibles de se produire dans un contexte particulier, on peut prendre avantage de cette connaissance et construire des *crossovers restreints*, qui ne fonctionnent que pour certaines restrictions sur les quadruplets. Nous verrons que combinés ensemble, plusieurs *crossovers restreints* parcimonieux permettent d'obtenir un crossover parcimonieux pour PLAN-3-COL.

Nos gadgets et leurs conventions : Le lecteur remarquera que parmi toutes nos figures (une centaine), celles décrivant un gadget et son fonctionnement, nous avons adopté les mêmes conventions (voir par exemple la section 2.4) : Sur la partie gauche de la figure, le graphe d'un gadget est dessiné une première fois entièrement, ce que nous appelons son *implémentation* et une convention graphique de *représentation* est introduite à côté.

Cette représentation est constituée d'un symbole graphique autour duquel gravitent les sommets distingués (ou les arêtes pendantes et distinguées) du gadget, et ce, dans le même ordre que son implémentation pour un sens fixé (e.g., trigonométrique) autour de la face externe du gadget.

Sur le côté droit de la figure, l'implémentation est ensuite reprise et coloriée pour exposer tous les états locaux possibles du gadget. Les configurations correspondantes utilisent le symbole de représentation.

Enfin, lorsqu'un gadget G est implémenté en termes d'autres sous-gadgets, ce sont les représentations de ces sous-gadgets qui sont utilisées dans les schémas d'implémentation de G.

2.4 Un problème bien pratique : $\frac{1}{3}$-SAT

Lorsque l'on veut réduire SAT ou PLAN-SAT à un problème choisi (par exemple 3-COL ou VERTEX-COVER, ou leurs restrictions au plan PLAN-3-COL ou PLAN-VERTEX-COVER) sur lequel on a peu d'intuition constructive quant à son expressivité fine (autrement dit, on ne voit pas comment l'exploiter pour créer des gadgets logiques ou l'on ne voit pas comment le faire de façon planaire ou parcimonieuse), et que l'on veut montrer qu'il est linéairement et parcimonieusement équivalent à SAT (ou PLAN-SAT), il est souvent malaisé d'utiliser directement SAT dans les réductions car ce problème est trop riche et hétérogène syntaxiquement : les clauses peuvent être de longueur quelconque, les littéraux y apparaissant peuvent être positifs ou négatifs, et les clauses peuvent être satisfaites de trop de façons différentes. Même pour 3-SAT dont les clauses sont toutes de longueur 3, il faut être capable de construire des gadgets admettant des configurations aussi variées que :

- $(0,0,1)$, $(0,1,0)$, et $(1,0,0)$, autrement dit, un unique littéral témoigne de la satisfaction de la 3-clause.
- $(0,1,1)$, $(1,0,1)$, et $(1,1,0)$: exactement deux littéraux témoignent de la satisfaction de la 3-clause.
- $(1,1,1)$: les trois littéraux témoignent de la satisfaction de la 3-clause.

Il est plus judicieux d'utiliser un problème linéairement et parcimonieusement équivalent à SAT (ou PLAN-SAT), et qui manipule des objets plus simples que des clauses sur des littéraux signés : le problème $\frac{1}{3}$-SAT (abréviation de MONOTONE-ONE-IN-THREE-SAT [76, 33]) introduit par Schaefer [76] avec sa restriction planaire PLAN-$\frac{1}{3}$-SAT est à ce titre bien adapté : Comme 3-SAT, $\frac{1}{3}$-SAT est le problème de la satisfaction d'une formule propositionnelle en CNF où les clauses sont toutes de longueur 3, mais à la différence de 3-SAT :

- les littéraux y sont tous *positifs*, et
- un assignement des variables satisfait la formule ssi *exactement un littéral* est valué à vrai dans chaque clause (que l'on appellera $\frac{1}{3}$-clause pour cette raison).

Simuler une $\frac{1}{3}$-clause ne nécessite donc aucune gestion de signe et il suffit de trouver un gadget admettant les configurations symétriques $(0,0,1)$, $(0,1,0)$ et $(1,0,0)$.

Il est assez facile de voir que $\frac{1}{3}$-SAT (resp. PLAN-$\frac{1}{3}$-SAT) est linéairement et parcimonieusement équivalent à SAT (resp. PLAN-SAT). En effet, il est d'une part immédiat de voir que $\frac{1}{3}$-SAT se réduit linéairement et parcimonieusement à SAT puisque toute $\frac{1}{3}$-clause (x,y,z) est parcimonieusement simulable par la conjonction des quatre clauses suivantes :

$$\begin{cases} C_1 : & \neg x \vee \neg y \\ C_2 : & \neg y \vee \neg z \\ C_3 : & \neg z \vee \neg x \\ C_4 : & x \vee y \vee z \end{cases}$$

Comme le gadget constitué par ces quatre clauses est planaire (voir Fig. 2.2), PLAN-$\frac{1}{3}$-SAT se réduit également linéairement et parcimonieusement à PLAN-SAT.

Réciproquement, SAT (resp. PLAN-SAT) se réduit linéairement et parcimonieusement à $\frac{1}{3}$-SAT (resp. PLAN-$\frac{1}{3}$-SAT). Pour le montrer, nous construisons trois gadgets logiques : le premier (NOT) simulant l'opérateur unaire de négation logique $NOT(x) = \neg x$, le deuxième (CONST) simulant les constantes vraie et fausse, et le troisième (EQV-NOR) simulant les deux opérateurs binaires EQV et NOR définis par $\mathrm{EQV}(x,y) = (x \iff y)$ et $\mathrm{NOR}(x,y) = \neg(x \vee y)$. La composition du NOR et du NOT nous donne naturellement l'opérateur OR défini par $\mathrm{OR}(x,y) = (x \vee y)$, et le montage en série de ce dernier opérateur nous permet d'évaluer une clause par accumulation

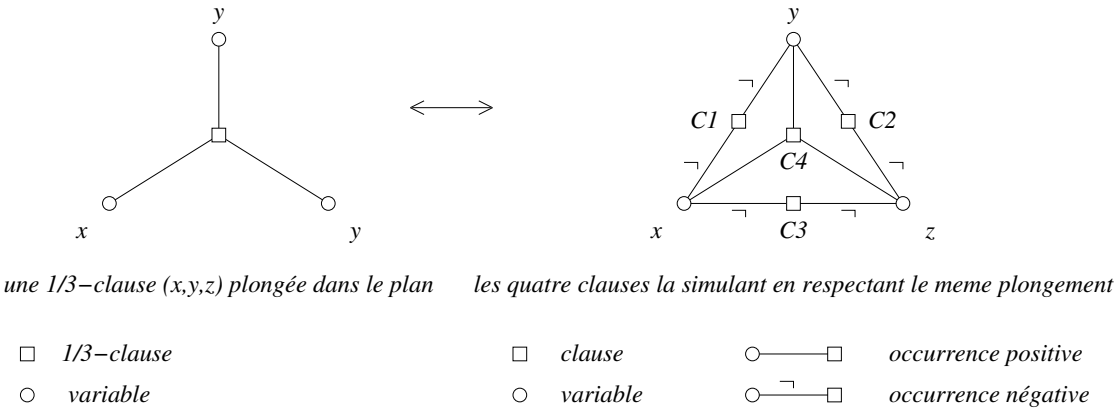


FIG. 2.2 – Réduction de PLAN-$\frac{1}{3}$-SAT à PLAN-SAT

successive des valeurs de vérité le long de la clause. Enfin, la connexion d'un gadget CONST à l'accumulateur final permet de forcer l'évaluation de la clause à vrai.

Le gadget NOT simulant $\mathrm{NOT}(x) = \neg x$ est constitué des trois $\frac{1}{3}$-clauses suivantes :

$$\begin{cases} (i,j,k) \\ (x,i,n) \\ (x,k,n) \end{cases}$$

où i, j et k sont de nouvelles variables propres à chaque copie du gadget. Sa correction et sa parcimonie découlent du raisonnement suivant : Supposons que j soit fausse. Alors exactement l'une des variables i ou k est vraie par la $\frac{1}{3}$-clause (i,j,k), et l'autre est fausse. L'une des deux $\frac{1}{3}$-clauses (x,i,n) ou (x,k,n) force alors x et n à être toutes deux fausses, alors que l'autre $\frac{1}{3}$-clause force exactement l'une des deux variables x ou n à être vraie, une contradiction. Donc j est nécessairement vraie, et i et k sont fausses par la $\frac{1}{3}$-clause (i,j,k). Les deux $\frac{1}{3}$-clauses (x,i,n) et (x,k,n) forcent alors à ce qu'exactement l'une des variables x ou n soit vraie. Il y a deux états locaux (x,n,i,j,k) possibles, $(true, false, false, true, false)$ et $(false, true, false, true, false)$ correspondant aux deux configurations (x,n) possibles, soit $(true, false)$ soit $(false, true)$: le gadget se comporte bien comme un inverseur parcimonieux. On voit par ailleurs sur la Fig. 2.3 que le gadget est planaire.

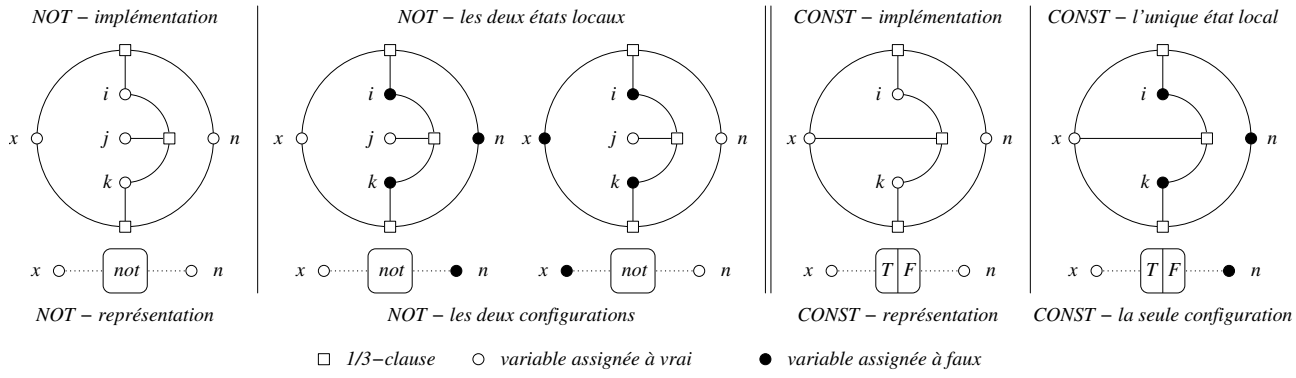


FIG. 2.3 – Les gadgets d'inversion et de constantes dans PLAN-$\frac{1}{3}$-SAT

Le gadget CONST est simplement obtenu en fusionnant les variables j et x du gadget NOT : j étant toujours vrai et x et n toujours opposées, x et n sont alors respectivement les

constantes vraie et fausse. Le gadget est parcimonieux, n'ayant qu'un seul état local $(x, n, i, k) = (true, false, false, false)$ pour une seule configuration $(x, n) = (true, false)$. Le gadget reste planaire après la fusion des sommets comme on peut le vérifier sur la Fig. 2.3.

Le gadget planaire EQV-NOR, représenté sur la Fig. 2.4 et réalisant $e = \text{EQV}(x, y)$, i.e. $e = (x \iff y)$, et $n = \text{NOR}(x, y)$, est constitué des trois $\frac{1}{3}$-clauses suivantes :

$$\begin{cases} (i, e, j) \\ (x, i, n) \\ (n, j, y) \end{cases}$$

où i et j sont des variables propres à chaque copie du gadget. Sa correction et sa parcimonie découlent du raisonnement suivant : Les trois $\frac{1}{3}$-clauses impliquent qu'au moins deux des trois variables i, j et n sont fausses. Si elles sont fausses toutes les trois, alors on a immédiatement x, y, et e qui sont toutes vraies, un état local illustré à droite de la Fig. 2.4. Sinon, exactement deux variables parmi i, j et n sont fausses : le gadget étant symétrique pour i, j et n, on peut supposer que i et j sont fausses et que n est vraie. On a alors nécessairement e vrai, et x et y faux, état local illustré à gauche de la Fig. 2.4, suivi de ses deux états locaux symétriques. On a donc en tout quatre états locaux pour EQV-NOR, résumés par les quatre lignes de la table de vérité suivante :

x	y	$i = \text{NRIMP}(x, y)$	$j = \text{NIMP}(x, y)$	$e = \text{EQV}(x, y)$	$n = \text{NOR}(x, y)$
$false$	$false$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$false$	$false$
$true$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$false$	$true$	$false$

On voit que les quatre combinaisons booléennes sont possibles pour x et y, et on peut donc interpréter les autres variables comme des fonctions de x et y. On constate en particulier sur la table de vérité que e et n portent respectivement le résultat des opérateurs EQV et NOR : $e = (x \iff y)$, et $n = \neg(x \vee y)$. Les autres opérateurs portés par i et j, resp. NRIMP et NIMP, sont moins intéressantes : $i = \neg(x \Longleftarrow y)$ et $j = \neg(x \Longrightarrow y)$. Là encore, le gadget est planaire et il y a bien quatre états locaux pour quatre configurations, i.e., le gadget est parcimonieux.

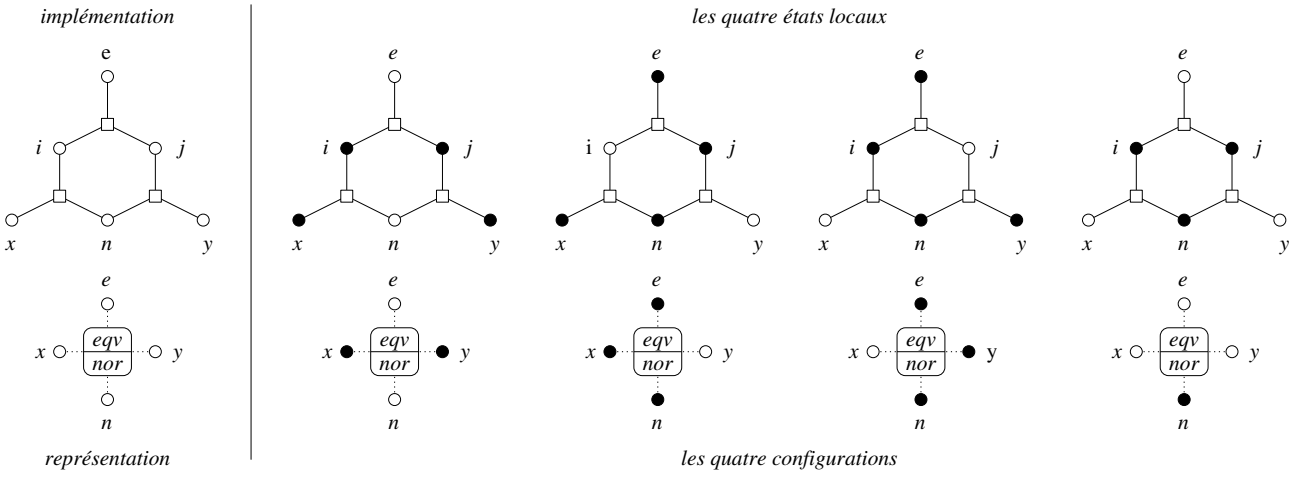


FIG. 2.4 – Le gadget EQV-NOR dans PLAN-$\frac{1}{3}$-SAT

Une clause $\ell_1 \vee \cdots \vee \ell_k$ construite sur les variables $x_1, \cdots, x_k$ (où x_j est la variable associée au littéral ℓ_j) est maintenant simulée par un gadget construit (entre autres) sur les sommets x_j, ℓ_j pour $1 \leq j \leq k$, les sommets n_j pour $1 < j \leq k$, et les sommets a_j pour $1 \leq j < k$:

- Pour tout $1 \leq j \leq k$, on pose $\ell_j = x_j$ si ℓ_j est une occurrence positive de x_j. Sinon, on connecte ℓ_j et x_j par un gadget NOT.
- On pose $\ell_1 = a_1$, et pour tout $1 < j \leq k$, on connecte a_{j-1} et ℓ_j en tant que portes opérandes d'un gadget EQV-NOR dont la porte NOR est n_j. Les sommets n_j sont connectés à a_j par un gadget NOT, à l'exception de n_k, forcé à porter la valeur $false$ comme porte n d'un gadget CONST.

On en tire les équivalences $a_i \iff \neg n_i$ pour tout $1 < i < k$, et $n_j \iff \neg(\ell_1 \vee \cdots \vee \ell_j)$ pour tout $1 < j \leq k$; n_k étant nécessairement fausse, $\ell_1 \vee \cdots \vee \ell_k$ doit être vraie.

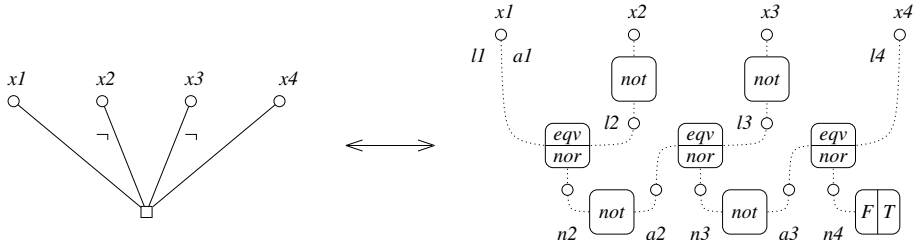


FIG. 2.5 – Le gadget simulant la clause $x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4$ dans PLAN-$\frac{1}{3}$-SAT

Une telle simulation de toutes les clauses de l'instance SAT conclut la réduction de SAT à $\frac{1}{3}$-SAT. La linéarité est immédiate, et la parcimonie de la construction découle de celle des gadgets CONST, NOT et EQV-NOR. De plus, la Fig. 2.5 montre que notre construction préserve la planarité des instances. Nous avons donc aussi construit une réduction linéaire et parcimonieuse de PLAN-SAT à PLAN-$\frac{1}{3}$-SAT.

Enfin, le gadget EQV-NOR est très pratique pour construire un crossover parcimonieux pour PLAN-$\frac{1}{3}$-SAT. Le but de ce crossover est d'obtenir un système planaire recopiant les valeurs portées par deux sommets distingués x et y dans les sommets distingués respectifs x' et y' avec pour ordre anti-trigonométrique x, y, x', y', i.e., x et x' (resp. y et y') sont plongés sur les coins opposés d'un carré imaginaire. Il suffit de connecter quatre exemplaires du gadget EQV-NOR comme sur la Fig. 2.6 pour obtenir un tel crossover. En effet, si le sommet central (qui est connecté à la porte EQV des quatre gadgets EQV-NOR) est à vrai, alors les quatre sommets x, y, x', y' doivent être équivalents : ils sont soit tous faux soit tous vrais. On obtient alors les deux configurations monochromes de gauche sur la Fig. 2.6. En revanche si le sommet central est mis à faux, alors les sommets opérandes de chaque gadget EQV-NOR doivent porter des valeurs opposées, et on a alors nécessairement un bicoloriage alternant sur le cycle (x, y, x', y'), ce qui donne les deux configurations de droite sur la Fig. 2.6 : soit x et x' sont à vrai avec y et y' à faux, soit on a l'inverse, i.e., x et x' sont à faux avec y et y' à vrai.

La simplicité de notre crossover parcimonieux pour PLAN-$\frac{1}{3}$-SAT se compare favorablement au crossover parcimonieux original de Lichtenstein [62] dans PLAN-SAT. Ce dernier repose sur les contraintes suivantes :

$$\bigwedge \left\{ \begin{array}{rcl} (x \wedge y') & \iff & \alpha \\ (x' \wedge \neg y) & \iff & \beta \\ (\neg x \wedge \neg y) & \iff & \gamma \\ (\neg x \wedge y') & \iff & \delta \end{array} \right. \bigwedge \left\{ \begin{array}{c} (\alpha \vee \beta \vee \gamma \vee \delta) \\ (\neg\alpha \vee \neg\beta) \\ (\neg\beta \vee \neg\gamma) \\ (\neg\gamma \vee \neg\delta) \\ (\neg\delta \vee \neg\alpha) \end{array} \right.$$

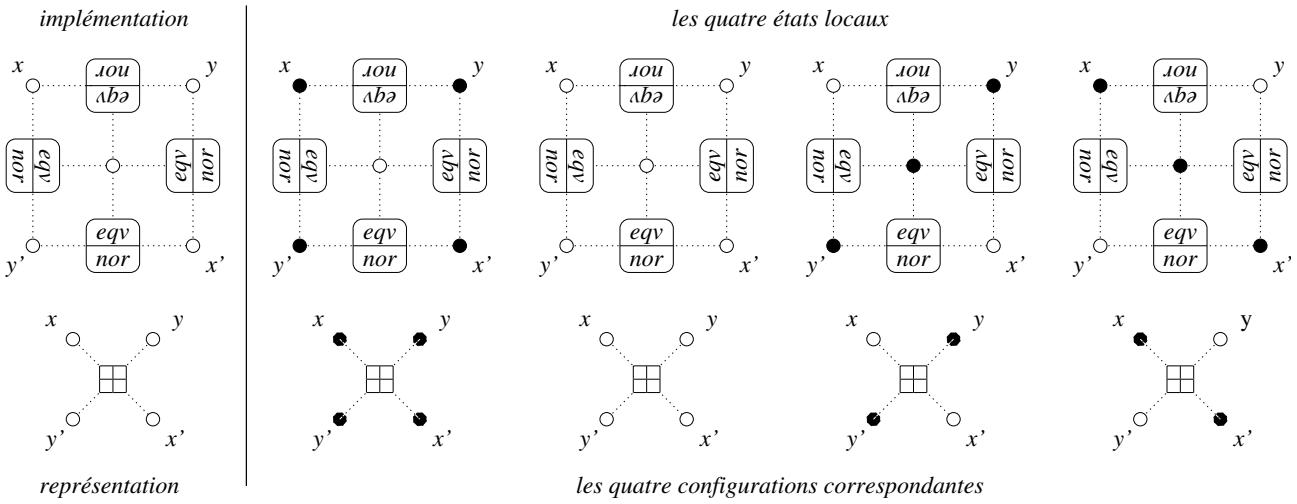


FIG. 2.6 – Notre crossover parcimonieux dans PLAN-$\frac{1}{3}$-SAT

Nous laissons au lecteur le soin de vérifier qu'il n'y a que quatre solutions à ce système, avec les valeurs escomptées pour un crossover sur x, y, x' et y' et que la Fig. 2.7 représente bien un plongement planaire du développement clausal de ce système.

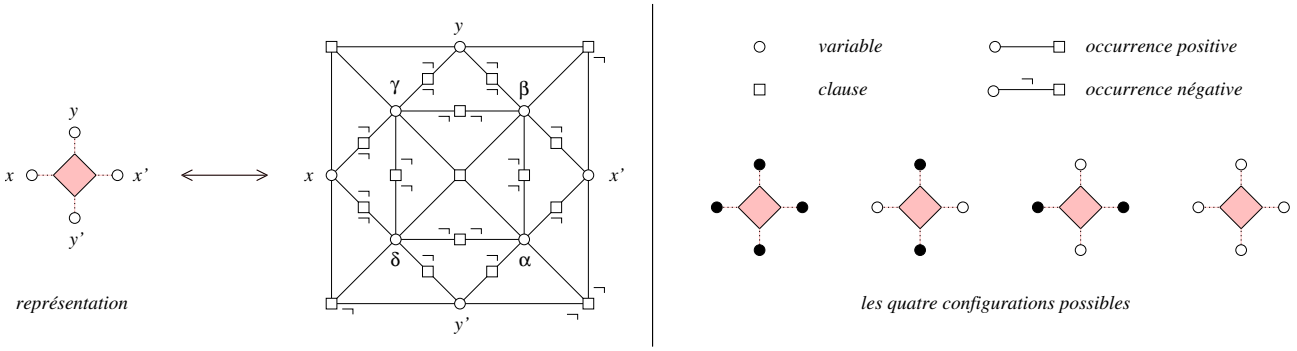


FIG. 2.7 – Le crossover parcimonieux de Lichtenstein dans PLAN-SAT

Ces crossovers permettent d'obtenir aisément des planarisations quadratiques de $\frac{1}{3}$-SAT et de SAT suivant le schéma commun de la Fig. 2.8. Soit $\varphi(V, L)$ une formule non planaire à planariser, et m la taille de φ, i.e. la somme des longueurs de ses clauses. On créé une grille virtuelle $m \times m$ où on réserve un bloc de x colonnes contiguës pour chaque variable ayant x occurrences, ainsi qu'un bloc de y lignes contiguës pour chaque clause de longueur y. Intuitivement, les m lignes correspondent aux m occurrences de variables groupées par clause, et les m colonnes représentent ces mêmes occurrences groupées par variable. Pour chaque occurrence d'une variable $v \in V$ dans une clause $c \in L$, on choisit une colonne libre j parmi celles allouées à v et une ligne libre i allouée à v, et on trace le chemin virtuel formé du segment horizontal $((i, 0)(i, j))$ et du segment vertical $((i, j), (0, j))$. Les sommets de variables et de clauses sont plongés dans leurs blocs respectifs. Pour chaque case de la grille qui contient un croisement de deux chemins, on place un crossover parcimonieux dans cette case. Les sommets distingués successifs de crossovers voisins le long d'un chemin sont fusionnés. Les extrémités des chemins correspondant à un même bloc sont fusionnés avec le sommet (de clause ou de variable) de ce bloc.

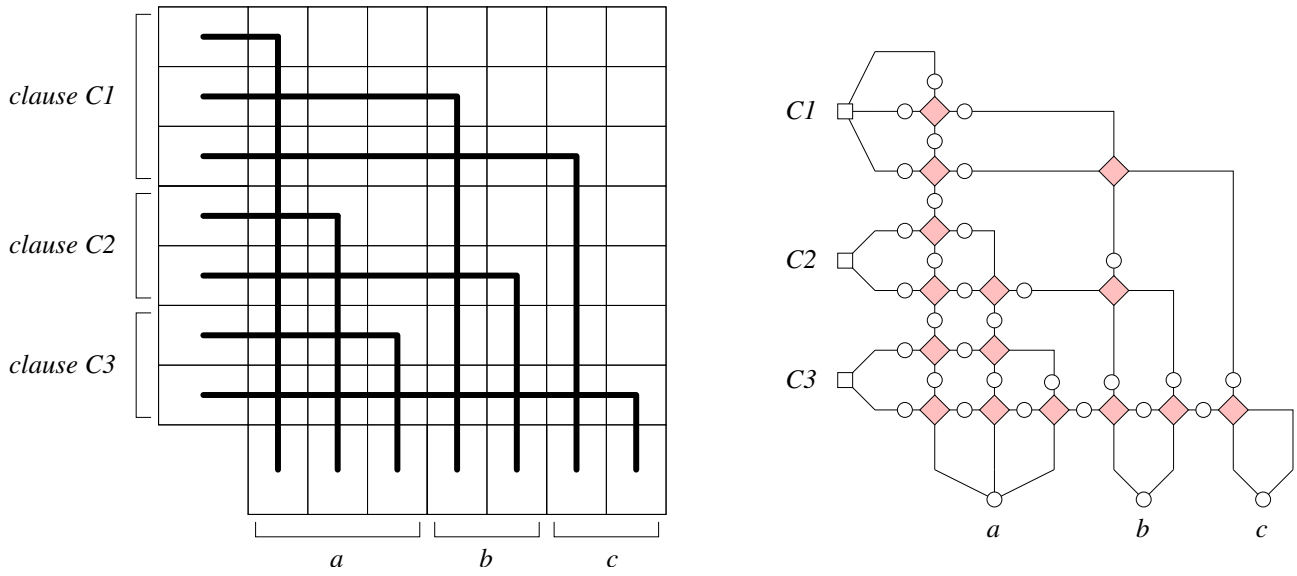


FIG. 2.8 – Schéma de planarisation pour SAT et $\frac{1}{3}$-SAT

Pour terminer, il est très facile de réduire parcimonieusement 3-COL à $\frac{1}{3}$-SAT. En effet, soit $G(V,E)$ le graphe de l'instance 3-COL à réduire. Il suffit d'associer à chaque sommet $v \in V$, les trois variables W_v, G_v, B_v, chacune témoignant du fait que v est colorié resp. en *white*, *gray* ou *black*. La $\frac{1}{3}$-clause (W_v, G_v, B_v) code alors naturellement le fait que le sommet v est colorié avec exactement une couleur parmi les trois. Chaque arête $e = (x,y) \in E$ est alors codée par les trois $\frac{1}{3}$-clauses suivantes :

$$\begin{cases} (W_x, & W_{x,y}, & W_y) \\ (G_x, & G_{x,y}, & G_y) \\ (B_x, & B_{x,y}, & B_y) \end{cases}$$

où $W_{x,y}$, $G_{x,y}$ et $B_{x,y}$ sont de nouvelles variables dont les valeurs de vérité signifient "ni x ni y ne sont coloriés resp. en *white*, *gray*, *black*". Voir Fig. 2.9. La correction est triviale. Si l'on souhaite filtrer les colorations isomorphes par permutation de couleurs, on peut simplement choisir deux sommets arbitraires b et g adjacacents dans G et connecter les sommets B_b et G_g à deux gadgets CONST fixant ces variables à vrai, ce qui revient à coder que b et g sont resp. coloriés en *black* et *gray*.

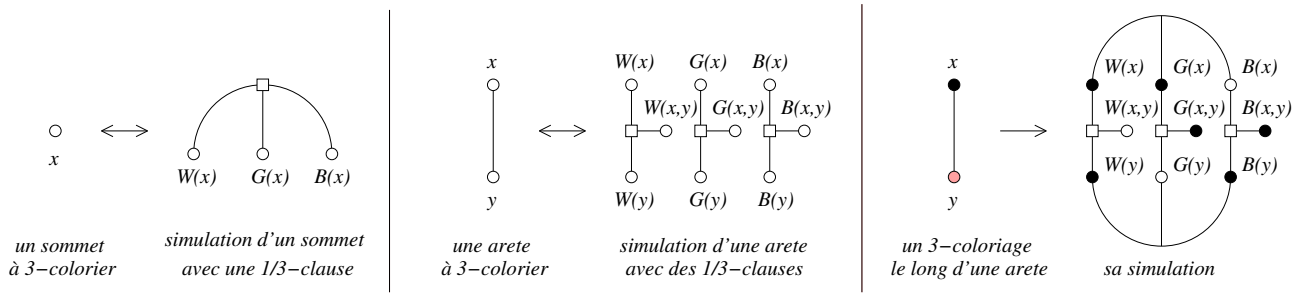


FIG. 2.9 – Réduction non planaire de 3-COL à $\frac{1}{3}$-SAT

Pour réduire (linéairement et parcimonieusement) PLAN-3-COL à PLAN-$\frac{1}{3}$-SAT, les choses se compliquent car la réduction linéaire et parcimonieuse ci-dessus ne préserve pas la planarité

de G, comme à chaque fois qu'on répartit de l'information concentrée en un seul sommet sur plusieurs. Deux problèmes se posent : les croisements au niveau du codage d'un sommet $v \in V$, et les croisements au niveau du codage d'une arête.

En ce qui concerne le premier problème, la technique standard est de dupliquer d fois les variables W_x, G_x, B_x pour chaque sommet $x \in G$ de degré d. Sur chaque "slot" de trois variables dupliquées se greffera le codage d'une arête. Là encore, le crossover parcimonieux pour PLAN-$\frac{1}{3}$-SAT nous permet de réaliser cela linéairement, en utilisant $O(d)$ crossovers, comme montré sur la Fig. 2.10.

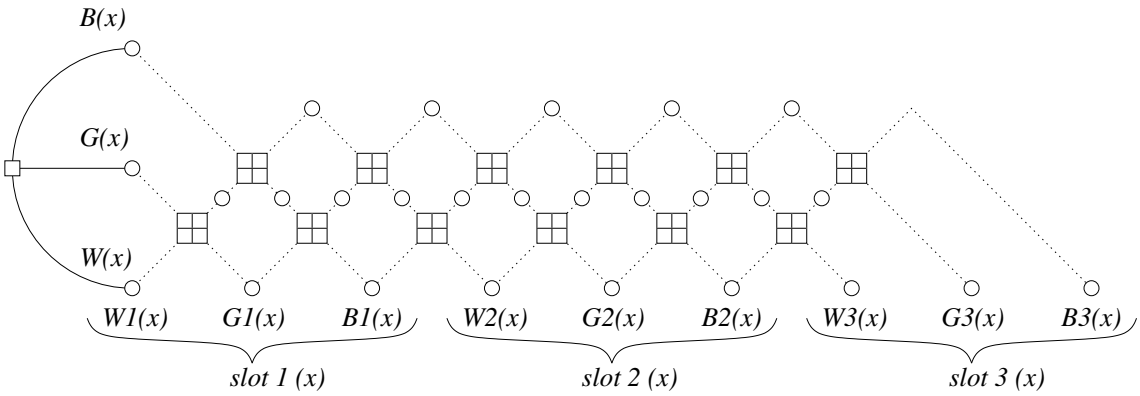


FIG. 2.10 – Duplication du codage d'un sommet selon son degré

Le deuxième problème vient du fait que les variables W_x, G_x, B_x de tout slot d'une même variable x sont ainsi ordonnés dans un sens fixé (e.g.,trigonométrique) autour du gadget simulant x. Pour le codage d'une arête (x, y), il en découle que les variables des slots correspondants ne vont pas être face à face (en "miroir"). L'utilisation de trois crossovers, comme indiqué sur la Fig. 2.11, permet de rétablir la situation.

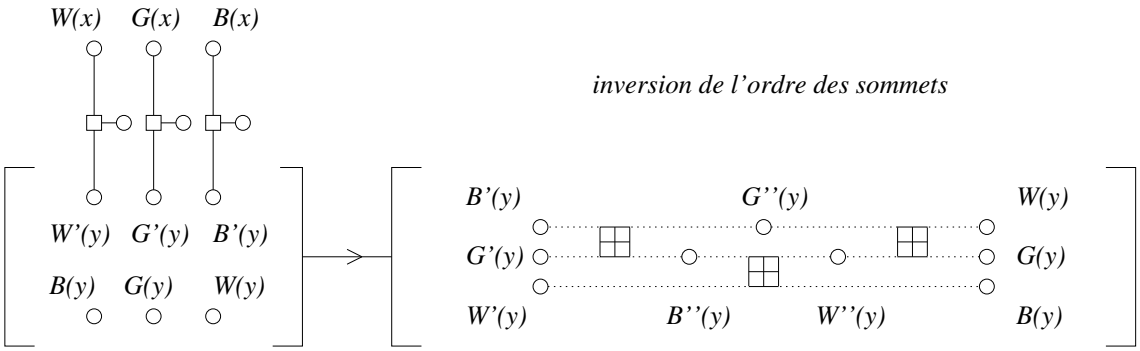


FIG. 2.11 – Inversion de l'ordre trigonométrique d'un slot

Ceci nous amène naturellement au chapitre 3, où l'on établira les réductions parcimonieuses réciproques, i.e., de $\frac{1}{3}$-SAT à 3-COL, et de PLAN-$\frac{1}{3}$-SAT à PLAN-3-COL.

3

Réductions parcimonieuses

ou la préservation de la structure des solutions

Sommaire

3.1 Introduction 28
3.2 De la 3-colorabilité à la 3-colorabilité planaire 31
3.2.1 La réduction générique : le principe de la planarisation 31
Le crossover non-parcimonieux classique 32
Le crossover faiblement parcimonieux de Hunt et al. 33
3.2.2 Vers un crossover parcimonieux et radiant 35
De l'intérêt d'être radiant pour être parcimonieux 35
Une analogie optique 37
3.2.3 L'implémentation parcimonieuse des convertisseurs 41
Un convertisseur bidirectionnel planaire non-parcimonieux 41
Un convertisseur bidirectionnel parcimonieux non-planaire 43
Un convertisseur bidirectionnel parcimonieux et planaire 45
Un convertisseur unidirectionnel parcimonieux et radiant 46
3.2.4 L'implémentation des gadgets radiants via les convertisseurs 47
3.2.5 La planarisation parcimonieuse de 3-Col 51
3.3 De la satisfaisabilité à la 3-colorabilité 52
3.3.1 Réductions non ou faiblement parcimonieuses de Sat à 3-Col 52
La réduction non-parcimonieuse de Kozen 52
La réduction faiblement parcimonieuse de Dewdney-Creignou-Hermann 53
3.3.2 La réduction parcimonieuse de $\frac{1}{3}$-Sat à 3-Col 55
3.4 Conclusion 56

3.1 Introduction

Ce chapitre réunit les réductions obtenues lors de cette thèse et pour lesquelles la parcimonie nous a semblé particulièrement difficile à établir. Ce sont toutes des réductions à 3-Col ou à Plan-3-Col. Rappelons que les réductions parcimonieuses, c'est-à-dire les transformations en temps polynomial qui préservent dans l'instance réduite le nombre exact de solutions de l'instance à réduire, sont intéressantes pour deux raisons :

1. Ces réductions préservent généralement non seulement les solutions mais aussi la structure même de ces solutions, puisqu'en pratique, elles réalisent par construction une correspondance bijective entre les ensembles de solutions, correspondance qui est elle-même calculable en temps polynomial, mais en fait souvent linéaire, et permet ainsi de conserver la complexité de l'énumération des solutions.
2. Réduire parcimonieusement Sat (ou un problème parcimonieusement équivalent) à un problème B dans NP permet d'obtenir, au delà de la NP-complétude de B, la #P-complétude du problème de comptage associé $\#B$, #Sat étant lui-même #P-complet [85], ainsi que la DP-complétude du problème Unique-B sous réductions polynomiales aléatoires, Unique-Sat étant lui même DP-complet sous ces réductions [87].

Lors de la conception d'une transformation entre deux problèmes de décision A à B, essayer de reproduire les solutions d'une instance à réduire dans l'instance réduite est le moyen le plus naturel de s'assurer de l'équivalence de la décision des deux instances, et il est donc très fréquent qu'une réduction établissant la NP-complétude d'un problème soit parcimonieuse. C'est par exemple le cas des deux réductions entre Plan-Sat et Plan-$\frac{1}{3}$-Sat, de la réduction de $\frac{1}{3}$-Sat à Plan-$\frac{1}{3}$-Sat, ou encore de la réduction de Plan-3-Col à Plan-$\frac{1}{3}$-Sat, qui ont toutes été présentées dans les préliminaires. Ce sera également le cas des réductions entre Plan-Sat et Plan-Hamilton présentées dans le chapitre suivant. Dans toutes ces réductions, la parcimonie vient naturellement, "sans forcer", car exprimer finement de la logique par des gadgets dans le problème d'arrivée se fait de façon relativement intuitive.

Ce n'est cependant pas vrai pour tous les problèmes NP-complets classiques, à commencer par ceux dont l'ensemble de solutions présente des symétries intrinsèques, comme la 3-Colorabilité. Typiquement, chaque solution d'une instance de 3-Col induit six solutions qui sont *isomorphes* par permutation de couleurs. Ainsi, le nombre de solutions de toute instance de 3-Col est un multiple de six. Un autre exemple est le problème Nae-3-Sat, où chacune des clauses (toutes de longueur 3 et monotones) doit etre satisfaite en ayant au moins une variable à vrai et une variable à faux. Chaque solution d'une instance de Nae-3-Sat induit deux solutions qui sont isomorphes par négation, et donc le nombre de solutions de toute instance de Nae-3-Sat est pair. Evidemment, de telles symétries intrinsèques sont absentes du problème Sat, et pour tout entier k, il est facile de construire une instance Sat de taille $O(\log k)$ possédant exactement k solutions : il suffit que les clauses simulent parcimonieusement la soustraction de deux nombres positifs signés écrits en binaire sur $1 + \lceil \log k \rceil$ bits en complémentation à 2, le premier portant la valeur $k - 1$, et de vérifier que le bit de signe du résultat est nul, attestant d'une différence positive (les k solutions sont alors les simulations de toutes les soustractions de k par les nombres de 0 à $k - 1$). Par conséquent, aucune réduction parcimonieuse ne peut évidemment exister de Sat à 3-Col, ou de Sat à Nae-3-Sat.

Cependant, il est naturel de considérer un ensemble de solutions isomorphes comme n'étant *qu'une et une seule solution*, et de compter les solutions modulo cet isomorphisme. Avec cette nouvelle convention de comptage, l'argument précédent disqualifiant 3-Col et Nae-3-Sat pour l'équivalence parcimonieuse à Sat ne tient plus, et il est naturel de se demander si des trans-

formations parcimonieuses existent de Sat à 3-Col et de Sat à Nae-3-Sat. En particulier, il n'est plus absurde de se demander si une instance particulière de 3-Col ou de Nae-3-Sat admet une solution unique : les problèmes Unique-3-Col et Unique-Nae-3-Sat sont bien définis, et exhiber des réductions parcimonieuses de Sat à 3-Col et Nae-3-Sat implique que Unique-3-Col et Unique-Nae-3-Sat sont aussi difficiles que de décider si une instance Sat admet une solution unique (problème Unique-Sat).

Dans tout ce chapitre, nous considérons à présent tout ensemble de solutions isomorphes comme *une seule et même* solution. Il est intéressant de noter qu'il existe déja une réduction parcimonieuse de Sat à Nae-3-Sat sous notre convention de comptage, puisque N. Creignou et M. Hermann [23] ont parcimonieusement réduit $\frac{1}{3}$-Sat à Nae-3-Sat, réduction qui est présentée dans ce chapitre. Cependant, nous ne connaissons pas de résultats similaires pour le problème plus intéressant 3-Col. En effet, les réductions classiques de Sat à 3-Col, e.g., celle présentée par Kozen dans son livre [60] et également exposée dans ce chapitre, ne sont même pas faiblement parcimonieuses, i.e., on ne peut même pas déduire une relation fonctionnelle précise entre le nombre de solutions de l'instance à réduire et celui de l'instance réduite, et ceci parce que chaque solution de l'instance à réduire se trouve dupliquée dans l'instance réduite un nombre de fois non-fixé, et qui dépend de la nature même de la solution.

Construire un gadget parcimonieux ayant un comportement logique précis dans 3-Col n'est pas chose facile : il se trouve toujours, ici ou là, un sommet "rebelle", dont on ne sait pas contrôler la coloration mais qui est indispensable à la correction du gadget. #3-Col est connu comme étant un problème #P-complet, mais de façon remarquable, la première preuve de ce résultat, dû à Linial [63], l'établit par une réduction parcimonieuse du problème #Stables dans les graphes bipartis, au problème #3-Col, également dans les graphes bipartis. Or, trouver un ensemble de sommets stables (non-joints par arête entre eux) est trivial si l'on n'exige pas de cardinalité minimale sur l'ensemble (auquel cas on obtiendrait le problème NP-complet Independent-Set), et trouver un 3-coloriage dans un graphe biparti (i.e., 2-coloriable) est également trivial : La #P-completude de #Stables ne donc peut être obtenue que par Turing-réduction et non par transformation (i.e., par Karp-réduction) de Sat. Une réduction *faiblement parcimonieuse* de Sat à 3-Col a été trouvée plus tard, prouvant à nouveau la #P-complétude de #3-Col de façon plus satisfaisante, par la composition de la réduction parcimonieuse de $\frac{1}{3}$-Sat à Nae-3-Sat déjà citée, et la réduction faiblement parcimonieuse de Nae-3-Sat à 3-Col, due à Dewdney [25], et exposée dans ce chapitre. Mais tandis que cette parcimonie faible est suffisante pour montrer la #P-complétude de #3-Col, elle ne nous est d'aucun secours pour comparer les expressivités relatives de Unique-3-Col et Unique-Sat, parce que la réduction de Dewdney multiplie le nombre de solutions par un facteur exponentiel en la taille de la donnée.

L'existence de réductions parcimonieuses depuis Plan-Sat peut également être posée pour Plan-3-Col et Plan-Nae-3-Sat, les versions planaires de 3-Col et Nae-3-Sat. Notons qu'en ce qui concerne Plan-Nae-3-Sat, c'est un problème trivial si l'on n'autorise pas les occurrences multiples à l'intérieur d'une même clause, conséquence du Théorème des quatre couleurs pour les graphes planaires : En effet, pour trouver une solution à une instance de Plan-Sat, il suffit de construire un graphe avec un sommet par variable, et de connecter en triangle les sommets associés aux variables apparaissant dans une même clause. Ce graphe est lui-même planaire, et donc 4-coloriable. De toute 4-coloration C sur $\{0,1,2,3\}$, on peut déduire la 2-coloration $C' : x \mapsto C(x) \bmod 2$. La 2-coloration C' ne laisse aucun triangle monochromatique, ce qui signifie que C' est aussi un assignement satisfaisant l'instance de Plan-Nae-3-Sat. Le problème reste polynomial même si l'on autorise des clauses de longueur quelconque.

En revanche, pour la 3-colorabilité, Plan-3-Col reste NP-complet et #Plan-3-Col reste #P-complet : Le premier résultat est montré par l'existence d'un crossover bien connu pour

Plan-3-Col et que l'on trouve dans tous les ouvrages de référence [68, 60, 33]. Ce crossover étant toutefois non-parcimonieux, le deuxième résultat a été établi par Hunt et al. [54, 55] en modifiant ce gadget pour le rendre faiblement parcimonieux. Ce crossover permet de contrôler le nombre de solutions lors de la planarisation mais avec un facteur de duplication qui est le carré d'une exponentielle du nombre de sommets.

Ainsi, à notre connaissance, il n'existe aucune réduction parcimonieuse de 3-Col à Plan-3-Col, de Sat à 3-Col, et de Plan-Sat à Plan-3-Col. En particulier, la difficulté de Unique-3-Col et Unique-Plan-3-Col restaient ouvertes jusqu'alors. Nous montrons dans ce chapitre que de telles réductions existent.

Proposition 3.1 3-Col *est parcimonieusement réductible à* Plan-3-Col *en temps quadratique.*

Ce résultat sera établi par l'existence d'un crossover parcimonieux pour Plan-3-Col. L'élaboration de cet objet, loin d'être triviale, nous occupera une bonne partie de ce chapitre. Beaucoup de gadgets intermédiaires seront nécessaires à sa construction.

Proposition 3.2 Sat *est parcimonieusement réductible à* 3-Col *en temps linéaire.*

Ce résultat sera obtenu par une réduction parcimonieuse intermédiaire de $\frac{1}{3}$-Sat à 3-Col. Curieusement, c'est cette réduction qui est la plus facile. En effet, une fois construits tous les objets nécessaires à l'élaboration du crossover parcimonieux, il est très facile de construire un simulateur de $\frac{1}{3}$-clause en les réutilisant. La réduction inverse étant déjà présentée dans les préliminaires, on déduit que :

Corollaire 3.3 Sat *et* 3-Col *sont équivalents sous réductions parcimonieuses et linéaires.*

Tous nos gadgets étant planaires, nous aurons le résultat analogue pour la version planaire de la 3-colorabilité :

Proposition 3.4 Plan-Sat *est parcimonieusement réductible à* Plan-3-Col *en temps linéaire.*

La réduction linéaire et parcimonieuse de 3-Col à $\frac{1}{3}$-Sat présentée en préliminaires ayant elle-même une version planaire, on déduit :

Corollaire 3.5 Plan-Sat *et* Plan-3-Col *sont équivalents sous réductions parcimonieuses et linéaires.*

Enfin, de ces réductions parcimonieuses, et conjointement, de la DP-complétude de Unique-Sat et Unique-Plan-Sat, nous déduisons que :

Corollaire 3.6 Unique-3-Col *et* Unique-Plan-3-Col *sont DP-complets sous réductions polynomiales aléatoires.*

A notre connaissance, ce dernier résultat est nouveau. Nos réductions utilisant toutes les mêmes objets, elles uniformisent également les preuves déjà existantes mais hétérogènes pour la NP-complétude et la #P-complétude de la 3-colorabilité, que ce soit dans le plan ou dans le cas général.

3.2 De la 3-colorabilité à la 3-colorabilité planaire

3.2.1 La réduction générique : le principe de la planarisation

La NP-complétude de Plan-3-Col est généralement prouvée par une transformation de 3-Col à Plan-3-Col qui élimine les croisements d'arêtes par l'utilisation de "crossovers". Rappelons qu'un crossover est un graphe planaire ayant pour fonction de maintenir identiques les couleurs de tous sommets situés dans les coins opposés d'un carré imaginaire, indépendamment des couleurs (elles-mêmes identiques) des sommets situés dans les coins adjacents (un crossover pour Plan-3-Col admet donc exactement $3 \times 3 = 9$ configurations). Dans cette section, nous noterons par

tout crossover pour Plan-3-Col. Une transformation quadratique de 3-Col à Plan-3-Col à base de crossovers est illustrée sur la Fig. 3.1 :

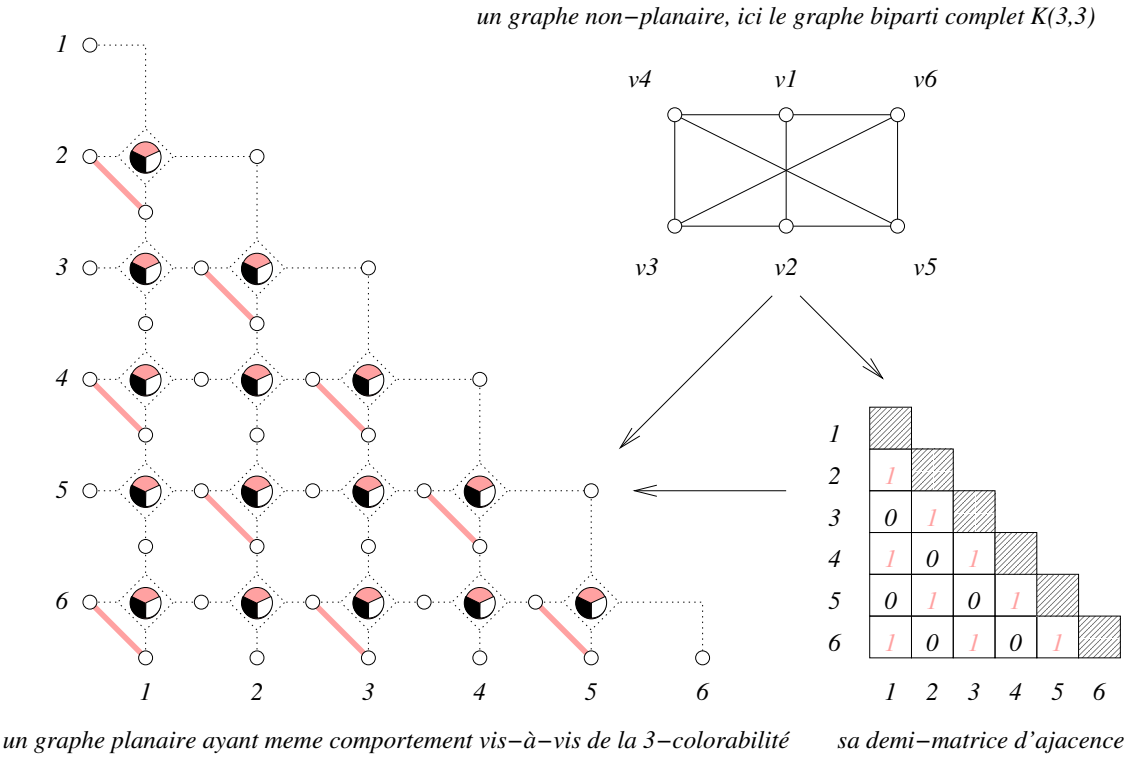


Fig. 3.1 – Transformation quadratique de 3-Col à Plan-3-Col

A partir du triangle inférieur gauche de la matrice d'adjacence M d'un graphe non-planaire $G(V = \{v_1, \cdots, v_n\}, E)$, on construit un graphe planaire $G'(V', E')$ ayant le même nombre de 3-colorations. Le plongement dans le plan de G' suit la grille physique des cases de ce triangle inférieur gauche de M :

1. On place un crossover $X_{i,j}$ dans chaque case $M_{i,j}$, $1 \leq i, j \leq n$, avec un sommet distingué sur chaque côté de la case. Les cases partageant un même côté partagent aussi les sommets distingués correspondant, de sorte que tous les sommets d'une même colonne ou rangée de crossovers aient la même couleur.

2. Pour tout $1 \leq i < n$, on fusionne le sommet Est du crossover $X_{i,i-1}$ avec le sommet Nord du crossover $X_{i+1,i}$, de sorte les sommets d'une colonne i aient la même couleur que les sommets de la rangée i et représentent tous le sommet $v_i \in V$.
3. Pour toute arête $(v_i, v_j) \in E$, on connecte par une arête les sommets distingués Sud et Ouest du crossover $X_{i,j}$.

Le graphe G' est de taille $\Theta(n^2)$ et il est 3-coloriable ssi G l'est aussi. De plus si le crossover est parcimonieux, alors la transformation l'est aussi. Dans toute la suite, nous nous intéressons à l'implémentation parcimonieuse d'un tel crossover.

Le crossover non-parcimonieux classique

Une implémentation classique du crossover ◈ est le graphe de gauche dans la Fig. 3.2, connu sous le nom de *diamond-graph* dans les livres de complexité tels que [60, 68, 33].

Les couleurs se propagent dans le graphe-diamant de la manière suivante : Une fois que l'on a arbitrairement fixé deux couleurs distinctes pour le sommet central c et son voisin i, les couleurs des sommets de la couronne intérieure sont toutes déterminées sous l'effet des quatre triangles centraux ; On peut ensuite choisir la couleur du sommet x parmi deux couleurs, et ce choix détermine successivement les couleurs de tous les sommets de la couronne extérieure, à la manière d'une cascade de dominos. Cette cascade se propage le long de l'anneau extérieur dans le sens anti-trigonométrique si la couleur de x est choisie identique à celle de c, et se propage dans le sens trigonométrique sinon. On montre que les combinaisons de couleurs possibles en x, y, x' et y' donnent au graphe-diamant la propriété de crossover pour Plan-3-Col.

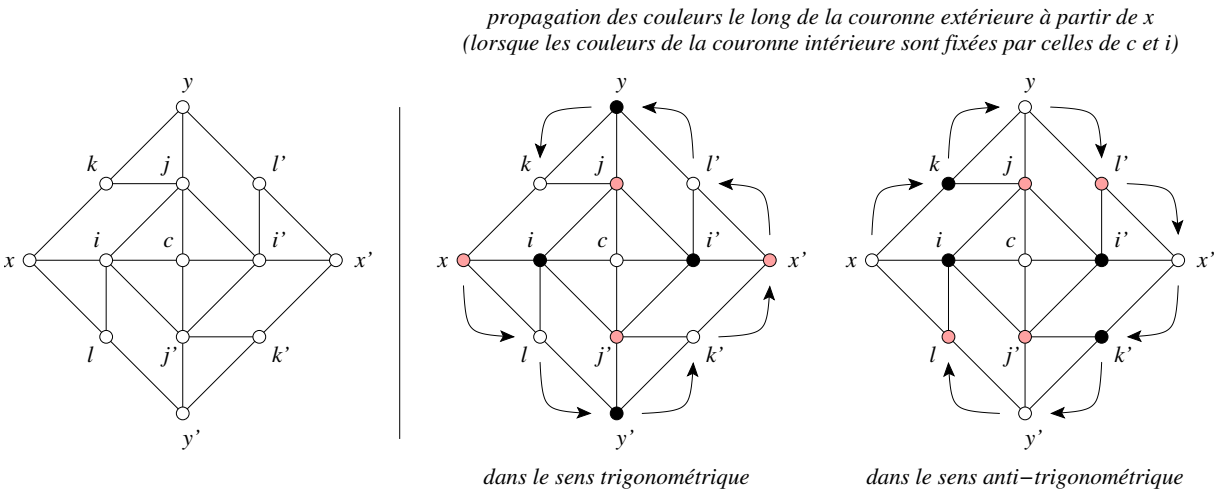


Fig. 3.2 – Propagation des couleurs dans le graphe-diamant

Propriété 3.7 *Le graphe-diamant de la Fig. 3.2 implémente non parcimonieusement un crossover ◈ pour* Plan-3-Col.

Preuve Soit C une 3-coloration du graphe-diamant telle que $C(c) = \textit{white}$ et $C(i) = \textit{black}$: Alors, nécessairement $C(j) = C(j') = \textit{gray}$ sous l'action des deux triangles (c, i, j) et (c, i, j'), et $C(i') = \textit{black}$ sous l'action des deux triangles (c, i', j) et (c, i', j'). Notre choix pour $C(i)$ entraîne $C(x) \neq \textit{black}$ via l'arête (x, i), ce qui nous laisse deux possibilités pour $C(x)$:

1. Soit $C(x) = gray$ – premier cas sur la Fig. 3.2 – et alors par effet de dominos depuis x dans le sens trigonométrique le long de la couronne extérieure : $C(l) = white, C(y') = black, C(k') = white, C(x') = gray, C(l') = white, C(y) = black$, et finalement $C(k) = white$.
2. Soit $C(x) = white$ – second cas sur la Fig. 3.2 – et alors par effet de dominos depuis x le long de la couronne extérieure, mais cette fois-ci dans le sens anti-trigonométrique : $C(k) = black$, $C(y) = white$, $C(l') = gray$, $C(x') = white$, $C(k') = black$, $C(y') = white$, et finalement $C(l) = gray$.

Dans les deux cas, $C(x) = C(x')$ et $C(y) = C(y')$, comme il est requis pour un crossover. Le premier cas est représentatif de la situation où les deux couleurs que l'on croise sont distinctes (configuration bicolore), c'est-à-dire lorsque $C(x) \neq C(y)$, alors que le second cas reflète le croisement de deux couleurs identiques (configuration monochrome), c'est-à-dire lorsque $C(x) = C(y)$. Maintenant, si l'on considère les couleurs des sommets distingués x, y, x' et y' comme fixées, nous pouvons remarquer dans le cas monochrome que la couleur des sommets i, i', k et k' peut être permutée avec celle des sommets j, j', l, et l', alors que ce n'est pas possible dans le cas bicolore. Par conséquent, chacune des trois configurations monochromes admet deux états locaux possibles alors qu'il y a une bijection entre les six configurations bicolores et les six états locaux associés, comme le montre la Fig. 3.3. Le graphe-diamant est donc un crossover non-parcimonieux pour Plan-3-Col. ■

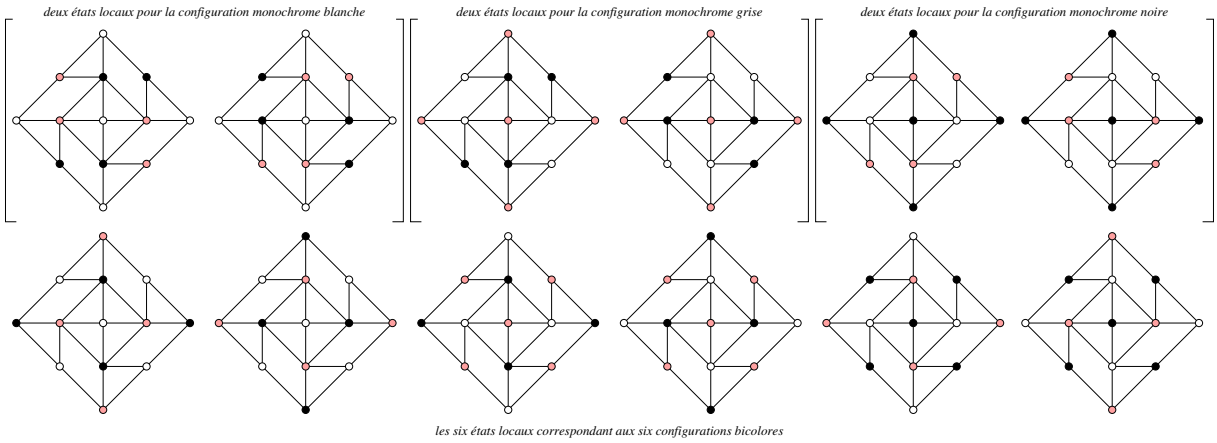


Fig. 3.3 – Les douze états locaux et les neuf configurations du graphe-diamant

Le crossover faiblement parcimonieux de Hunt et al.

L'utilisation du graphe-diamant pour transformer une instance G de 3-Col en une instance G' de Plan-3-Col ne permet pas a priori de déduire le nombre de solutions de G' en fonction de celui de G, ce graphe a donc été modifié par Hunt et al. dans [54] afin de le rendre faiblement parcimonieux. Rappelons que le graphe-diamant se comporte bien pour les configurations bicolores, mais qu'il se comporte mal (il dédouble les solutions) pour les configurations monochromes. La modification de Hunt et al. vise à ce que le graphe-diamant se comporte *également mal* pour les configurations bicolores, c'est-à-dire à dédoubler artificiellement les état locaux correspondant aux configurations bicolores. De sorte que toute configuration, qu'elle soit monochrome ou bicolore, acceptera deux états locaux.

Ce crossover donne une transformation faiblement parcimonieuse de 3-Col à Plan-3-Col. En utilisant c crossovers, une instance $G(V,E)$ de 3-Col avec $\#G$ solutions est transformée en une instance $G'(V',E')$ avec $\#G' = 2^c \times \#G$ solutions. Notons que c est lui-même $O(|V|^2)$ comme le montre le schéma de transformation de la Fig. 3.1, mais qu'il est également $\Omega(|V|^2)$ pour tout autre schéma de transformation sous l'hypothèse que Sat n'admet pas d'algorithme sous-exponentiel dans le cas général. De cette transformation et de la #P-complétude de #3-Col découle la #P-complétude de #Plan-3-Col. Le graphe-diamant modifié est présenté dans la Fig. 3.4. Nous montrons que c'est un crossover faiblement parcimonieux pour Plan-3-Col.

Propriété 3.8 *Le graphe-diamant modifié de la Fig. 3.4 est une implémentation faiblement parcimonieuse de crossover pour* Plan-3-Col.

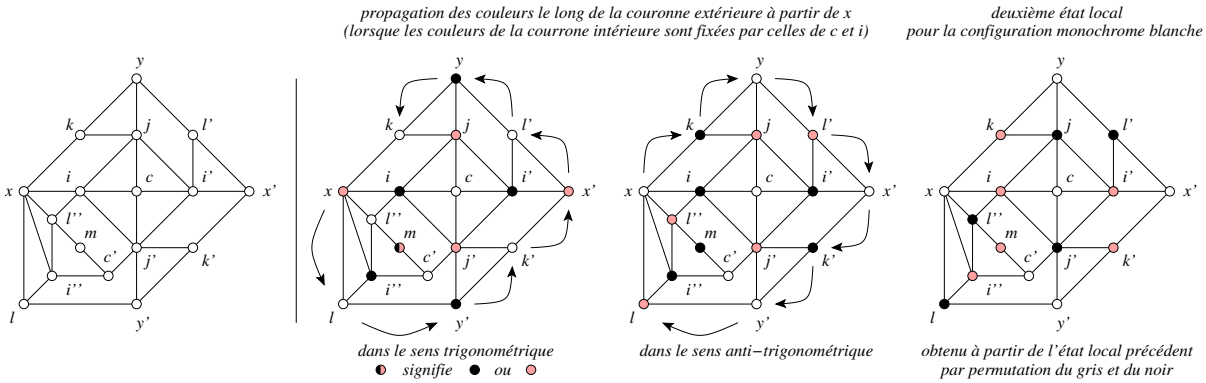


Fig. 3.4 – Propagation des couleurs dans le graphe-diamant modifié par Hunt et al.

Preuve Soit C une 3-coloration du graphe-diamant modifié, et fixons arbitrairement $C(c) = \mathit{white}$ et $C(i) = \mathit{black}$: Alors, nécessairement $C(j) = C(j') = \mathit{gray}$ sous l'action des deux triangles gauches du losange central, et $C(i') = \mathit{black}$ sous l'action de ses deux triangles droits. Notre choix pour $C(i)$ entraîne $C(x) \neq \mathit{black}$ via l'arête (x,i), ce qui nous laisse deux possibilités pour $C(x)$:

1. Soit $C(x) = \mathit{gray}$ – premier cas sur la Fig. 3.4. Alors $C(l'') = \mathit{white}$ sous l'action du triangle (x,i,l''), $C(i'') = \mathit{black}$ sous l'action du triangle (x,i'',l''), et $C(c') = \mathit{white}$ sous l'action du 2-chemin (j',c',i''). L'effet de domino le long de la couronne extérieure se produit ensuite exactement comme pour le graphe-diamant original dans le sens trigonométrique (cf. flèches), et la configuration produite est une configuration bicolore $\mathit{black}/\mathit{gray}$ de crossover, c'est-à-dire $C(x) = C(x') = \mathit{gray} \neq C(y) = C(y') = \mathit{black}$. Enfin, le sommet m n'ayant que deux voisins blancs, $C(m)$ est libre d'être gris ou noir, produisant ainsi artificiellement deux états locaux par configuration bicolore.
2. Soit $C(x) = \mathit{white}$ – deuxième cas sur la Fig. 3.4. Alors $C(l'') = \mathit{gray}$ sous l'action du triangle (x,i,l''), $C(i'') = \mathit{black}$ sous l'action du triangle (x,i'',l''), et $C(c') = \mathit{white}$ sous l'action du 2-chemin (j',c',i''). L'effet de domino le long de la couronne extérieure se produit ensuite exactement comme pour le graphe-diamant original dans le sens anti-trigonométrique (cf. flèches), et la configuration produite est une configuration monochrome white de crossover, c'est-à-dire $C(x) = C(x') = C(y) = C(y') = \mathit{white}$. Contrairement au cas précédent, la couleur de m n'est pas libre et sous l'action du 2-chemin (l'',m,c'), $C(m) = \mathit{black}$.

Le second état local pour une configuration est obtenu en permutant la couleur des sommets i, i', i'', m avec celle des sommets j, j', l, l', l'', comme montré à droite de la Fig. 3.4, c'est-à-dire en permutant les deux couleurs *gray* et *black*.

La Fig. 3.5 montre les dix-huit états locaux produits : en haut les six associés aux trois configurations monochromes, et en bas les douze associés aux six configurations bicolores. ■

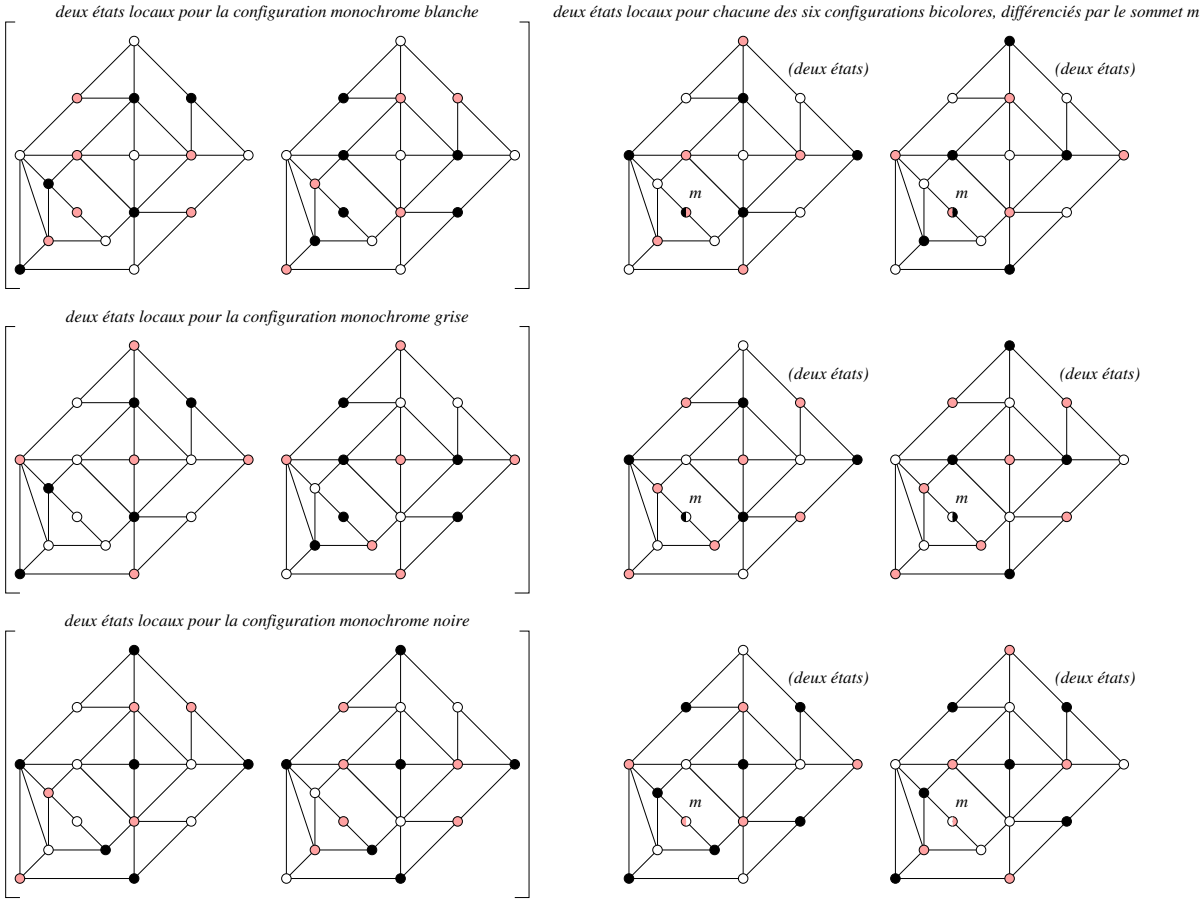


FIG. 3.5 – Les dix-huit états locaux et les neuf configurations du graphe-diamant modifié

3.2.2 Vers un crossover parcimonieux et radiant

La démarche de Hunt et al. est remarquable sur un point : plutôt que de raffiner le graphe-diamant original en éliminant les trois états locaux superflus des trois configurations monochromes, ils rajoutent artificiellement des états locaux aux configurations bicolores. Il est naturel de se demander si la démarche inverse est possible : Existe-t-il un crossover pour PLAN-3-COL qui soit strictement parcimonieux ?

De l'intérêt d'être radiant pour être parcimonieux

En fait, il est facile de s'apercevoir que le problème dont souffrent les graphes-diamants (original ou modifié) – à savoir la duplication de l'état local d'une configuration monochrome

par simple permutation des deux couleurs absentes de la configuration – est général si x, x', y et y' sont les seuls sommets distingués du crossover. Nous pourrions cependant envisager qu'en pré-coloriant s sommets supplémentaires utilisés en appui, il soit possible d'obtenir un gadget parcimonieux à $4+s$ sommets distingués ayant la propriété de crossover pour les 4 premiers sommets :

Définition 3.9 (Sommets distingués primaires et secondaires, palette de référence, configurations primaires et gadget à palette) *Soit G un gadget à $p+s$ sommets distingués $x_1, \cdots, x_p$* (les p sommets distingués primaires)*, $y_1, \cdots, y_s$* (les s sommets distingués secondaires)*, et d'ensemble de configurations $\overline{C}$. Soit C'' une 3-coloration de $y_1, \cdots, y_s$, appelée* palette de référence*. Pour toute configuration $C \in \overline{C}$, on note C/C'' la restriction de C sur $x_1, \cdots, x_p$ telle que $C(y_i) = C''(y_i)$ pour tout $i \leq s$. On appelle $\overline{C'} = \overline{C}/C'' = \{C/C'', C \in \overline{C}\}$ l'ensemble des* configurations primaires *de G pour la palette C''. Le gadget G est alors appelé* gadget à palette*, plus précisément de palette C'' et d'ensemble de configurations primaires $\overline{C'}$.*

Définition 3.10 (Crossover à palette) *Un gadget G à $4+s$ sommets distingués (4 primaires et s secondaires) et d'ensemble de configurations $\overline{C}$ est appelé* crossover à palette *s'il existe une palette C'' sur les s sommets distingués secondaires telle que G a la propriété de crossover pour les configurations primaires $\overline{C'} = \overline{C}/C''$ sur ses 4 sommets primaires.*

Nous allons construire un crossover à palette parcimonieux en partant sur la base de $s = 2$ sommets distingués. Pour justifier ce choix, montrons que l'on ne peut être parcimonieux avec moins de sommets secondaires et donc que notre choix est *minimum* :

Remarque 3.11 *Il n'existe pas de crossover à palette à $4+s$ sommets distingués pour* Plan-3-Col *qui soit parcimonieux si $s \in \{0, 1\}$.*

> **Preuve** Soient x, x', y, y' les quatre sommets distingués primaires et z le sommet distingué secondaire s'il existe. Remarquons tout d'abord qu'il n'existe pas de crossover $G(V, E)$ (parcimonieux ou non) constitué uniquement de sommets distingués : Quels que soient $u, v \in \{x, x', y, y'\}$, il ne peut y avoir d'arête (u, v) car sinon aucune configuration monochrome n'existe, et il ne peut y avoir d'arête (z, v) car sinon la configuration monochrome de couleur $C(z)$ n'existe pas. Tous les sommets sont alors isolés et toutes les configurations C telles que $C(x) \neq C(x')$ ou $C(y) \neq C(y')$ existent et G n'est pas un crossover.
>
> Donc il doit exister au moins un sommet t non-distingué dans G, adjacent à l'un des sommets distingués. Soit C un état local de G pour une configuration monochrome de couleur c_1, où c_1 est arbitraire si z n'existe pas, et où $c_1 = c(z)$ sinon. Soient c_2 et c_3 les deux autres couleurs, $c_2 = C(t) \neq c_1$ et $c_3 \neq c_1, c_2$. Soit $\{S_1, S_2, S_3\}$ la partition des sommets de G correspondant aux couleurs respectives c_1, c_2, c_3 (S_3 pouvant éventuellement être vide). On peut déduire un deuxième état local C' pour la même configuration monochrome de couleur c_1 en échangeant les couleurs c_2 et c_3 : Poser $C'(v \in S_1) = C(v) = c_1$, $C'(v \in S_2) = c_3$ et $C'(v \in S_3) = C(t) = c_2$. C' est bien une 3-coloration légale car s'il existait $(u, v) \in E$ tel que $C'(u) = C'(v)$ alors on aurait aussi $C(u) = C(v)$, une contradiction. De plus, C' représente bien une configuration monochrome de couleur c_1 puisque $\{x, y, x', y'\} \subseteq S_1$. Enfin, C' est bien distinct de C puisque $C(t) = c_2 \neq C'(t) = c_3$. ■

Par conséquent, il nous faut au moins $s = 2$ sommets distingués secondaires pour espérer obtenir un crossover à palette parcimonieux pour Plan-3-Col. Dans ces conditions, et afin que le crossover soit utilisable, il est souhaitable que la palette de référence portée par ces sommets soient communiquée à tous les crossovers. Pour ce faire, les crossovers doivent propager les couleurs de la palette de référence de voisin à voisin.

Deux crossovers dans un graphe planaire $G(V, E, F)$ sont dit voisins s'ils partagent la frontière d'une face $f \in F$. La co-accessibilité de deux crossovers est la clôture transitive de la relation de voisinage, i.e., deux crossovers X et X' sont dit co-accessibles s'ils sont voisins, ou s'il existe une suite $X = X_1, X_2, \cdots, X_p = X'$ de crossovers tel que X_i et X_{i+1} sont voisins pour tout $i < p$. Un ensemble de crossovers est dit connexe si ces crossovers sont tous co-accessibles deux à deux.

Problème : En supposant que l'ensemble des crossovers est connexe, comme c'est le cas pour le schéma de transformation de la Fig. 3.1, les sommets distingués secondaires de deux crossovers voisins dont on souhaite faire partager les couleurs ne sont pas forcément placés sur la même face. Nous avons donc tout intérêt à ce que tout crossover dispose d'une copie de ces deux sommets dans chacun des quatre secteurs situés entre ses quatre sommets distingués primaires *et qu'il assure la cohérence de leur coloration.* Nous qualifierons de *radiant* un gadget ayant une telle caractéristique. L'idée est celle d'une source lumineuse ponctuelle rayonnant dans toutes les directions. C'est ce qu'on exprime dans la définition suivante :

Définition 3.12 (Secteurs, paires de référence, et radiance) *Soit G un gadget planaire à palette et à $p+2p$ sommets distingués $x_1, b_1, g_1, \cdots, x_p, b_p, g_p$ disposés anti-trigonométriquement dans cet ordre, où les x_i – resp. b_i et g_i – sont les sommets distingués primaires – resp. secondaires. On appelle un couple (b_i, g_i) une* paire de référence *et (x_i, x_{i+1}) son* secteur *(avec par convention $p + 1 = 1$). Le gadget G est appelé radiant si pour toute configuration C de G, on a pour tout $1 \leq i, j \leq p$:*

1. *$C(g_i) \neq C(b_i)$, et*
2. *$C(b_j) = C(b_i)$ et $C(g_j) = C(g_i)$.*

Autrement dit, une paire de référence est nécessairement bicolore et ses couleurs se propagent aux paires de référence des autres secteurs. La palette de référence est donc entièrement déterminée par la coloration d'une seule paire de référence, que l'on appellera aussi palette de référence *par abus de langage.*

Remarque 3.13 *Par souci de simplification, Pour chaque gadget radiant, nous nommerons indistinctement par b, resp. g, tous ses sommets secondaires b_i, resp. g_i.*

Dans ce qui suit, nous élaborons un crossover parcimonieux et radiant pour Plan-3-Col et nous l'utilisons pour construire une réduction parcimonieuse de 3-Col à Plan-3-Col.

Une analogie optique

La réalisation de notre crossover parcimonieux et radiant passe par la conception de plusieurs objets intermédiaires, radiants ou non-radiants. Dans toute cette section, nous allons exclusivement nous intéresser à la définition de l'interface de ces objets et à l'élaboration de la stratégie exploitant leur comportement (c'est-à-dire l'ensemble de leurs configurations primaires) pour atteindre l'objectif final. L'implémentation parcimonieuse de ces objets ne débutera qu'en Section 3.2.3.

Nous construirons en particulier des objets que nous appellerons *crossovers restreints* : Un crossover restreint est un objet qui n'a la propriété de crossover que sous des hypothèses d'utilisation limitée. Deux crossovers restreints seront introduits : le *crossover exclusif* (qui est non-radiant) et le crossover bicolore (qui est radiant) :

Définition 3.14 (Crossover exclusif) *Un gadget planaire G à 4 sommets distingués x, y, x' et y' disposés anti-trigonométriquement dans cet ordre est un* crossover exclusif *et est noté* ⊕ *s'il vérifie les deux propriété suivantes :*

1. *Pour toute configuration C, $C(x) = C(x') \neq C(y) = C(y')$*
2. *Pour toutes couleurs distinctes c et c', il existe une configuration C tel que $C(x) = c$ et $C(y) = c'$.*

La Fig. 3.6 résume les six configurations requises pour un crossover exclusif. Cet objet, extrêmement simple à implémenter, va nous permettre de rendre nos gadgets radiants. Il permettra aussi d'implémenter le crossover restreint suivant ·

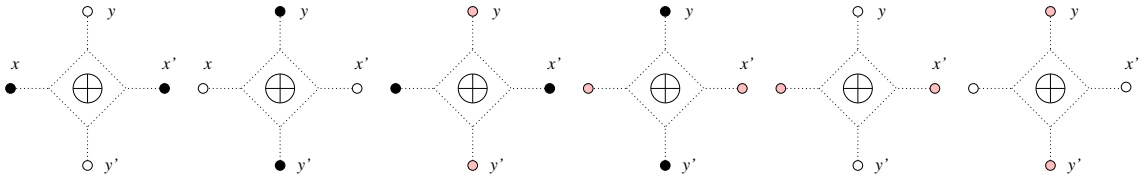


FIG. 3.6 – Les six configurations requises pour le crossover exclusif

Définition 3.15 (Crossover bicolore) *Un gadget planaire G à palette et à 4 sommets distingués primaires x, y, x' et y' disposés anti-trigonométriquement dans cet ordre est un crossover bicolore et est noté* ◈ *s'il vérifie les trois propriétés suivantes, pour une palette de référence C'' choisie :*

1. *La palette C'' détermine exactement une couleur interdite c'' pour les sommets distingués primaires x, x', y et y'.*
2. *Pour toute configuration C, $C(x) = C(x') \neq c''$ et $C(y) = C(y') \neq c''$.*
3. *Pour toutes couleurs c, c' distinctes de c'' mais non nécessairement distinctes entre elles, il existe une configuration C tel que $C(x) = c$ et $C(y) = c'$.*

La Fig. 3.7 résume les six configurations primaires requises pour un crossover bicolore.

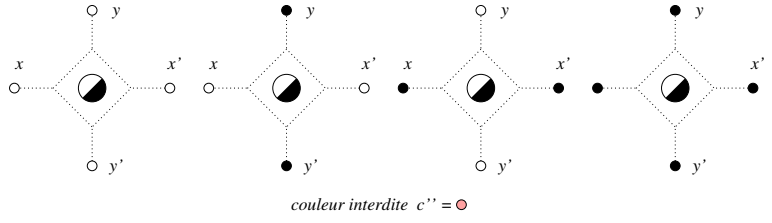


FIG. 3.7 – Les quatre configurations primaires requises pour le crossover bicolore

Remarque 3.16 *En reprenant exactement la même preuve que celle de la remarque 3.11 avec $s = 0$, il est facile de voir qu'il n'existe pas de crossover bicolore qui soit parcimonieux et sans sommet distingué secondaire. Nous verrons qu'il en existe un avec $s = 1$ sommet secondaire. Nous construirons cependant un crossover bicolore parcimonieux et radiant, et où il y a donc deux sommets secondaires par secteur.*

La conception de notre crossover non-restreint est guidée par une analogie optique : la décomposition et recomposition de la lumière qui se produit au travers d'un prisme. Ici, on décide arbitrairement que le blanc et le noir sont des couleurs pures. Les trois couleurs blanc, gris, et noir existent également sous forme composite. Une couleur composite c est vue comme la somme d'une fréquence basse $low(c)$ et haute $high(c)$. Les décompositions chromatiques opérées par le prisme sont les suivantes :

- le noir se décompose en $high(black) = black$ et $low(black) = black$,
- le gris se décompose en $high(gray) = black$ et $low(gray) = white$,
- le blanc se décompose en $high(white) = white$ et $low(white) = white$.

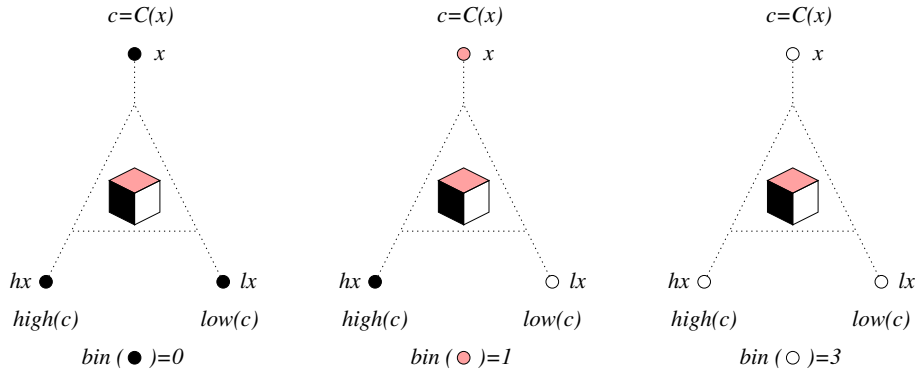


FIG. 3.8 – Les trois configurations primaires requises pour le prisme

Une autre façon de le voir est d'interpréter les couleurs pures noire et blanche comme les bits 0 et 1 resp., les couleurs composites noire, grise, blanche comme les nombres sur deux bits 0, 1, et 3 resp., et la décomposition chromatique d'une couleur composite c comme l'écriture de c en binaire : $bin(c) = (high(c), low(c))$, $high(c)$ étant le bit de poids fort et $low(c)$ le bit de poids faible.

Si nous disposons d'un gadget planaire noté agissant comme un prisme, c'est-à-dire un gadget planaire à trois sommets distingués primaires x, h_x et l_x autorisant exactement les trois configurations primaires de la forme $(C(x), C(h_x) = high(C(x)), C(l_x) = low(C(x)))$ montrées sur la Fig. 3.8, alors nous avons un schéma pour fabriquer notre crossover non-restreint en combinant quatre prismes avec quatre crossovers bicolores comme il est montré sur la Fig. 3.9. L'idée est de ne pas tenter de propager directement la couleur composite en x (resp. y) vers x' (resp. y') mais de décomposer la couleur composite en x (resp. y) au travers d'un premier prisme et de la recomposer en x' (resp. y') au travers d'un deuxième prisme. On croise alors quatre couleurs au lieu de deux, mais comme elles sont pures, il suffit d'un crossover bicolore pour les croiser en chacun des quatre points de croisement.

L'action du prisme se divise en deux applications $c \mapsto high(c)$ et $c \mapsto low(c)$ qui laissent toutes deux invariantes les couleurs $c = black$ et $c = white$, et qui ne diffèrent que dans l'image de $gray$: On a $high(gray) = black$ et $low(gray) = white$, autrement dit $high$ assombrit le gris tandis que low l'éclaircit. Pour cette raison, on appellera *convertisseur sombre unidirectionnel*, et on notera

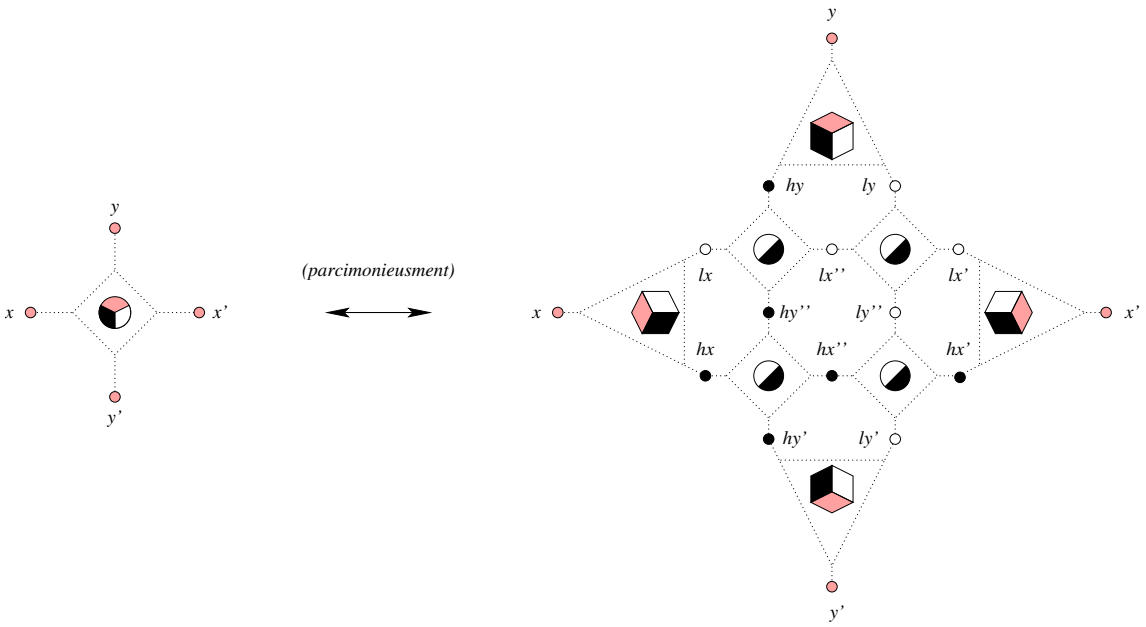


FIG. 3.9 – Schéma de construction d'un crossover non-restreint

, tout gadget de sommets distingués primaires (x, x') dont les configurations primaires sont de la forme $(C(x), high(C(x))$. De même, on appellera *convertisseur clair unidirectionnel*, et on notera , tout gadget dont les configurations primaires sont de la forme $(C(x), low(C(x))$. Voir Fig. 3.10.

Remarque 3.17 *Le qualificatif* unidirectionnel *vient du fait que l'ensemble des configurations primaires n'est pas stable sous permutation de x et x'. Autrement dit, la conversion est orientée. Nous verrons que les convertisseurs unidirectionnels sont implémentables en terme d'autres convertisseurs, dits* bidirectionnels*, laissant l'ensemble des configurations stable sous permutation de x et x'.*

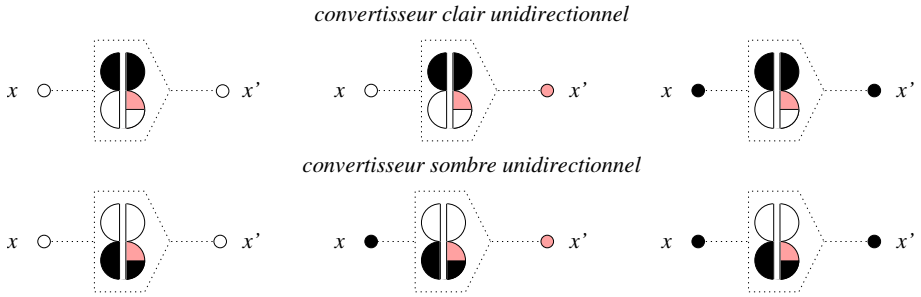


FIG. 3.10 – Les trois configurations primaires requises pour les convertisseurs unidirectionnels

Les convertisseurs unidirectionnels sont de véritables "couteaux suisses" et ne servent pas seulement à implémenter les prismes : Nous verrons en Section 3.3 qu'ils sont la clé de voûte pour construire les gadgets logiques donnant l'équivalence parcimonieuse de SAT et 3-COL ; De plus, les convertisseurs unidirectionnels servent aussi à élaborer un schéma simulant un crossover bicolore à partir d'un crossover exclusif .

En effet, il suffit de remarquer qu'un crossover exclusif fait à peu de choses près le travail d'un crossover bicolore si nous ne lui présentons que les couleurs noire et blanche en ses sommets distingués i, j, i', j' : Seules les deux configurations monochrome blanche et monochrome noire ne passent pas. Nous réglons le problème comme suit :

1. Pour faire passer la configuration monochrome blanche, il suffit que la couleur blanche de i et i' soit préalablement convertie en gris, de sorte que le crossover exclusif croise bien deux couleurs distinctes blanche et grise. Ceci est réalisé en connectant des convertisseurs unidirectionnels clairs en i et i''.
2. De même, pour faire passer la configuration monochrome noire, il suffit que la couleur noire de j et j' soit préalablement convertie en gris, de sorte que le crossover exclusif croise bien deux couleurs distinctes grise et noire. Ceci est réalisé en connectant des convertisseurs unidirectionnels sombre en j et j''.

Notre schéma de construction du crossover bicolore est résumé dans la Fig. 3.11. Le schéma est en lui-même non-parcimonieux mais nous verrons en Section 3.2.4 que ce problème se corrige facilement.

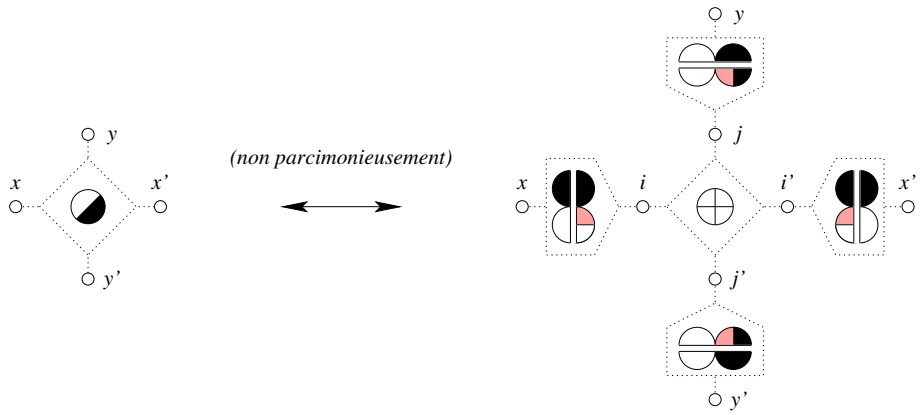


FIG. 3.11 – Schéma de construction d'un crossover bicolore

3.2.3 L'implémentation parcimonieuse des convertisseurs

Tout repose donc sur l'implémentation d'un convertisseur unidirectionnel parcimonieux et radiant. Pour l'obtenir, nous passerons par plusieurs raffinements intermédiaires :

1. Nous construirons tout d'abord un convertisseur planaire à palette, mais bidirectionnel, non-parcimonieux et non-radiant ;
2. Nous le rendrons ensuite parcimonieux mais en perdant la planarité (certains sommets distingués secondaires ne se trouvant pas sur la face externe) ;
3. Nous restaurerons la planarité du gadget en faisant échapper ces sommets vers la face externe grâce au crossover exclusif dont l'implémentation sera également introduite.
4. Enfin, nous le rendrons à la fois unidirectionnel et radiant.

Un convertisseur bidirectionnel planaire non-parcimonieux

Nous définissons tout d'abord les notions de couleurs claires et sombres, et les relations d'équivalence associées $\overset{wg}{\equiv}$ (pour *white/gray* c'est-à-dire les couleurs claires) et $\overset{gb}{\equiv}$ (pour *gray/black*

c'est-à-dire les couleurs sombres), puis nous définissons les gadgets établissant ces équivalences entre les couleurs de ses sommets distingués primaires :

Définition 3.18 (couleurs claires et sombres, relations d'équivalence $\overset{wg}{\equiv}$ et $\overset{gb}{\equiv}$) *L'ensemble des couleurs claires est par définition* $light = \{white, gray\}$ *et celui des couleurs sombres est* $dark = \{gray, black\}$. *Le gris est donc à la fois clair et sombre. La relation d'équivalence sombre, notée* $\overset{gb}{\equiv}$, *et la relation d'équivalence claire, notée* $\overset{wg}{\equiv}$, *sont définies par :*

$$c \overset{gb}{\equiv} c' \quad \text{ssi} \quad c \in dark \iff c' \in dark$$
$$c \overset{wg}{\equiv} c' \quad \text{ssi} \quad c \in light \iff c' \in light$$

Un *convertisseur sombre bidirectionnel* est un gadget à deux sommets distingués primaires x et x' ayant pour configurations primaires toutes les paires $(C(x), C(x'))$ vérifiant $C(x) \overset{gb}{\equiv} C(x')$. De même, un *convertisseur clair bidirectionnel* a pour configurations primaires toutes les paires $(C(x), C(x'))$ vérifiant $C(x) \overset{wg}{\equiv} C(x')$. Ces configurations primaires sont résumées sur la Fig. 3.12 pour les deux convertisseurs bidirectionnels. En les comparant avec celles de la Fig. 3.10, on voit que l'ensemble des configurations primaires d'un convertisseur bidirectionnel inclut celui de son homologue unidirectionnel.

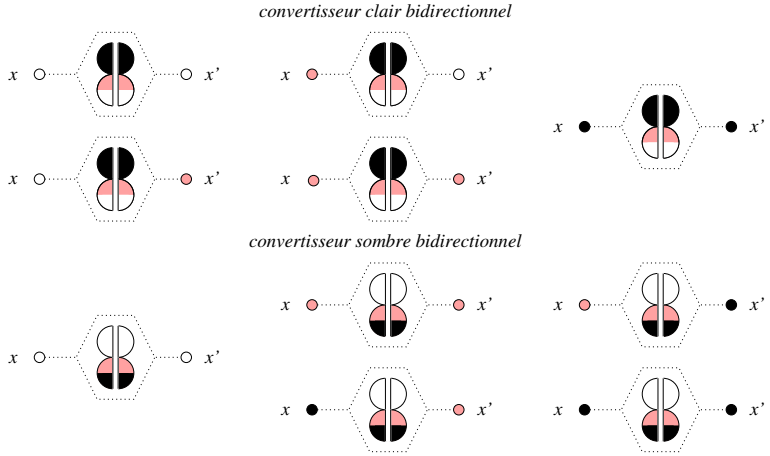


FIG. 3.12 – Les cinq config. primaires requises pour les deux convertisseurs bidirectionnels

L'ensemble des configurations primaires d'un convertisseur (unidirectionnel ou bidirectionnel) n'étant pas stable par permutation de couleurs, il est nécessaire d'avoir $s > 0$ sommets distingués secondaires. Nous construisons un premier convertisseur sombre bidirectionnel à $s = 2$ sommets distingués secondaires b et g, avec pour palette de référence les deux témoins de la classe sombre, $C(b) = black$ et $C(g) = gray$:

Propriété 3.19 *Le gadget planaire à palette de la Fig. 3.13, de sommets distingués* x, b, x', g *disposés anti-trigonométriquement dans cet ordre, implémente non parcimonieusement un convertisseur sombre bidirectionnel*  *pour les sommets distingués primaires* x *et* x' *dès que les sommets distingués secondaires* b *et* g *sont resp. black et gray.*

Preuve Soit C un état local du gadget tel que $C(b) = black$ et $C(g) = gray$. Il y a trois cas selon la couleur de x :

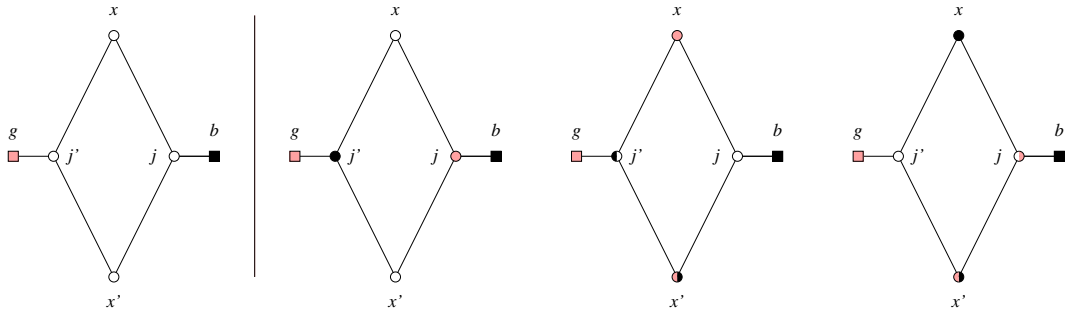


FIG. 3.13 – Un convertisseur sombre bidirectionnel planaire mais non-parcimonieux

1. Supposons que $C(x) = white$ (premier cas sur la Fig. 3.13). Alors à droite, $C(j) = gray$ à cause du 2-chemin (x, j, b), A gauche, $C(j') = black$ à cause du 2-chemin (x, j', g). Enfin, $C(x') = white$, à cause du 2-chemin (j, x', j'), et on a bien $C(x) = white \overset{gb}{\equiv} C(x') = white$;
2. Supposons que $C(x) = gray$ (deuxième cas sur la Fig. 3.13). Alors à droite, $C(j) = white$ à cause du 2-chemin (x, j, b). A gauche, $C(j') = black$ ou $white$, x et g étant tous deux $gray$. Dans le premier cas, $C(x') = gray$ à cause du 2-chemin (j, x', j'). Dans le deuxième cas, $C(x')$ est libre d'être $gray$ ou $black$, mais on a toujours $C(x) = gray \overset{gb}{\equiv} C(x') \in dark$. La configuration $(gray, gray)$ admet deux états locaux.
3. Le troisième cas $C(x) = black$ est le symétrique du précédent : $C(j') = white$, $C(j) = gray$ ou $white$, et resp. $C(x') = black$ ou $gray$. La configuration $(black, black)$ admet deux états locaux.

L'ensemble des configurations $(C(x), C(x'))$ est donc bien $(white, white)$, $(gray, black)$, $(black, gray)$, $(black, black)$, et $(gray, gray)$, qui sont exactement les configurations requises pour un convertisseur sombre bidirectionnel. En revanche, le gadget est non-parcimonieux : il admet sept états locaux pour cinq configurations, les deux configurations monochromes sombres ayant chacune deux états locaux. ■

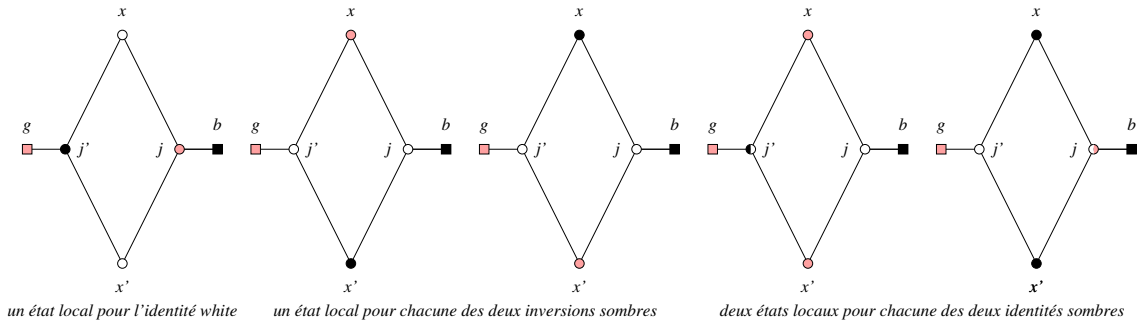


FIG. 3.14 – La non-parcimonie du convertisseur est due aux identités sombres

Un convertisseur bidirectionnel parcimonieux non-planaire

L'étude de la Fig. 3.14 nous montre que le convertisseur bidirectionnel sombre précédent se comporte de façon parcimonieuse dans un certain nombre de configurations primaires : celle de

l'identité blanche ainsi que les deux inversions sombres. Seules les deux identités sombres posent problème en laissant un degré de liberté à $C(j)$ ou $C(j')$. Afin de rendre le gadget parcimonieux, il faut supprimer l'un des deux choix sur $C(j)$ pour l'identité noire et supprimer l'un des deux choix sur $C(j')$ pour l'identité grise.

L'idée est la suivante : réinjecter un exemplaire du gadget dans lui même avec pour sommets distingués primaires j et j' comme montré sur la Fig. 3.15. Ceci a pour effet d'imposer l'équivalence $C(j) \overset{gb}{\equiv} C(j')$ et donc d'éliminer les états locaux tel que $C(j) \overset{gb}{\not\equiv} C(j')$.

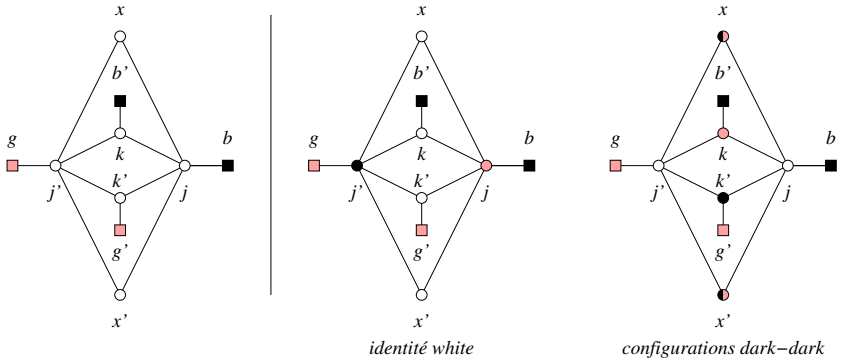


FIG. 3.15 – Le convertisseur devient parcimonieux s'il est réinjecté dans lui même

Remarquons tout d'abord qu'aucune des trois configurations primaires représentée par un seul état local n'est supprimée. En effet :

1. Dans le cas de l'identité blanche, on a $C(j) = \mathit{black}$ et $C(j') = \mathit{gray}$, et donc $C(j) \overset{gb}{\equiv} C(j')$.
2. Dans le cas des deux inversions sombres, on a $C(j) = \mathit{white}$ et $C(j') = \mathit{white}$ et donc $C(j) \overset{gb}{\equiv} C(j')$ également.

En revanche, exactement un des deux états locaux de chacune des identités sombres est supprimé :

1. Dans le cas de l'identité grise, $C(j)$ était toujours *white* et $C(j')$ pouvait être *white* ou *black*. Mais comme $\mathit{white} \overset{gb}{\equiv} \mathit{white}$ et $\mathit{white} \overset{gb}{\not\equiv} \mathit{black}$, le deuxième état local disparaît au profit du premier : $C(j') = \mathit{white}$.
2. De même, dans le cas de l'identité noire, $C(j')$ était toujours *white* et $C(j)$ pouvait être *white* ou *gray*. Mais comme $\mathit{white} \overset{gb}{\equiv} \mathit{white}$ et $\mathit{white} \overset{gb}{\not\equiv} \mathit{gray}$, le deuxième état local disparaît au profit du premier : $C(j) = \mathit{white}$.

Ceci donne un comportement parcimonieux au convertisseur de sommets distingués primaires (x, x'), mais encore faut-il que le convertisseur de sommets distingués primaires (j, j') se comporte lui-même parcimonieusement. C'est le cas, puisque de fait, les seules équivalences $C(j) \overset{gb}{\equiv} C(j')$ qui se produisent sont $\mathit{white} \overset{gb}{\equiv} \mathit{white}$ (quatre fois) et $\mathit{gray} \overset{gb}{\equiv} \mathit{black}$ (une fois), or nous avons déjà remarqué que l'identité blanche et l'inversion sombre n'admettent chacune qu'un seul état local. Par conséquent :

Propriété 3.20 *Le gadget à palette de la Fig. 3.15 de sommets distingués x, x', b, g, b', g' implémente parcimonieusement un convertisseur sombre bidirectionnel pour x et x' dès que $C(b) = C(b') = \mathit{black}$ et $C(g) = C(g') = \mathit{gray}$.*

Un convertisseur bidirectionnel parcimonieux et planaire

Malgré l'absence de croisement d'arêtes, le gadget de la Fig. 3.15 n'est pas planaire, car deux de ses sommets distingués secondaires b' et g' ne se trouvent pas sur la face externe. On a donc gagné la parcimonie au prix de la planarité. On souhaite restaurer cette planarité en conservant la parcimonie.

Idéalement, il faudrait que b' aille rejoindre son frère jumeau b dans le secteur à droite de (x, x'). Le sommet g' devrait faire de même avec g dans le secteur gauche, ce qui réduirait du même coup le nombre de sommets distingués secondaires de quatre à deux. Mais pour ce faire, il faudrait franchir les arêtes (x', j') et (x, j). Ceci nécessite en première analyse la pose de deux crossovers parcimonieux sur ces arêtes, ce dont nous ne disposons pas puisque c'est exactement ce que nous nous employons à construire. Cependant, la totalité de la puissance d'un crossover parcimonieux n'est pas nécessaire. En effet, si le premier crossover croise les couleurs de g et j', alors il ne croise que du gris avec du blanc ou avec du noir. De même si le deuxième crossover croise les couleurs de b et j, alors il ne croise que du noir avec du blanc ou avec du gris. Les crossovers n'ont donc besoin d'être parcimonieux que pour les configurations bicolores (les configurations monochromes n'apparaissant jamais).

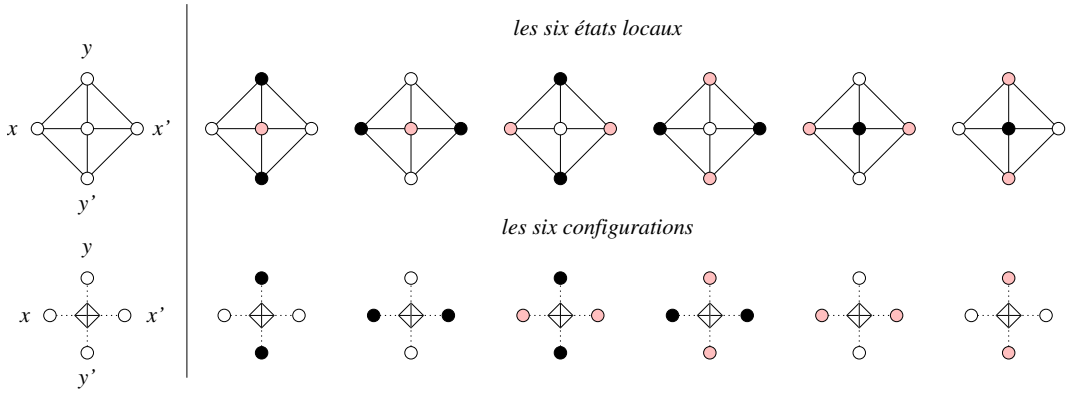


FIG. 3.16 – Un crossover exclusif parcimonieux

Nous avons déjà remarqué que le graphe-diamant de la Fig. 3.2 avait un comportement parcimonieux sur les configurations bicolores, et par conséquent, on peut l'utiliser ici malgré sa non-parcimonie sur les configurations monochromes. Cependant, un crossover exclusif parcimonieux suffit également et présente l'avantage d'être beaucoup plus simple. En effet, il est trivial de voir que :

Propriété 3.21 *Le graphe de la Fig. 3.16, noté* ⌖*, est une implémentation parcimonieuse du crossover exclusif* ⊕*.*

Des propriétés 3.20 et 3.21 , il découle immédiatement que :

Propriété 3.22 *Le gadget à palette de la Fig. 3.17, noté* ⌷*, et de sommets distingués x, b, x', g disposés anti-trigonométriquement dans cet ordre, implémente parcimonieusement un convertisseur sombre bidirectionnel* ⊗ *pour x et x' dès que $C(b) = black$ et $C(g) = gray$.*

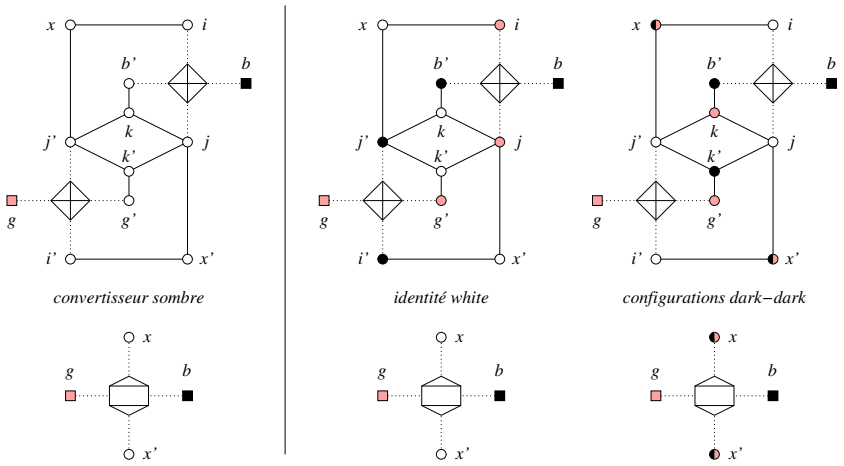


FIG. 3.17 – Un convertisseur sombre bidirectionnel, planaire et parcimonieux

Un convertisseur unidirectionnel parcimonieux et radiant

Le gadget de la Fig. 3.17 présente l'inconvénient de ne pas être radiant : Nous souhaiterions obtenir un convertisseur ayant une paire (b, g) dans chacun des secteurs de part et d'autre de (x, x') de telle sorte que les couleurs se propagent entre les deux paires. Traverser les arêtes (x', j) et (x, j') en utilisant deux crossovers exclusifs comme nous l'avons fait pour traverser les arêtes (x, j) et (x', j') n'est pas possible, car des configurations monochrome grise et monochrome noire se présenteraient pour ces deux crossovers exclusifs. L'obtention de la radiance est donc obtenue non pas en modifiant ◊ mais en disposant en série des exemplaires de ◊, ◊, ◊ et ◊ dans cet ordre, comme montré sur la Fig. 3.18. L'idée est de propager les couleurs de b et g d'un secteur à l'autre via les crossovers exclusifs ◊. Les deux convertisseurs sombres bidirectionnels évitent que des configurations monochromes sombres se présentent aux crossovers exclusifs.

Propriété 3.23 *Le gadget à palette de la Fig. 3.18, noté ◊, et de sommets distingués x, b, g, x', b, g disposés anti-trigonométriquement dans cet ordre, est radiant, et il constitue une implémentation parcimonieuse du convertisseur sombre unidirectionnel ◊ pour les sommets distingués primaires x et x' dès lors qu'une paire de référence arbitraire (b, g) est coloriée $(black, gray)$.*

Preuve Soit C une configuration de ◊. Tous les sommets nommés b ont la même couleur $C(b)$ à cause de la chaîne de crossovers exclusifs les reliant. De même, tous les sommets nommés g ont la même couleur $C(g)$. De plus $C(b) \neq C(g)$ car b et g partagent l'arête d'un crossover exclusif. Le gadget ◊ est donc radiant. Supposons maintenant que $(C(b), C(g)) = (black, gray)$. Tous les convertisseurs bidirectionnels ◊ ont alors leurs sommets distingués secondaires (b, g) coloriés en $(black, gray)$, et se comportent donc comme des convertisseurs sombres ◊. Maintenant, du fait que x et i partagent l'arête d'un crossover exclusif avec g, nous avons $C(x) = C(i) \neq gray$, ce qui laisse deux cas :

1. Soit $C(x) = C(i) = white$ (premier cas sur la Fig. 3.18). Alors $C(j) = white$ car $C(j) \overset{gb}{\equiv} C(i)$ par propriété du convertisseur bidirectionnel sombre. Enfin, pour les même raisons, $C(k) = white$, et $C(x') = white$ car $C(x') \overset{gb}{\equiv} C(k)$.
2. Soit $C(x) = C(i) = black$ (second cas sur la Fig. 3.18). Alors $C(j) \in dark$ car $C(j) \overset{gb}{\equiv} C(i)$. Mais $C(j) = C(k) \neq black$ car j et k partagent un crossover

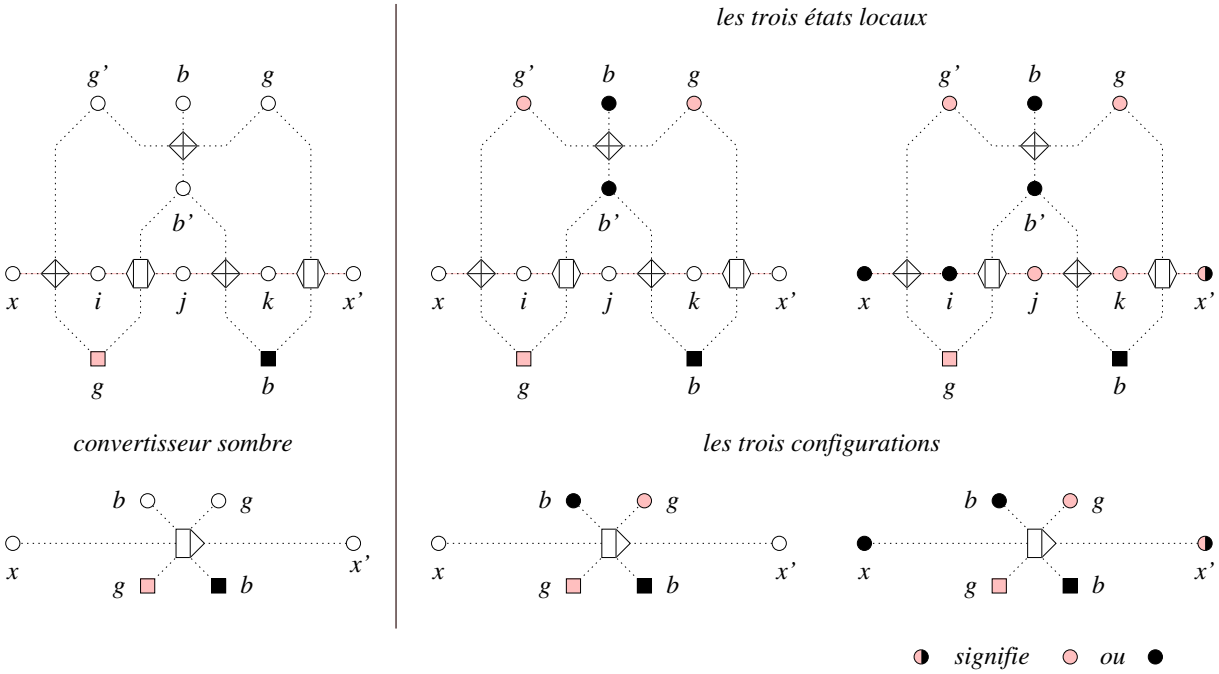


FIG. 3.18 – Un convertisseur sombre unidirectionnel, radiant, et parcimonieux

exclusif avec b et b'. Donc $C(j) = C(k) = gray$. Puisque $C(x') \overset{gb}{\equiv} C(k)$ on a finalement $C(x') = black$ ou $gray$.

Le gadget est parcimonieux et ses trois configurations primaires $(C(x), C(x'))$ sont $(white, white)$, $(black, black)$ et $(black, gray)$ comme il est requis pour un convertisseur unidirectionnel sombre. ■

Remarque 3.24 *Par simple permutation des couleurs white et black, il est immédiat de constater que le convertisseur , resp. , est une implémentation parcimonieuse et radiante du convertisseur bidirectionnel clair , resp. unidirectionnel clair dès lors que* $C(g) = gray$ *et* $C(b) = white$.

3.2.4 L'implémentation des gadgets radiants via les convertisseurs

Nous avons maintenant tous les outils pour obtenir facilement les implémentations parcimonieuses et radiantes du prisme, du crossover bicolore, et du crossover non-restreint.

Propriété 3.25 *Le gadget à palette montré sur la Fig. 3.19 et noté , à neuf sommets distingués* x, b, g, l_x, b, g, h_x, b, g *disposés anti-trigonométriquement dans cet ordre, est une implémentation radiante et parcimonieuse du prisme dès lors qu'une paire de référence arbitraire* (b, g) *est coloriée* $(black, gray)$

Preuve Soit C une 3-coloration de . Par propriété du crossover exclusif et par radiance des deux convertisseurs unidirectionnels , tous les sommets g on la même couleur $C(g)$, e.g., $C(g) = gray$. La radiance du convertisseur unidirectionnel de droite implique aussi que tous les sommets nommés w ont la même couleur

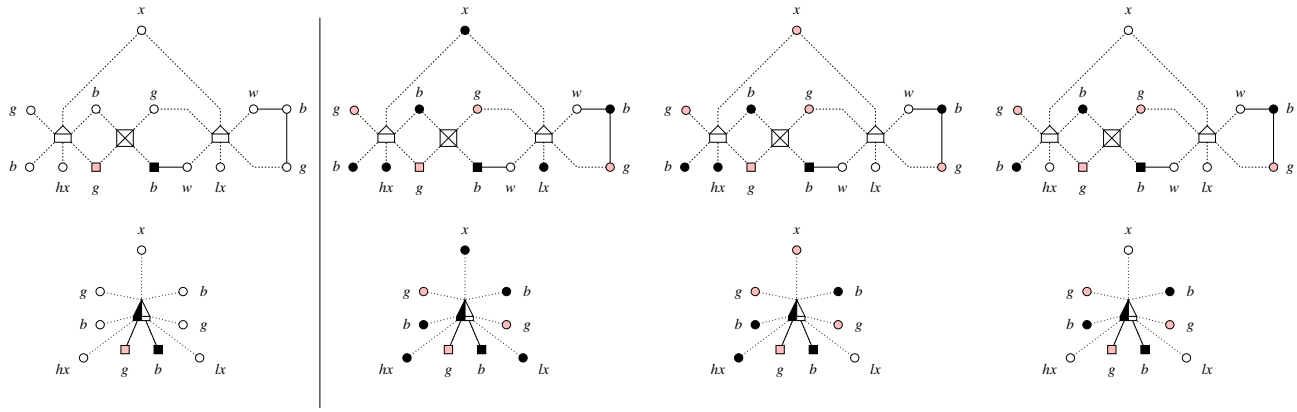


FIG. 3.19 – Un prisme radiant et parcimonieux

$C(w) \neq C(g)$, e.g. $C(w) = white$. Les deux sommets nommés b qui sont adjacents à un sommet nommé w sont aussi adjacents à un sommet nommé g (par l'intermédiaire d'un crossover exclusif pour l'un), et donc ces sommets b sont coloriés avec une troisième couleur $C(b)$ distincte de $C(w)$ et de $C(g)$, qui est *black* en suivant l'exemple. Cette couleur se propage à gauche aux autres sommets nommés b par propriété du crossover exclusif et par radiance du convertisseur unidirectionnel de gauche. Le gadget est donc radiant.

Supposons maintenant que $C(g) = gray$ et $C(b) = black$ (et donc $C(w) = white$). Alors le convertisseur unidirectionnel de gauche, resp. de droite, a ses sommets distingués secondaires portant une palette de référence sombre, resp. claire. Il se comporte donc comme un convertisseur unidirectionnel sombre, resp. clair, et il suit que $C(l_x) = low(C(x))$, resp. $C(h_x) = high(C(x))$ – voir Fig. 3.8. La parcimonie de découle de la parcimonie de et . ■

Remarque 3.26 *Les prismes étant destinés à aller deux par deux en se faisant face dans les Fig. 3.9 et 3.21, l'un des prismes d'une paire doit avoir ses sommets l_x et h_x inversés pour l'ordre anti-trigonométrique. Il est trivial de modifier la Fig. 3.19 pour obtenir l'ordre anti-trigonométrique x, b, g, h_x, b, g, l_x, b, g (il suffit de placer les sommets w sur le convertisseur gauche). On notera et on appellera* prisme miroir *un tel prisme (le côté noir dans et indique le côté du convertisseur sombre, i.e., celui de h_x).*

Propriété 3.27 *Le gadget à palette montré sur la Fig. 3.20 et noté , à douze sommets distingués x, b, g, y, b, g, x', b, g, y', b, g, disposés anti-trigonométriquement dans cet ordre, est radiant et implémente parcimonieusement un crossover bicolore dès lors qu'une paire de référence arbitraire (b, g) est coloriée $(black, gray)$.*

Preuve Soit C une 3-coloration du gadget . Par propriété du crossover exclusif et par radiance du convertisseur unidirectionnel, tous les sommets nommés g ont la même couleur $C(g)$, e.g., $C(g) = gray$. De même, tous les sommets nommés b *situés au-dessus de la ligne* (x, x') ont même couleur $C(b)$, qui est de plus distincte de $C(g)$, e.g., $C(b) = black$. Les sommets nommés w situés au-dessus de cette même ligne portent toutes la troisième couleur $C(w)$, distincte de $C(g)$ par radiance des deux convertisseurs de la lignes (x, x'), et distincte de $C(b)$ à cause des deux arêtes

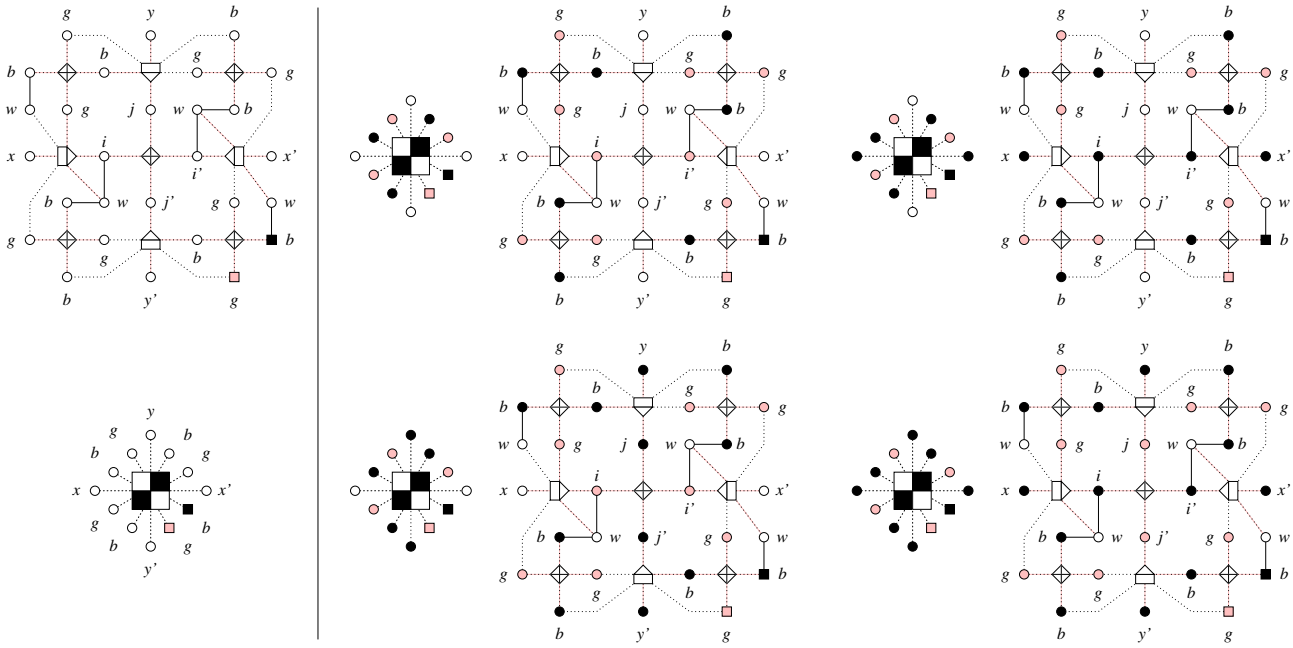


FIG. 3.20 – Un crossover bicolore radiant et parcimonieux

(b, w). En suivant l'exemple, $C(w) = white$. La couleur $C(w)$ se propage aux autres sommets nommés w situés en dessous de la ligne (x, x') par radiance du convertisseur unidirectionnel. Enfin, tous les sommets nommés b *situés en dessous de la ligne* (x, x') portent la même couleur par propriété du crossover exclusif et par radiance du convertisseur unidirectionnel. Cette couleur est distincte de $C(w)$ à cause des deux arêtes (b, w) et distincte de $C(g)$ par propriété du crossover exclusif, et il s'agit donc *black*. Le gadget est bien radiant.

Supposons maintenant que $C(b) = black$ et $C(g) = gray$. Alors $C(w) = white$ et les convertisseurs unidirectionnels situés sur l'axe (x, x'), resp. sur l'axe (y, y'), sont clairs, resp. sombres. Le gadget est donc une implémentation du schéma de construction présenté sur la Fig. 3.11 à la différence près que les arêtes (w, i) et (w, i') ont été rajoutées (afin d'obtenir un schéma parcimonieux).

Le crossover exclusif central force $C(i) = C(i') \neq C(j) = C(j')$. Les deux convertisseurs unidirectionnels sur l'axe (x, x'), resp. (y, y'), forcent $gray \neq C(x) \overset{gb}{\equiv} C(i) = C(i') \overset{gb}{\equiv} C(x') \neq gray$, resp., $gray \neq C(y) \overset{gb}{\equiv} C(j) = C(j') \overset{gb}{\equiv} C(y') \neq gray$, ce qui implique $C(x) = C(x') \neq gray$, resp. $C(y) = C(y') \neq gray$. De plus :

1. Le blanc circule toujours inchangé le long de l'axe (y, y'), i.e. $C(y) = C(y') = white \implies C(j) = C(j') = white$, car le blanc est seul dans sa classe pour l'équivalence sombre.

2.1. Le blanc est toujours converti en gris le long de l'axe (x, x'), i.e. $C(x) = C(x') = white \implies C(i) = C(i') = gray$, car $C(i) = C(i') \neq white$ à cause des arêtes (w, i) et (w, i').

2.2. Le noir circule toujours inchangé le long de l'axe (x, x'), i.e. $C(x) = C(x') = black \implies C(i) = C(i') = black$, car le noir est seul dans sa classe pour l'équivalence claire.

Les combinaisons (1+2.1) et (1+2.2) donnent lieu aux deux états locaux de la partie supérieure de la Fig. 3.20 pour $C(y) = white$. Reste le cas $C(y) = black$:

3.1. Si de plus $C(x) = white$, alors on sait que $C(i) = C(i') = gray$ par l'étude de cas 2.1, et alors $C(j) = C(j') = black$ par propriété du crossover exclusif.

3.2. Inversement, si $C(x) = black$, alors on sait que $C(i) = C(i') = black$ par l'étude de cas 2.2, et alors $C(j) = C(j') = gray$ par propriété du crossover exclusif.

Le gadget est donc bien parcimonieux, et on obtient bien les quatre configurations requises pour un crossover bicolore. ■

Propriété 3.28 *Le gadget à palette montré sur la Fig. 3.21 et noté* ⊞*, à douze sommets distingués x, b, g, y, b, g, x′, b, g, y′, b, g, disposés anti-trigonométriquement dans cet ordre, est une implémentation radiante et parcimonieuse du crossover non-restreint* ⊗*, quelle que soit la palette de référence portée par* (b, g).

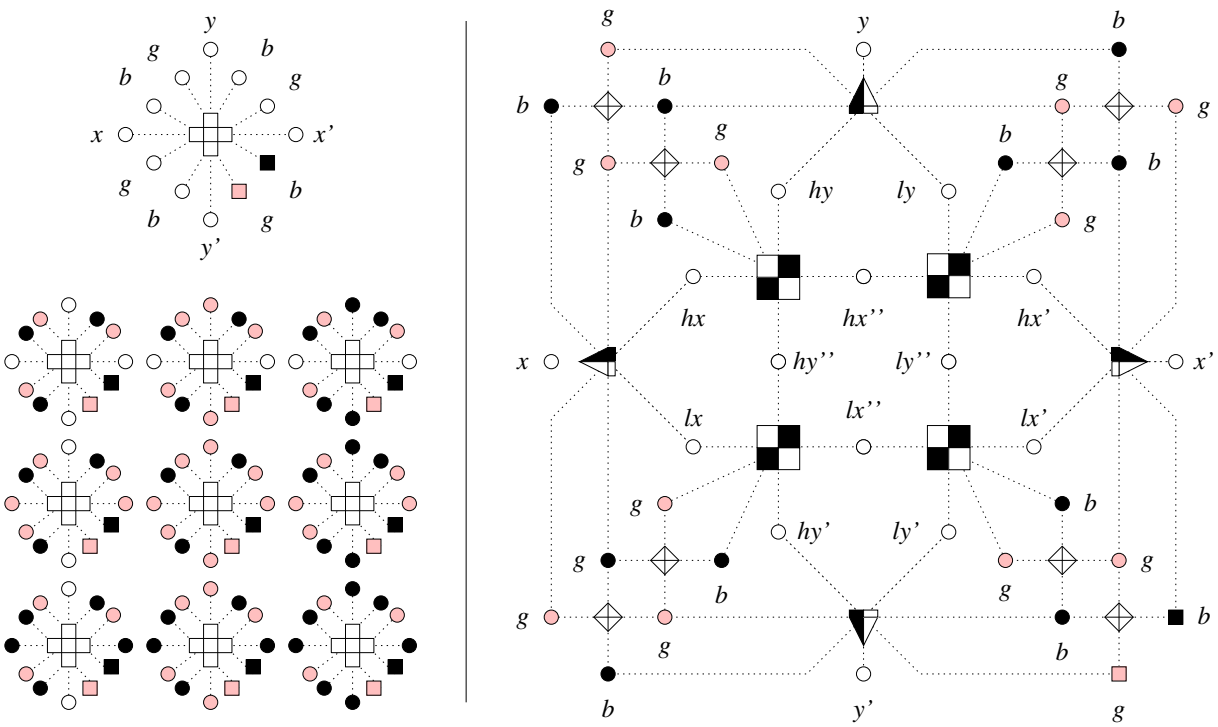


FIG. 3.21 – Un crossover non-restreint radiant et parcimonieux

Preuve Soit C une 3-coloration du gadget ⊞. Tous les sommets b, resp. g, portent la même couleur $C(b)$, resp. $C(g)$ telles que $C(b) \neq C(g)$ par propriété du crossover exclusif et par radiance des gadgets ⊤ et ⊤. Le gadget ⊞ est donc radiant.

L'ensemble des configurations primaires d'un crossover non-restreint ⊗ étant stable par permutation de couleurs, il suffit de prouver que ⊞ est une implémentation parcimonieuse de ⊗ pour une seule palette de référence : le choix d'une autre palette de référence donnera le même ensemble de configurations primaires.

Supposons donc que $(C(b), C(g)) = (black, gray)$: Les gadgets / et implémentent parcimonieusement , et , et par correction et parcimonie du schéma de construction de la Fig. 3.9, implémente parcimonieusement . ■

3.2.5 La planarisation parcimonieuse de 3-Col

Notre crossover parcimonieux étant à palette et radiant, quelques modifications au schéma de transformation de la Fig. 3.1 présentée en début de chapitre sont nécessaires :

1. D'une part, tous les crossovers doivent partager la même palette de référence, sinon le nombre de solutions serait multiplié par 6^c, où c est le nombre de crossovers posés, qui est lui-même $\Theta(|V|^2)$. Il faut donc exploiter la radiance de notre crossover pour propager la palette de voisin à voisin en utilisant un duplicateur de palette que l'on note dans le schéma de transformation modifié de la Fig. 3.22.
2. D'autre part, la palette de référence doit être entièrement déterminée par la coloration des sommets distingués primaires des crossovers, car sinon le nombre de solutions serait multiplié par 6, i.e., autant que de palettes de référence possibles. Pour cela, il suffit de choisir une arête $(v_i, v_j) \in E$, dans notre exemple (v_4, v_3), et de fusionner le sommet distingué primaire Sud du crossover $X_{|V|,j}$ avec le sommet distingué secondaire g de son secteur Sud-Est, ainsi que le sommet distingué primaire Ouest du crossover $X_{i,1}$ avec le sommet distingué secondaire b de son secteur Nord-Ouest.

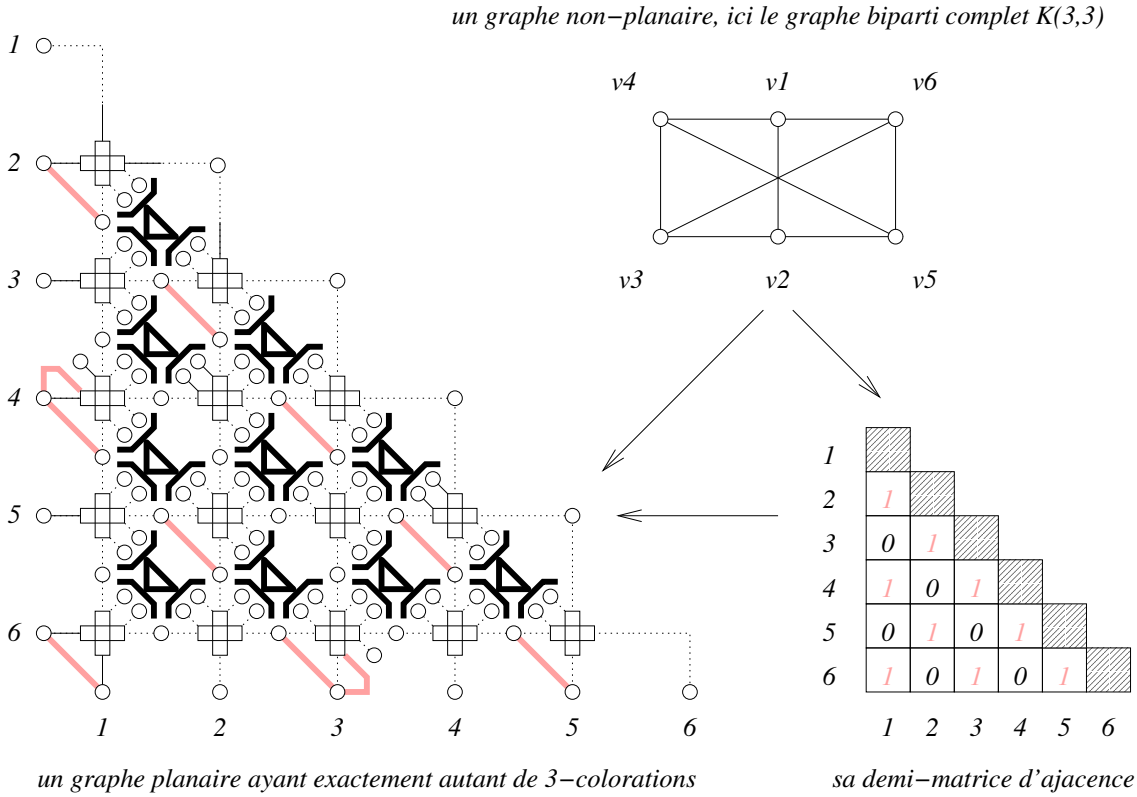


Fig. 3.22 – Transformation quadratique et parcimonieuse de 3-Col à Plan-3-Col

Le *duplicateur de paires* s'implémente très simplement comme une série de crossovers exclusifs connectés comme le montre la Fig. 3.23.

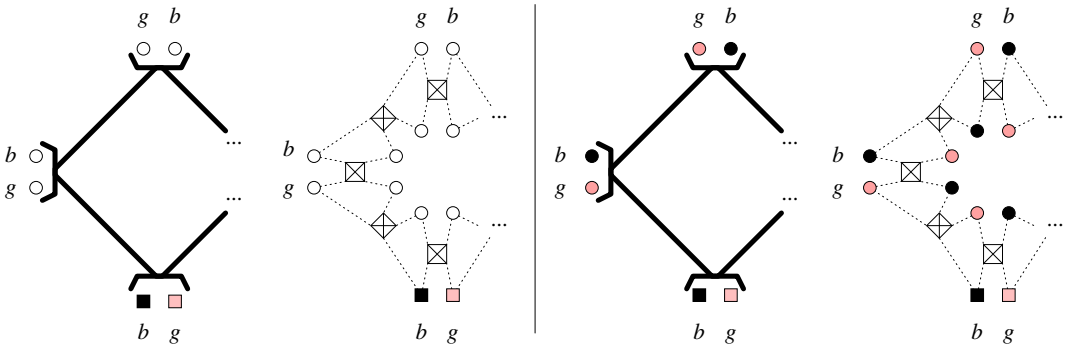


FIG. 3.23 – Un duplicateur de paires

3.3 De la satisfaisabilité à la 3-colorabilité

Les convertisseurs sombres obtenus dans la section précédente étant des ponts entre le domaine trivalué $\{white, gray, black\}$ et le domaine booléen $\{white, dark\}$, ils constituent un outil pratique pour obtenir des réductions linéaires parcimonieuses de SAT à 3-COL et de PLAN-SAT à PLAN-3-COL.

Dans cette section, nous présentons tout d'abord deux réductions de la littérature. Une réduction classique de SAT à 3-COL, linéaire mais non-parcimonieuse, ainsi qu'une réduction linéaire et faiblement parcimonieuse passant par les problèmes intermédiaires NAE-3-SAT (non-monotone) et $\frac{1}{3}$-SAT et qui multiplie le nombre de solutions par un facteur exponentiel. Nous construisons ensuite une réduction linéaire et parcimonieuse qui exploite les convertisseurs sombres en passant par le problème intermédiaire $\frac{1}{3}$-SAT. Nous la raffinerons ensuite pour qu'elle préserve le plan.

3.3.1 Réductions non ou faiblement parcimonieuses de SAT à 3-COL

La réduction non-parcimonieuse de Kozen

Soit $\varphi(L, V)$ une instance de SAT où L est une liste de clauses sur un ensemble de variables V. La réduction que l'on trouve dans le livre de Kozen [60] construit un graphe $G'(V', E')$ qui est 3-coloriable ssi φ est satisfaisable.

1. On crée tout d'abord un triangle (W, G, B) dans G', où l'on peut supposer sans perte de généralité que W, G, et B sont resp. coloriés en *white*, *gray* et *black*. On interprète le blanc et le noir comme représentant resp. les valeurs vraie et fausse, le gris ne représentant pas de valeur de vérité.
2. Pour toute variable $v \in V$, on crée un triangle $(G, pos(v), neg(v))$ où $pos(v)$ et $neg(v)$ sont les sommets associés resp. aux littéraux v et $\overline{v}$. Ainsi, les couleurs de $pos(x)$ et $neg(x)$ doivent représenter une valeur de vérité (elles ne peuvent être grises) et sont bien l'inverse l'une de l'autre.
3. Pour toute clause $c \in L$ de longueur k telle que $c = v_1 \vee \cdots v_j \vee \overline{v_{j+1}} \vee \cdots \vee \overline{v_k}$, on crée un gadget de clause tel que représenté sur la Fig. 3.24, où $l_i = pos(v_i)$ pour $i \leq j$ et $l_i = neg(v_i)$ sinon. Le sommet X à droite est B si k est impair, et W sinon.

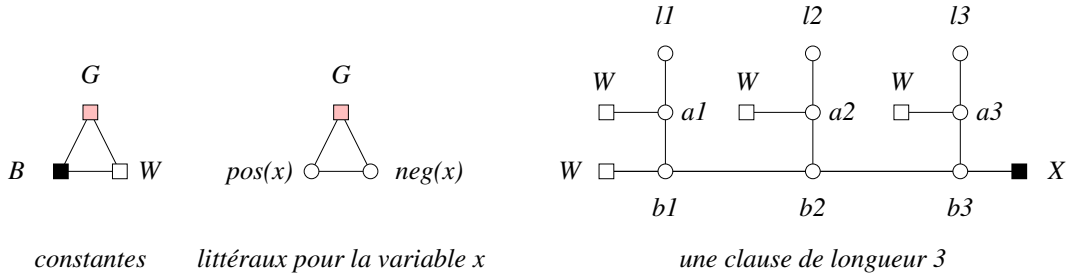


FIG. 3.24 – Une réduction non-parcimonieuse de SAT à 3-COL

L'idée est la suivante : Si tous les l_i d'un gadget de clause sont coloriés en noir (i.e., la clause associée n'est pas satisfaite) alors tous les sommets a_i sont gris, à cause du 2-chemin (W, a_i, l_i). Le sommet W à gauche de b_1 force les b_i à alterner noir et blanc, avec b_{2i+1} en noir et b_{2i} en blanc. La nature du sommet X (B ou W) est déterminée selon la parité de la longueur de la clause pour forcer l'alternance inverse : Un "clash" de couleur doit se produire.

Inversement, s'il existe un sommet l_j colorié en blanc dans tout gadget de clause (i.e., l_j témoigne de la satisfaction de la clause associée) alors on peut colorier a_j en noir et b_j en gris et colorier le reste comme décrit auparavant. L'alternance de couleurs sur les b_i est alors noir/blanc pour les i pairs/impairs à gauche de j et blanc/noir pour les i pairs/impairs à droite de j : Le sommet gris b_j fait tampon entre l'alternance de gauche et de droite et le graphe est entièrement 3-coloriable.

La source de non-parcimonie vient du fait que si une clause est satisfaite par plusieurs littéraux, on peut par exemple choisir le sommet b_j témoignant de la satisfaction de cette clause dans le schéma de coloration précédent, mais d'autres schémas sont également possibles.

La réduction faiblement parcimonieuse de Dewdney-Creignou-Hermann

La réduction de Creignou-Hermann est une transformation parcimonieuse de $\frac{1}{3}$-SAT à NAE-3-SAT (si on ne compte que les solutions non-isomorphes pour NAE-3-SAT). La réduction de Dewdney est une transformation faiblement parcimonieuse de NAE-3-SAT (non-monotone) à 3-COL avec multiplication exponentielle du nombre de solutions.

La réduction de $\frac{1}{3}$-SAT à NAE-3-SAT. La réduction parcimonieuse de $\frac{1}{3}$-SAT à NAE-3-SAT remplace simplement toutes les $\frac{1}{3}$-clauses (x, y, z) par les quatre NAE-3-clauses (x, y, z), (x, y, T), (y, T, z), (T, y, z), où T est une unique nouvelle variable dans le nouveau système. Comme on ne compte que les solutions non-isomorphes par inversion booléenne, on peut supposer que T est toujours mis à la valeur vraie. Maintenant mettre les trois variables à faux contredit la NAE-3-clause (x, y, z) (de même que les assigner toutes à vrai). En mettre au moins deux à vrai, e.g., x et y, contredit la NAE-3-clause (x, y, T). Il faut donc en mettre exactement une à vrai et on obtient les trois configurations de la Fig. 3.25.

Le réduction de NAE-3-SAT à 3-COL. La réduction faiblement parcimonieuse de NAE-3-SAT à 3-COL simule chaque variable x du système par un triangle $(neg(x), pos(x), G)$, où $pos(x)$, resp. $neg(x)$, représente le littéral x, resp. $\overline{x}$, et où G est un sommet commun à tous les triangles. A chaque NAE-3-clause (l_x, l_y, l_z) du système où l_x, l_y et l_z sont des littéraux de variables respectives x, y, et z, on associe un triangle $(occ(x), occ(y), occ(z))$ où $occ(x)$, $occ(y)$ et $occ(z)$ sont des

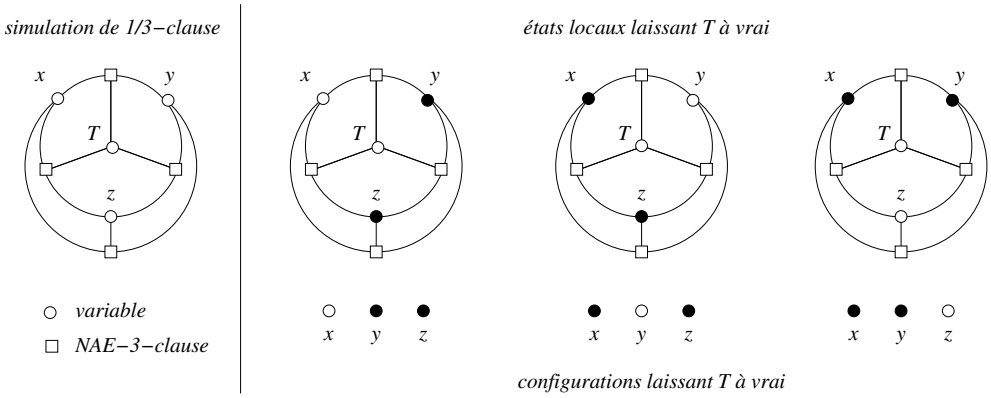


FIG. 3.25 – Simulation d'une $\frac{1}{3}$-clause (x, y, z) par quatre NAE-3-clauses

nouveaux sommets associés aux occurrences respectives des littéraux l_x, l_y et l_z dans la NAE-3-clause, et on connecte par arête tout sommet $occ(x)$ à $pos(x)$, resp. $neg(x)$, si l_x est négatif, resp. positif, comme indiqué sur la Fig. 3.26.

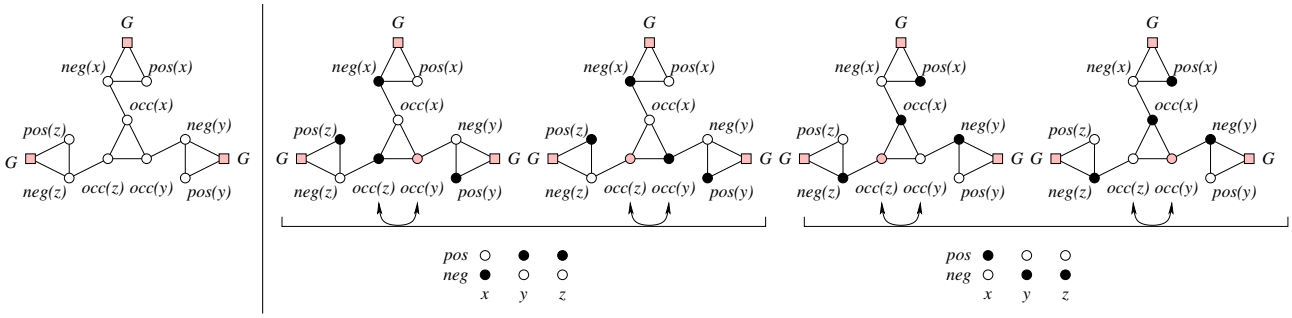


FIG. 3.26 – Simulation d'une NAE-3-clause (x, y, z) avec 3-COL

Le noir et le blanc représentent resp. les valeurs booléennes fausse et vraie, le gris représentant une valeur booléenne libre. Comme on ne compte pas les 3-colorations isomorphes, on peut supposer que pour toute coloration C, $C(G) = gray$ et $C(pos(t)) = white$ pour une variable t arbitrairement choisie : Colorier G en gris implique que $C(pos(x))$ représente une valeur booléenne pour toute variable x et que $C(neg(x))$ représente la valeur opposée. Colorier de plus t en blanc implique que tous les assignements booléens simulés sont non-isomorphes par négation booléenne. Maintenant tout triangle $(occ(x), occ(y), occ(z))$ associé à une NAE-3-clause (l_x, l_y, l_z) doit distribuer les trois couleurs sur ses trois sommets, e.g., $C(occ(x)) = white$, $C(occ(y)) = gray$, et $C(occ(z)) = black$, impliquant à leur tour $C(pos(x)) = white$, $C(neg(x)) = black$ et $C(pos(z)) = black$, $C(neg(z)) = white$ si l_x, l_y, l_z sont tous positifs. $C(occ(y)) = gray$ implique que $(C(pos(y)), C(neg(y)))$ est librement $(black, white)$ ou $(white, black)$. Les deux littéraux l_x et l_z portent donc des valeurs booléennes différentes et l_y reste libre dans l'assignement simulé et la NAE-3-clause peut donc être satisfaite de toute les manières possibles (i.e., par deux occurrences négatives et une positive ou vice-versa).

La non-parcimonie de la réduction vient du fait que pour une 3-coloration donnée des sommets de littéraux $pos(x), pos(y)$ et $pos(z)$ d'une même clause, il existe deux façons de colorier le triangle de cette clause $(occ(x), occ(y), occ(z))$, la seconde coloration étant obtenue à partir de la première en permutant dans le triangle le gris et le noir si deux sommets de littéraux sont noirs, et en

permutant dans ce même triangle le gris et le blanc si deux sommets de littéraux sont blancs. Les colorations de chaque triangle pouvant varier indépendamment, chaque solution d'un système de m NAE-3-clauses est représentée par 2^m colorations non-isomorphes.

3.3.2 La réduction parcimonieuse de $\frac{1}{3}$-SAT à 3-COL

Dans notre réduction parcimonieuse de $\frac{1}{3}$-SAT à 3-COL, nous allons interpréter le blanc comme la valeur booléenne vraie, et les deux couleurs sombres comme la valeur booléenne fausse, le noir en étant cependant la représentation canonique pour les sommets associés aux variables, le gris n'étant utilisé que dans les gadgets simulant les $\frac{1}{3}$-clauses.

Avec cette convention, un simple triangle (x, y, z) simule déjà une $\frac{1}{3}$-clause puisque les trois couleurs doivent être y distribuées, la blanche y apparaissant exactement une fois. Cette simulation est non-parcimonieuse car les deux couleurs sombres peuvent permuter et toujours représenter le même assignement, mais cela est trivialement corrigé en connectant x à un sommet gris et y à un sommet noir, de sorte que les trois configurations possibles pour le triangle (x, y, z) sont $(white, gray, black)$, $(black, white, gray)$ et $(black, gray, white)$, le blanc pouvant se placer sur chacun des trois sommets.

Les représentations canoniques des valeurs booléennes sont naturellement obtenues en connectant un convertisseur unidirectionnel sombre aux seuls sommets pouvant être éventuellement gris, c'est-à-dire y et z. On obtient :

Propriété 3.29 *Le gadget à palette de la Fig.3.27, noté*  *, qui a* $3 + 2 \times 3$ *sommets distingués* $x, b, g, y, b, g, z, b, g$*, disposés anti-trigonométriquement dans cet ordre, est radiant et simule parcimonieusement une* $\frac{1}{3}$*-clause* (x, y, z)*, i.e., n'admet que les trois configurations primaires* $(white, black, black)$*,* $(black, white, black)$*,* $(black, black, white)$*, dès lors qu'une paire de référence arbitraire* (b, g) *est coloriée* $(black, gray)$*.*

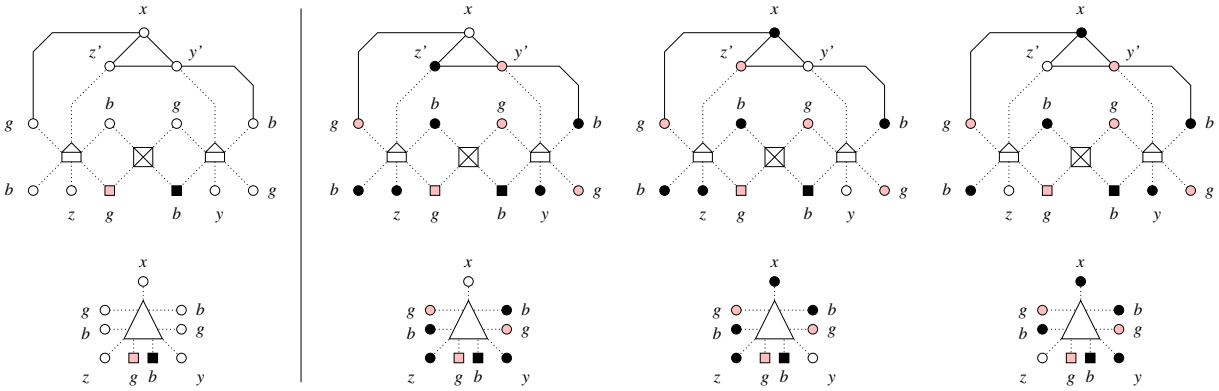


FIG. 3.27 – Un simulateur parcimonieux et radiant de la $\frac{1}{3}$-clause (x, y, z)

Comme toujours, la radiance du gadget est une conséquence de la propriété du crossover exclusif et de la radiance des convertisseurs unidirectionnels. La réduction parcimonieuse de $\frac{1}{3}$-SAT à 3-COL consiste simplement à associer un sommet v_x à chaque variable x de l'instance $\frac{1}{3}$-SAT φ et à associer un simulateur de clause (v_x, v_y, v_z) à chaque $\frac{1}{3}$-clause de φ. Si l'instance n'est pas planaire ou si l'on ne souhaite pas préserver sa planarité, il suffit ensuite de fusionner en un seul sommet tous les sommet g – resp. tous les sommets b – des paires de référence.

Si l'instance est planaire, i.e., si le graphe biparti G_φ associé à φ est planaire et que l'on souhaite préserver cette planarité, i.e., réduire PLAN-$\frac{1}{3}$-SAT à PLAN-3-COL, alors, en supposant

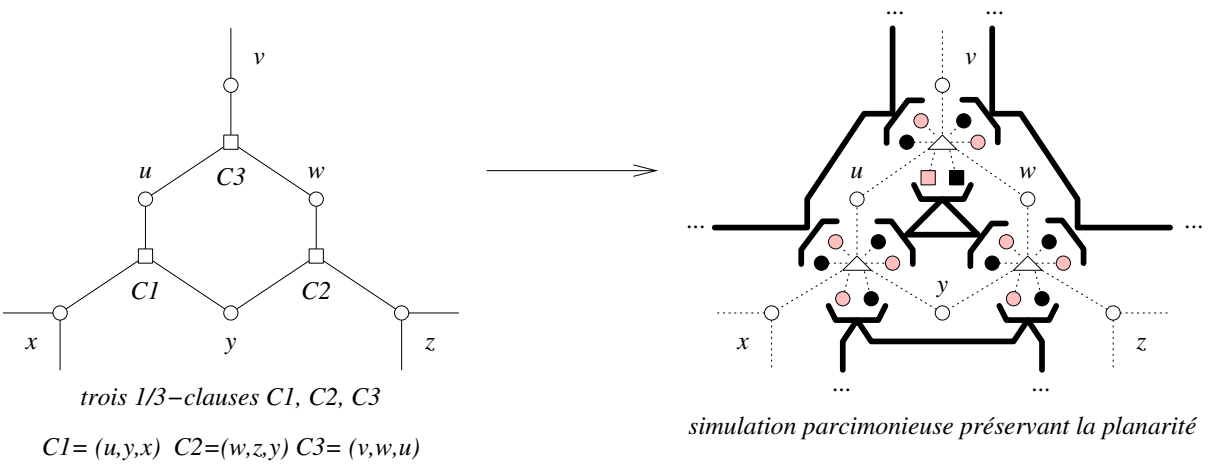


FIG. 3.28 – La réduction parcimonieuse de PLAN-$\frac{1}{3}$-SAT à PLAN-3-COL

G_φ connexe, il suffit de placer les simulateurs de clauses en respectant l'ordre anti-trigonométrique des sommets de variables autour des sommets de clauses pour le plongement planaire choisi pour G_φ, et connecter toutes les paires de référence situées dans une même face à un duplicateur de palette, comme montré sur la Fig. 3.28. La correction et la parcimonie découlent de la propriété des simulateurs de $\frac{1}{3}$-clauses et de leur radiance.

3.4 Conclusion

Dans ce chapitre, nous avons exhibé les réductions :
- $\frac{1}{3}$-SAT$\leq$ 3-COL, linéaire et parcimonieuse,
- PLAN-$\frac{1}{3}$-SAT$\leq$ PLAN-3-COL, linéaire et parcimonieuse,
- 3-COL$\leq$ PLAN-3-COL, quadratique et parcimonieuse.

Nous ne nous sommes jamais préoccupé d'optimiser le *degré maximal* des graphes obtenus par nos réductions. Notons que toute réduction non-parcimonieuse, resp. faiblement parcimonieuse, vers 3-COL ou PLAN-3-COL peut toujours se réécrire en une réduction non-parcimonieuse, resp. faiblement parcimonieuse, telle que les graphes obtenus soient de degré maximal 4. En effet, il est facile d'exploser un sommet de degré $d > 4$ en d copies de degré au plus 4.

Un gadget évident pour réaliser cette duplication est celui de la Fig. 3.29. On voit aisément que les trois triangles (x_i, y, z), $1 \leq i \leq 3$, forcent les couleurs $C(x_i)$ à être identiques. Le gadget est cependant source de parcimonie faible car les couleurs $C(y)$ et $C(z)$ peuvent permuter pour une même configuration monochrome (cf. Fig. 3.29 en haut à droite), créant ainsi deux états locaux par configuration. Notons que y et z sont de degré 4, et que les x_i sont de degré 2, de telle sorte que l'on peut monter en série plusieurs instances du gadget, comme illustré sur la Fig. 3.29 en bas à droite, sans jamais avoir un sommet de degré > 4 : ici, un sommet est explosé en sept copies $x_1, \cdots, x_7$, toutes de degré 2, grâce à cinq gadgets de copie et quatre sommets intermédiaires a, b, c, d qui tous de degré $2 + 2 = 4$.

Le gadget n'est pas planaire en l'état, mais peut le devenir (voir Fig. 3.30) en plongeant x_3 à l'intérieur de la face (y, z, x_2) et en en faisant une copie x_3' par la pose d'un crossover exclusif ⊕ sur l'arête (z, x_2). Ceci a la fâcheuse conséquence de faire passer le degré de z à 5, et le degré de x_3' à 3, mais comme l'une des arêtes extérieures du crossover exclusif ⊕ est inutile (hormis pour l'esthétique), les degrés de z et x_3' redescendent resp. à 4 et 2 lorsqu'on la supprime. Le

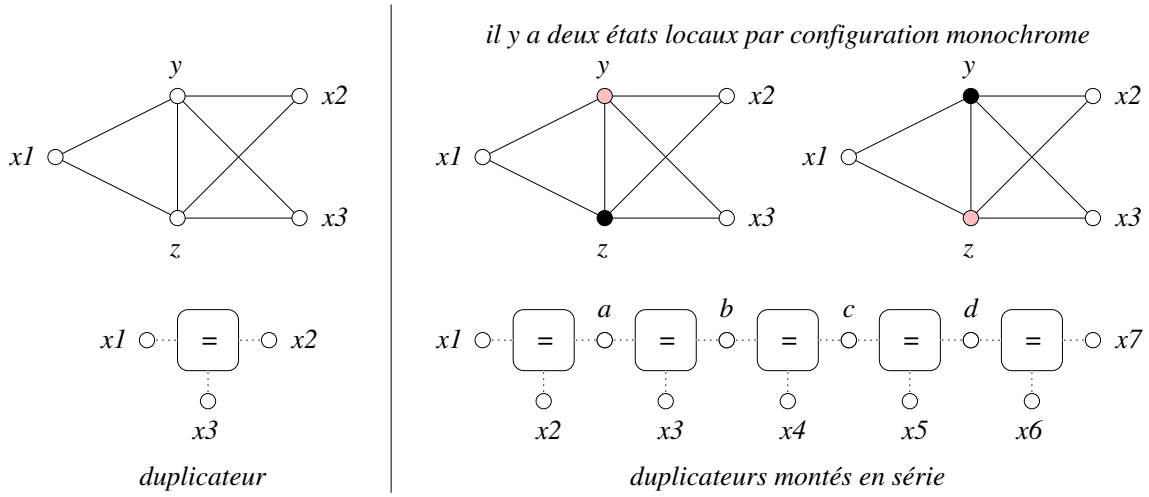


FIG. 3.29 – Comment obtenir des sommets de degré maximal 4

crossover exclusif ⊕ étant parcimonieux, le nombre d'état locaux par configuration monochrome du gadget planaire résultant est toujours de 2.

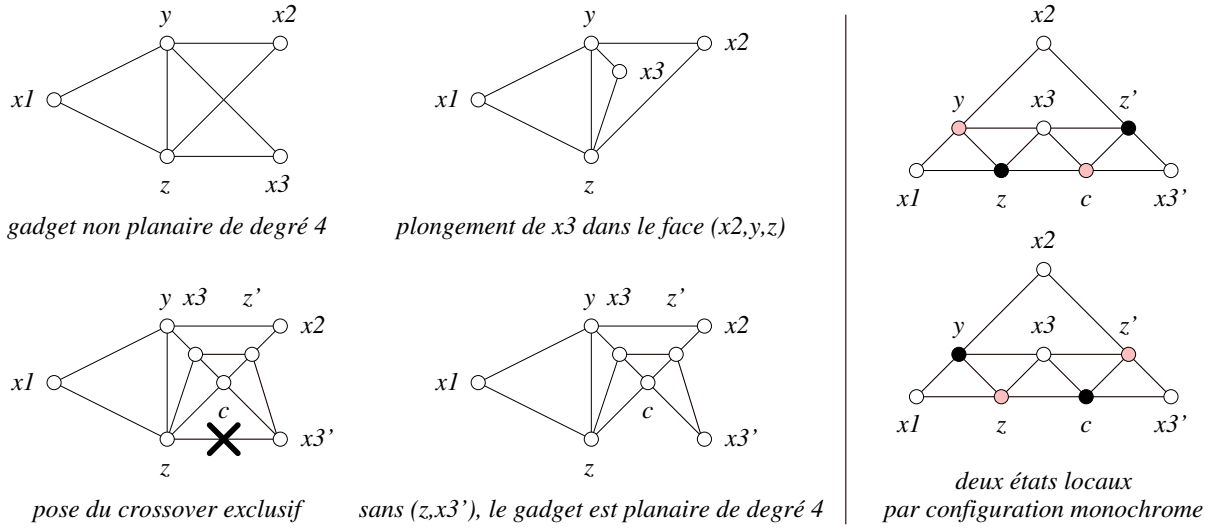


FIG. 3.30 – Planarisation du gadget de copie en préservant le degré

Notons que l'on ne peut descendre le degré maximal en dessous de 4. En effet, le théorème de Brooks affirme que :

Théorème 3.30 (Brooks) *Tout graphe connexe G de degré maximal $k \geq 3$ est k-coloriable ssi G n'est pas la $(k+1)$-clique K_{k+1}. De plus, si G est k-coloriable, un k-coloriage est calculable en temps linéaire.*

Par conséquent, sous l'hypothèse P $\neq$ NP, il n'existe pas de réduction polynomiale d'un problème NP-complet à la 3-colorabilité des graphes de degré maximal 3, et par conséquent un degré 4 est optimal. Nos réductions parcimonieuses produisent des graphes de degré largement supérieur à 4, et l'on peut se demander quel est l'entier minimal k tel que SAT se réduise parcimonieusement à 3-COL sur les graphes de degré maximal k. Nous laissons cette question ouverte,

de même que la question analogue pour les versions planaires de ces deux problèmes. Nous terminons ce chapitre par la preuve du Théorème de Brooks. Elle est issue de l'application fine de l'algorithme de coloriage "naïf" suivant :

Algorithme 3.31 (coloriage naïf)
- *Entrée : Un graphe $G(V, E)$ de degré maximal k et un ordre arbitraire $v_1, \cdots, v_n$ de V.*
- *Sortie : Un $(k+1)$-coloriage de V.*
- *Méthode : Colorier les v_i dans l'ordre, en choisissant pour v_i une couleur arbitraire parmi les couleurs différentes de celles des voisins v_j de v_i, $j < i$.*

L'algorithme peut toujours $k+1$-colorier G.

> **Preuve** A toute étape i, une couleur pour v_i est trivialement toujours disponible puisque nous disposons d'une palette de $(k+1)$ couleurs et que v_i a au plus k voisins. ■

L'idée est de montrer que cet algorithme naïf est plus performant (i.e., qu'il n'utilisera que k couleurs) pour un ordre bien choisi sur les sommets et si le graphe vérifie une bonne propriété, comme c'est presque toujours le cas : Intuitivement, si l'on couvre un graphe de degré k par un arbre (de racine r) et que l'on choisit de colorier les sommets selon un ordre "bottom-up" le long de cet arbre, alors il n'y aura jamais de conflit lors du coloriage d'une feuille ou d'un noeud interne. Le seul conflit possible se situe donc à la racine r, et il peut être évité si deux sommets x et y adjacents de r ont été coloriés identiquement. Il faut donc s'assurer de l'existence d'un tel *bon triplet* (x, y, r) :

Définition 3.32 (Bon triplet) *Dans un graphe connexe $G(V, E)$, un* bon triplet (x, r, y) *est un triplet de trois sommets $x, y, r \in V$ tels que :*
- *$(r, x) \in E$ et $(r, y) \in E$, et*
- *$(x, y) \notin E$, et*
- *$G \setminus \{x, y\}$ reste connexe.*

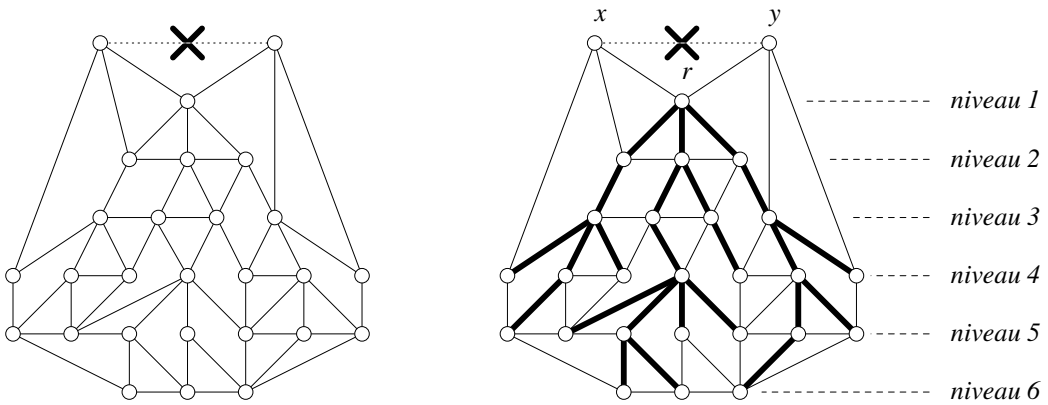


FIG. 3.31 – Exploitation d'un bon triplet dans un graphe

Lemme 3.33 *Soit $G(V, E)$ un graphe connexe de degré maximal $k \geq 3$ possédant un bon triplet (x, r, y), et muni d'un arbre couvrant T de racine r dans le sous-graphe $G \setminus \{x, y\}$. Soit $v_1, \cdots, v_n$ un ordre sur V tel que :*

- $v_1 = x$, $v_2 = y$, $v_n = r$, *et*
- $v_3, \cdots, v_n$ *est un ordre postfixe sur* T.

alors, l'algorithme naïf 3.31 peut toujours k-colorier G pour cet ordre.

Preuve Vérifions que ce coloriage "bottom-up" le long de T peut toujours utiliser k couleurs seulement :
- A l'étape $i = 2$, l'algorithme peut colorier $v_2 = y$ avec la même couleur que $v_1 = x$ puisque $(x, y) \notin E$.
- Ensuite, pour toute étape $2 < i < n$, au moins un sommet voisin de v_i, son père dans T, n'est pas encore colorié. Il y a donc au plus $k-1$ voisins de v_i déjà coloriés et l'algorithme peut donc colorier v_i en s'interdisant la $(k+1)^{eme}$ couleur.
- Enfin, à l'étape $i = n$ (i.e., $v_i = r$), il reste à colorier r qui a au plus k voisins dont x et y. Comme x et y sont coloriés avec la même couleur, les voisins de r n'utilisent au maximum que $k-1$ couleurs, et l'algorithme peut colorier r en s'interdisant la $(k+1)^{eme}$ couleur.

■

Les trois lemmes suivants montrent qu'on peut toujours trouver un bon triplet dans un graphe G connexe non-complet de degré k (ou dans un petit surgraphe), et ainsi k-colorier G.

Lemme 3.34 *Tout graphe* $G(V, E)$*, triconnexe mais non-complet, admet un bon triplet* (x, r, y).

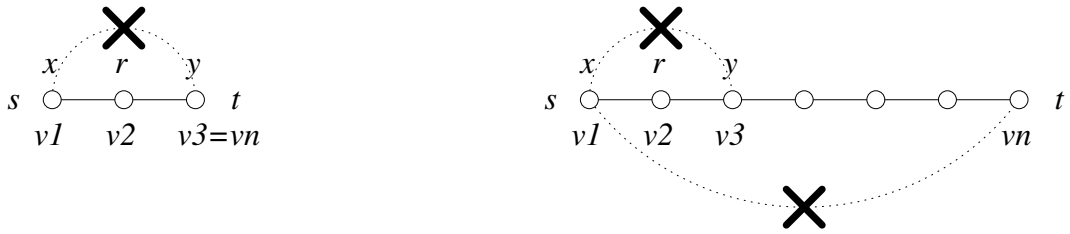


FIG. 3.32 – Trouver un bon triplet (v_1, v_2, v_3) dans un graphe triconnexe non-complet

Preuve Voir Fig. 3.32 pour l'intuition : G n'est pas complet donc il existe deux sommets $s, t \in V$ tels que $(s, t) \notin E$. Soit P le plus court (s, t)-chemin $s = v_1, v_2, v_3, \cdots, v_n = t$. Notons que $n \geq 3$, et que de plus $(v_1, v_3) \notin E$ car sinon P n'est pas un plus court (s, t)-chemin. Par conséquent, on peut choisir (v_1, v_2, v_3) comme bon triplet : Comme G est triconnexe, le sous-graphe $G \setminus \{x, y\}$ reste bien connexe. ■

Lemme 3.35 *Tout graphe* $G(V, E)$ *de degré* $k \geq 3$*, biconnexe mais non-triconnexe, possède un bon triplet* (x, r, y) *ou en possède un après l'ajout d'une arête* $e \notin E$ *ne changeant pas le degré maximal de* G.

Preuve Voir Fig. 3.33 pour l'intuition :
- Comme G n'est pas triconnexe, il existe deux sommets $s, t \in V$ tels que le sous-graphe $G \setminus \{s, t\}$ n'est pas connexe. Tous deux sont de degré au moins 2, et on peut toujours supposer que l'un d'eux, disons s, a un degré ≥ 3 : En effet, si s et t sont tous deux de degré 2, on peut augmenter E par l'arête $e = (s, t) \notin E$ sans augmenter le degré $k \geq 3$ de G car s et t sont alors de degré 3.

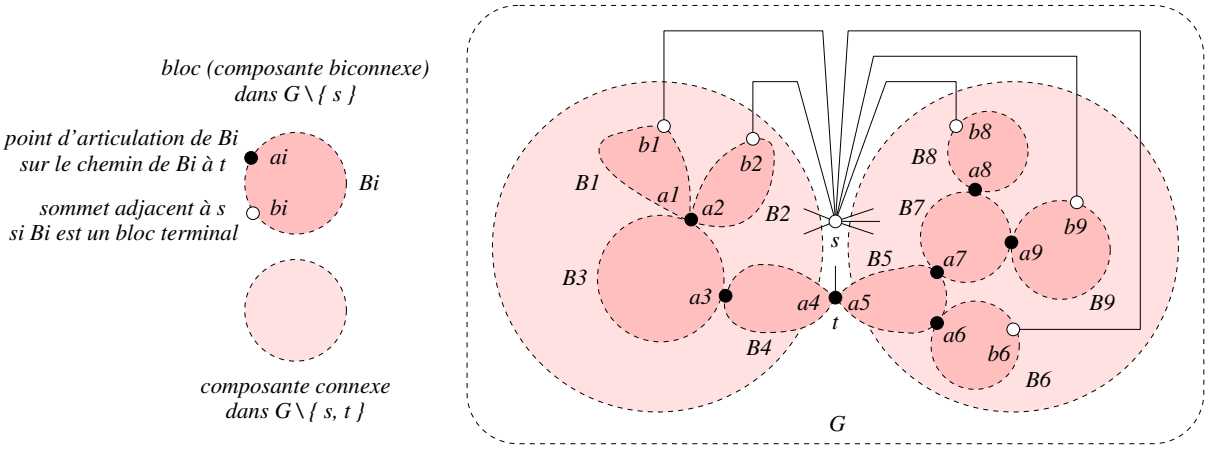


FIG. 3.33 – Trouver un bon triplet (b_i, s, b_j) dans un graphe biconnexe non-triconnexe

- Comme G est biconnexe, $G \setminus \{s\}$ est connexe mais non-biconnexe, car t en est un point d'articulation. Notons T l'arbre des blocs (composantes biconnexes) $B_1, \cdots, B_n$ de $G \setminus \{s\}$ et a_i le point d'articulation de B_i se trouvant sur l'unique chemin reliant B_i à t dans T :
- Il doit exister une arête (s, b_i) telle que $b_i \in B_i$ pour tout bloc terminal B_i (feuille de l'arbre T) et telle que b_i n'est pas un point d'articulation de $G \setminus \{s\}$. En effet, si un bloc terminal B_i n'avait pas une telle arête, le point d'articulation a_i dans $G \setminus \{s\}$ serait aussi un point d'articulation dans G, qui est biconnexe, une contradiction.

Soient deux sommets b_i et b_j dans deux blocs terminaux distincts tel que $(s, b_i) \in E$ et $(s, b_j) \in E$. (b_i, s, b_j) est un bon triplet. En effet, les blocs étant distincts, on a bien $(b_i, b_j) \notin E$. De plus $G \setminus \{b_i, b_j\}$ reste connexe car :

- $G \setminus \{b_i, s, b_j\}$ est connexe. En effet : $G \setminus \{s\}$ est connexe, et b_i et b_j appartiennent à deux blocs terminaux distincts et ne sont pas des points d'articulation dans $G \setminus \{s\}$, donc leur suppression ne déconnecte pas $G \setminus \{b_i, s, b_j\}$.
- s étant de degré ≥ 3, s est nécessairement connecté à $G \setminus \{b_i, s, b_j\}$ par au moins une troisième arête distincte de (s, b_i) et de (s, b_j).

■

Lemme 3.36 *Soit $k \geq 3$. Tout graphe $G(V, E)$ connexe mais non-biconnexe est k-coloriable ssi chacune de ses composantes biconnexes (ou blocs) est k-coloriable.*

Preuve L'implication ($\Longrightarrow$) est triviale. Montrons l'implication ($\Longleftarrow$) : Soit G un graphe non-biconnexe dont les blocs $B_1, \cdots B_n$ ont les k-coloriages respectifs $C_1, \cdots, C_k$. Soit T l'arbre des blocs de G : deux blocs voisins dans T ont toujours exactement un point d'articulation en commun. Soit r une racine choisie arbitrairement dans T. On k-colorie G en conservant C_1 pour le bloc r et en coloriant le reste de façon "top-down" par un parcours de T en largeur : Lors de la visite d'un bloc B_i de père B_j, l'unique conflit potentiel entre C_i et C_j apparaît en leur point d'articulation commun a. Si $C_i(a) \neq C_j(a)$, permuter les couleurs de C_j de telle façon que $C_i(a) = C_j(a)$. Les seuls nouveaux conflits potentiels apparaissent avec les fils de B_i qui seront visités plus tard lors du parcours en largeur. ■

Des trois lemmes précédents découle le théorème de Brooks :

Preuve du Théorème 3.30 (Brooks) Si G est la $(k+1)$-clique, G est trivialement non-k-coloriable. Pour la réciproque : Soit $G \neq K_{k+1}$ un graphe connexe de degré $k \geq 3$. Pour toute composante biconnexe G' de G :

- Si G' est complet alors, comme $G \neq K_{k+1}$, $G' = K_{k'}$, pour $k' < k$, et donc G est k-coloriable par l'algorithme naïf 3.31 pour un ordre quelconque sur V.
- Si G' est triconnexe mais non-complet, alors le Lemme 3.34 s'applique à G', et par le lemme 3.33, on peut donc k-colorier G'.
- Si G' n'est pas triconnexe, alors le Lemme 3.35 s'applique à G', et par le lemme 3.33, on peut donc k-colorier G'.

Par le Lemme 3.36, on obtient un k-coloriage de G à partir des k-coloriages de tous ses blocs G'. ■

4

Réductions linéaires

ou la préservation de la complexité des problèmes

Sommaire

4.1 Introduction 64
4.2 De la satisfaisabilité planaire à l'Hamiltonicité planaire 66
4.2.1 La réduction quadratique de Plan-$\frac{1}{n}$-Sat à Plan-UHam-Cycle . . . 66
La version linéaire mais non-planaire 68
Une version planaire mais quadratique 70
4.2.2 Vers une réduction planaire et linéaire de Sat à UHam-Cycle 72
4.2.3 Des gadgets Hamiltoniens logiques et fonctionnels 74
4.2.4 Le flip-flap-flop 77
4.3 De l'Hamiltonicité planaire à la satisfaisabilité planaire 80
4.3.1 Une caractérisation locale de l'Hamiltonicité planaire 82
4.3.2 La réduction linéaire de Plan-UHam-Cycle à Plan-Sat 86
4.3.3 Implémentation des contraintes 1/N et 2/N dans le plan 87
4.4 Sat et les problèmes de cardinalité 89

4.1 Introduction

Ce chapitre réunit les réductions obtenues lors de cette thèse pour lesquelles le caractère linéaire nous a paru difficile à obtenir. Ce sont les deux réductions qui établissent *l'équivalence linéaire et parcimonieuse* de l'Hamiltonicité et de la Satisfaisabilité *dans le plan*. Nous y ajoutons également deux réductions faciles établissant l'équivalence linéaire (et parcimonieuse) de Vertex-Cover et Sat.

La difficulté de réduire linéairement Hamilton à Sat vient du fait que le problème Hamilton fait intervenir dans son énoncé une contrainte de *connexité globale* sur la solution. En effet, on peut caractériser une solution, c'est-à-dire un cycle Hamiltonien comme suit : Premièrement, chaque sommet du graphe doit avoir exactement deux arêtes dans l'ensemble solution, et deuxièmement, l'ensemble des arêtes de cette solution doit former un ensemble connexe. La première contrainte est locale à chaque sommet, alors que la deuxième est globale sur la totalité du graphe. Ce dernier aspect est évidemment absent dans le problème Sat : Dans une solution vue comme un coloriage du graphe biparti associé à la formule, tout sommet de clause doit avoir au moins un sommet de variable à vrai et adjacent par arête étiquetée positivement, ou au moins un sommet de variable à faux et adjacent par arête étiquetée négativement.

Cette contrainte de connexité sur la solution pour Hamilton et absente dans Sat nous empêche de trouver une réduction linéaire de Hamilton à Sat dans le cas général parce que cela revient à deviner un ordre cyclique sur les n sommets. Pour cela, on ne sait pas faire mieux que deviner $O(\log n)$ variables booléennes par sommet qui codent en binaire le rang du sommet dans l'ordre deviné, fixer le rang d'un sommet arbitraire à 0 et – s'il l'on choisit le problème du circuit Hamiltonien dans un graphe orienté – coder pour tout sommet que celui-ci possède exactement un arc rentrant et un arc sortant participant à la solution, et enfin coder que chacun des m arcs doit forcer le rang de son sommet d'arrivée à suivre immédiatement le rang de son sommet de départ si cet arc participe à la solution. Une telle réduction est parcimonieuse, mais est de complexité $O((n+m)\log n)$.

Nous pensons qu'aucune réduction linéaire n'existe de Hamilton à Sat. Cependant, nous ne pouvons pas raisonnablement espérer le prouver, car sinon, on aurait immédiatement Hamilton $\notin$ DLIN (tous les problèmes de DLIN étant trivialement DLIN-réductibles à Sat), et on aurait donc une borne basse *non-linéaire prouvée* pour Hamilton.

Nous montrerons à l'inverse que Hamilton se réduit bien à Sat linéairement et parcimonieusement dans le cas très particulier des instances planaires. Nous choisirons pour cela la variante du cycle Hamiltonien dans les graphes non-orientés : Plan-UHam-Cycle.

Proposition 4.1 Plan-UHam-Cycle *est parcimonieusement réductible à* Plan-Sat *en temps linéaire.*

Ce résultat est surprenant, et montre que le problème Hamilton, *n'est plus*, malgré son apparence, un problème de connexité dans le plan, et peut être ramené à un simple *problème de coloriage* avec *contraintes locales*. A ce titre, les mêmes stratégies de diviser-pour-régner utilisées pour Sat peuvent s'appliquer pour résoudre Plan-Hamilton en temps sous-exponentiel $2^{O(\sqrt{n})}$. Ici curieusement, c'est le plan qui nous aide à construire notre réduction linéaire alors que d'habitude, il constitue un obstacle à la linéarité. Nous ne sommes d'ailleurs pas capable de généraliser notre résultat à d'autres surfaces, comme le tore.

En revanche, dans la réduction réciproque, le plan joue à nouveau son rôle de trouble-fête, et nous aurons besoin de crossovers pour Plan-Hamilton, comme on l'a déjà vu pour 3-Col dans le chapitre précédent, et pour $\frac{1}{3}$-Sat dans les préliminaires. Un crossover existe déja pour Plan-Hamilton dans la littérature, mais, comme il sera exposé dans ce chapitre, son utilisation donne

lieu à une réduction quadratique de Plan-Sat à Plan-Hamilton. Nous aurons donc besoin d'un autre type de crossover, dont nous spécifierons les propriétés et que nous construirons, pour finalement établir que :

Proposition 4.2 Plan-Sat *est parcimonieusement réductible à* Plan-UHam-Cycle *en temps linéaire.*

Il est remarquable que dans chacune de nos deux réductions, le graphe dual de l'instance de départ sera exploité : Pour la réduction de Plan-UHam-Cycle à Plan-Sat, c'est la façon dont le complémentaire du dual du cycle Hamiltonnien se structure qui sera la clé de la réduction, et pour la réduction de Plan-Sat à Plan-UHam-Cycle, c'est un arbre couvrant dans le graphe dual du graphe de la formule qui formera la colonne vertébrale du cycle Hamiltonien que l'on construira. Ces deux réductions nous permettent d'obtenir le résultat suivant dans le plan, qui contraste avec le cas général :

Corollaire 4.3 Plan-UHam-Cycle *et* Plan-Sat *sont équivalents sous réductions parcimonieuses et linéaires.*

On peut se demander si cette équivalence tient au fait qu'on a choisi comme variante du Problème Hamilton celle du cycle Hamiltonien dans un graphe non-orienté, plutôt qu'une autre comme, par exemple, celle du problème du chemin Hamiltonien à extrémités non-fixées dans un graphe orienté. Il n'en est rien : Dans le chapitre suivant, nous établirons qu'un grand nombre de variantes du problème Hamilton sont linéairement et parcimonieusement équivalentes dans le plan, et donc également équivalentes à Plan-Sat. S'il est très facile d'établir les équivalences linéaires des différentes variantes de Hamilton dans le cas non-planaire, nous verrons dans le prochain chapitre qu'il est plus pénible d'établir ces équivalences dans le plan. Le présent chapitre aura cependant fourni tous les gadgets de base nécessaires aux réductions sous-jacentes.

Ce chapitre se terminera sur les deux réductions simples établissant que :

Proposition 4.4 Sat *et* Vertex-Cover *sont linéairement et parcimonieusement équivalents.*

Ceci contredit l'intuition développée dans [19] selon laquelle ni Vertex-Cover (ni aucun problème combinant des contraintes locales avec une contrainte globale de cardinalité) n'est réductible linéairement à Sat. En fait, le simple additionneur de taille linéaire que nous construirons montre que tous ces problèmes, tels Dominating-Set ou Max-Sat sont linéairement et parcimonieusment réductibles à Sat. L'intuition semble cependant vraie dans le plan : il semble difficile de réduire Plan-Vertex-Cover à Plan-Sat, bien que ces deux problèmes aient des algorithmes analogues et sous-exponentiels en $2^{O(\sqrt{n})}$, par application du théorème des séparateurs planaires de Lipton et Tarjan [65, 60]. La réduction réciproque utilise $\frac{1}{3}$-Sat comme problème intermédiaire. Cette réduction linéaire et parcimonieuse de $\frac{1}{3}$-Sat à Vertex-Cover est également planaire, et montre que :

Proposition 4.5 Plan-Sat *se réduit linéairement et parcimonieusment à* Plan-Vertex-Cover.

Ce résultat améliore la preuve de #P-complétude de Hunt et al., obtenue par une réduction analogue mais faiblement parcimonieuse, avec multiplication du nombre de solution par un facteur exponentiel, et prouve également que :

Corollaire 4.6 Unique-Plan-Vertex-Cover *est DP-complet sous réductions polynomiales aléatoires.*

4.2 De la satisfaisabilité planaire à l'Hamiltonicité planaire

Une réduction classique de la satisfaisabilité à l'Hamiltonicité est celle présentée par Johnson et Papadimitriou dans [58]. Le problème intermédiaire est $\frac{1}{n}$-SAT, une extension de $\frac{1}{3}$-SAT où les clauses $\frac{1}{N}(x_1, \cdots, x_N)$ peuvent être de longueur N quelconque et forcent exactement une des variables $x_1, \cdots, x_n$ à être vraie. La variante choisie pour l'Hamiltonicité est UHAM-CYCLE (i.e., l'existence d'un cycle Hamiltonien dans un graphe non-orienté).

Cette réduction est linéaire et parcimonieuse, mais *elle ne préserve pas la planarité* : les gadgets de variables étant toujours alignés en série et faisant face aux gadgets de clauses également alignés en série, les gadgets reliant les gadgets de variables à leurs occurrences dans les gadgets de clauses peuvent se croiser. Ces croisements peuvent être éliminés par la pose de crossovers (parcimonieux) pour obtenir une instance de PLAN-UHAM-CYCLE, mais *un nombre quadratique* de ces objets peut être nécessaire même si l'instance initiale était planaire.

Dans cette section, nous présentons d'abord cette réduction parcimonieuse et quadratique de PLAN-$\frac{1}{n}$-SAT à PLAN-UHAM-CYCLE, dont nous conservons ou modifions certains outils, puis nous réduisons linéairement et parcimonieusement PLAN-SAT à PLAN-UHAM-CYCLE, sans problème intermédiaire.

4.2.1 La réduction quadratique de PLAN-$\frac{1}{n}$-SAT à PLAN-UHAM-CYCLE

Intéressons-nous spécifiquement à la simulation des $\frac{1}{3}$-clauses, puisque le problème particulier $\frac{1}{3}$-SAT est équivalent à SAT linéairement et parcimonieusement, y compris pour leurs versions planaires.

Dans cette transformation, chaque $\frac{1}{3}$-clause (occ_a, occ_b, occ_c) est associée à un gadget planaire la simulant, et présenté en haut de la Fig. 4.1. Ce gadget planaire a deux arêtes pendantes en x_1 et t_1 et trois arêtes distinguées $a = (x_1, y_1)$, $b = (y_1, z_1)$, et $c = (z_1, t_1)$, associées resp. aux occurrences occ_a, occ_b et occ_c. Lorsqu'on donnera la valeur vraie, resp. fausse, à une occurrence occ_u ($u \in \{a, b, c\}$) l'arête distinguée u associée à occ_u sera maigre, resp. grasse :

Propriété 4.7 *Le gadget de la Fig. 4.1 admet trois états locaux, chacun de ces états étant constitué d'un chemin reliant les deux arêtes pendantes et évitant exactement l'une des trois arêtes a, b et c (et ainsi rendant vraie exactement l'une des occurrences occ_a, occ_b, occ_c).*

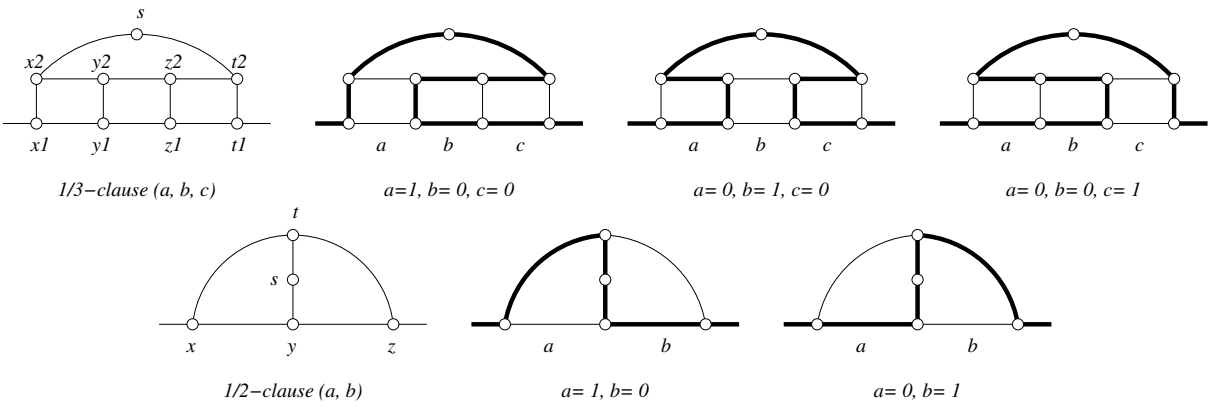


FIG. 4.1 – Les simulateurs de $\frac{1}{3}$-clause et de $\frac{1}{2}$-clause

Preuve Pour tout état local C, les deux seules arêtes pendantes sont nécessairement grasses, de même que les arêtes (x_2, s) et (s, t_2) car s est de degré deux. Il y a trois cas :

1. Supposons que l'arête (x_1, x_2) est grasse (cas en haut à gauche de la Fig. 4.1). Alors (x_1, y_1) et (x_2, y_2) sont maigres, impliquant que le 3-chemin (z_1, y_1, y_2, z_2) est gras. L'arête (z_1, z_2) est ensuite maigre car sinon elle fermerait un 4-cycle gras avec le 3-chemin précédent, impliquant que (z_1, t_1) (z_2, t_2) sont gras et (t_1, t_2) est maigre. L'arête a est la seule à être maigre parmi a, b et c.
2. Supposons que l'arête (t_1, t_2) est grasse (cas en haut à droite de la Fig. 4.1). Alors on obtient le cas symétrique du premier cas et l'arête c est la seule à être maigre parmi a, b et c.
3. Supposons que les deux arêtes (x_1, x_2) et (t_1, t_2) sont maigres (cas en haut au centre de la Fig. 4.1). Alors les arêtes (x_1, y_1) et (z_1, t_1) sont nécessairement grasses. L'arête (y_1, z_1) ne peut être grasse car sinon le chemin (a, b, c) traverse le gadget sans avoir visité tous les sommets. Donc (y_1, z_1) est maigre et (y_1, y_2) et (z_1, z_2) sont grasses. A son tour, l'arête (y_2, z_2) ne peut être grasse car le chemin formé traverserait le gadget sans visiter tous les sommets. Donc (y_2, z_2) est maigre et (x_2, y_2) et (z_2, t_2) sont grasses. L'arête b est la seule à être maigre parmi a, b et c.

Les trois états locaux sont donc bien constitués d'un unique chemin reliant les deux arêtes pendantes, et évitant chacune une arête différente parmi a, b et c. ■

On voit aisément que le gadget est généralisable pour les $\frac{1}{n}$-clauses de longueur quelconque. Bien que cela ne soit pas nécessaire, les auteurs font un cas particulier (voir bas de la Fig. 4.1) de la simulation des $\frac{1}{2}$-clauses (occ_a, occ_b). Ces gadgets ont deux arêtes pendantes et deux arêtes distinguées a et b, resp. associées à v_a et v_b. Nous laissons au lecteur le soin de vérifier que ce gadget a deux états locaux, autorisant exactement l'une des arêtes a ou b à être maigre.

Le gadget simulant une variable v_a du système est réduit à sa plus simple expression : deux sommets x et y ayant chacun une arête pendante, et deux arêtes de mêmes extrémités $a = (x, y)$ et $\overline{a} = (x, y)$. Seule l'arête $\overline{a}$ est distinguée. Il y a trivialement deux états locaux dans un tel gadget, ou bien a est maigre et $\overline{a}$ est grasse, ou bien l'inverse. La convention pour la représentation des valeurs booléennes est identique à celle du gadget de $\frac{1}{n}$-clause : a est maigre, resp. grasse, si v_a est mise à vrai, resp. faux.

Pour mettre en accord la valeur d'une occurrence occ_a d'une $\frac{1}{n}$-clause, avec la valeur de sa variable v_a, il faut forcer l'arête distinguée $\overline{a}$ du gadget de variable associé à v_a à être grasse ssi l'arête distinguée a associée à occ_a dans le gadget de clause est maigre. Le gadget de la Fig. 4.2 permet d'établir cette "relation XOR" entre deux arêtes :

Propriété 4.8 (échelle XOR) *Le gadget planaire de la Fig. 4.2, appelé* échelle XOR*, à quatre arêtes pendantes en x_1, x_3, t_1 et t_3 admet exactement deux états locaux symétriques, chacun constitué d'un chemin reliant l'arête pendante en x_i et l'arête pendante en t_i, pour $i = 1$ ou 3, et laissant les deux autres arêtes pendantes maigres. Le couple d'arêtes pendantes en x_i et t_i pour $i = 1$ (resp $i = 3$) peut être donc vu comme une seule arête virtuelle a (resp. b), a étant maigre ssi b est grasse.*

Preuve Les sommets x_2, y_2, z_2, t_2 étant de degré 2, les quatre 2-chemins (x_1, x_3), (y_1, y_3), (z_1, z_3) et (t_1, t_3) sont toujours gras. Le gadget étant symétrique, on peut

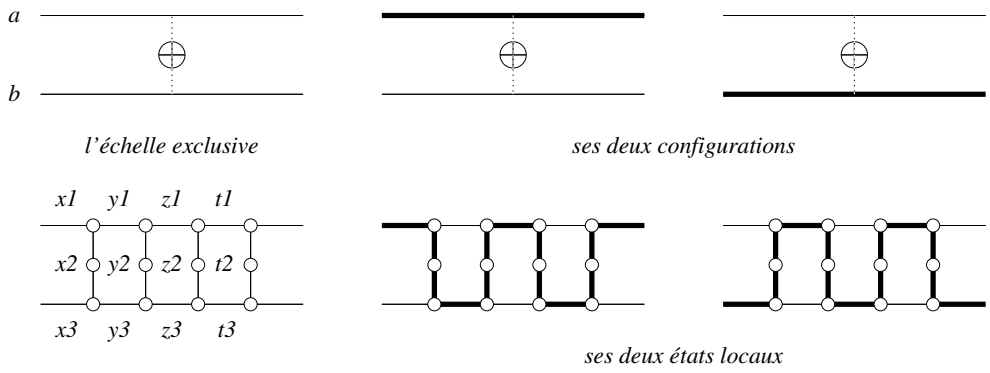


FIG. 4.2 – L'échelle XOR

supposer que l'arête pendante en x_1 est grasse (et donc que (x_1, y_1) est maigre) : Alors l'arête pendante en x_3 ne peut être grasse, car alors, (x_3, y_3) serait maigre, et (y_1, z_1) et (y_3, z_3) seraient toutes deux grasses, refermant ainsi un cycle gras au centre de l'échelle. Donc l'arête pendante en x_3 est maigre, impliquant en cascade que les arêtes (x_3, y_3), (y_1, z_1), (z_3, t_3) et l'arête pendante en t_1 sont grasses. L'état local obtenu forme un unique chemin en zig-zag, décrit au centre de la Fig. 4.2. Le deuxième état local (à droite) est obtenu par symétrie. ■

La version linéaire mais non-planaire

La version linéaire de la réduction de $\frac{1}{n}$-SAT à UHAM-CYCLE consiste à ordonner les variables et les $\frac{1}{n}$-clauses de l'instance $\frac{1}{n}$-SAT selon deux ordres arbitraires, à créer un gadget de variable pour chaque variable v_a et un gadget de $\frac{1}{n}$-clause pour chaque $\frac{1}{n}$-clause$(v_a, \cdots, v_z)$, et à connecter les gadgets en série selon les deux ordres choisis. Les deux séries sont elles-même connectées entre elles à leurs deux extrémités. Chaque gadget est séparé de son successeur par un sommet de degré 2 connectant une de leurs arêtes pendantes. Enfin, pour chaque occurrence occ_a d'une variable v_a dans une $\frac{1}{n}$-clause $(\cdots, occ_a, \cdots)$, on crée une échelle XOR entre l'arête distinguée a correspondant à occ_a dans le gadget de $\frac{1}{n}$-clause et l'arête distinguée $\overline{a}$ du gadget de variable associé à v_a. La Fig. 4.3 illustre la représentation du système de $\frac{1}{n}$-clauses $(a, c), (a, b, d), (c, d)$. La Fig. 4.4 montre le même système avec les échelles XOR "expansées".

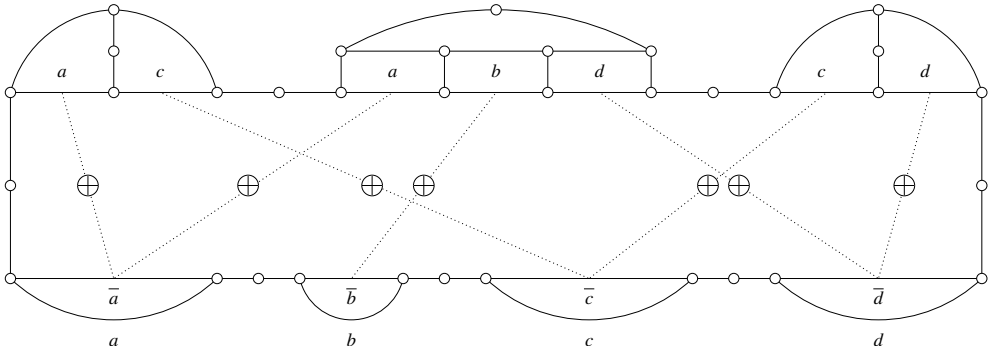


FIG. 4.3 – La représentation du système $\{(a, c), (a, b, d), (c, d)\}$

Chaque gadget de variable étant précédé et suivi d'un sommet de degré 2, tout cycle hamiltonien H doit visiter la série de gadgets de variables en choisissant de visiter soit l'arête

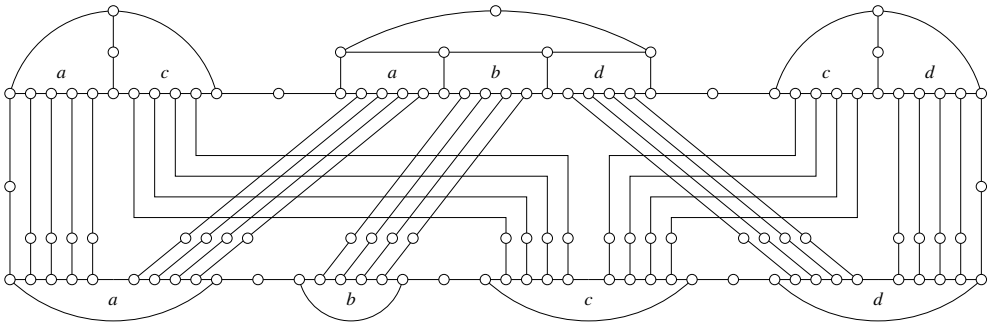


FIG. 4.4 – Le graphe obtenu après "expansion" des échelles XOR

distinguée a, soit l'arête distinguée $\overline{a}$ de chacun de ces gadgets, ce qui signifie que H représente bien une valuation du système (à toute variable est donnée une unique valeur de vérité). Par la propriété des échelles XOR, la maigreur de l'arête distinguée a de chaque gadget de variable pour v_a est "recopiée" dans toutes les arêtes distinguées a des gadgets de $\frac{1}{n}$-clauses où l'on trouve une occurrence occ_a de v_a : les valeurs des occurrences sont donc cohérentes. Exactement une des arêtes distinguées de chaque gadget de clause devant être maigre, H représente bien une solution du système $\frac{1}{n}$-SAT. Les Figs. 4.5 et 4.6 (avec les échelles XOR expansées) montrent le cycle Hamiltonien associé à la solution $(a = d = faux, b = c = vrai)$ pour le système $\frac{1}{n}$-SAT de l'exemple précédent.

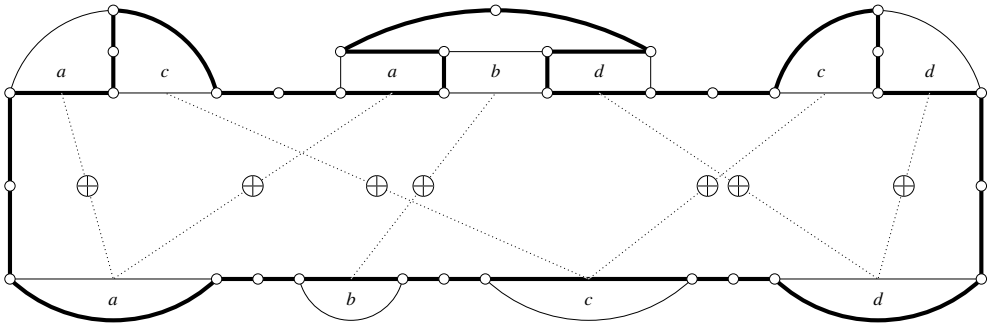


FIG. 4.5 – La représentation d'une solution du système

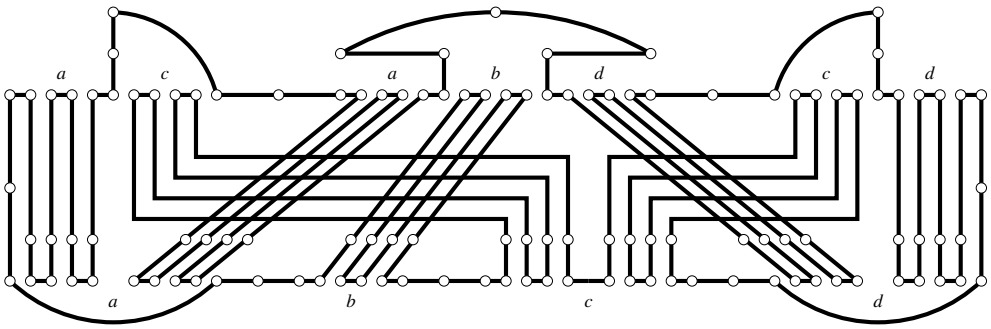


FIG. 4.6 – Le cycle Hamiltonien associé à la solution

La réduction est clairement parcimonieuse et linéaire, mais l'instance obtenue n'est pas nécessairement planaire même si l'instance $\frac{1}{n}$-SAT d'origine l'est, du fait du montage en série des gadgets de variables et de $\frac{1}{n}$-clauses, qui peuvent faire croiser plusieurs échelles XOR entre elles.

Une version planaire mais quadratique

La version planaire de la réduction de $\frac{1}{n}$-SAT à UHAM-CYCLE reprend la réduction précédente en résolvant les croisements des échelles XOR par la pose d'un crossover au niveau de chaque croisement. Ce crossover est représenté sur la Fig. 4.7.

Propriété 4.9 *Le gadget planaire de la Fig. 4.7 qui a huit arêtes pendantes en x_i, t_i, x'_i, t'_i (pour $i = 1$ ou 3) admet exactement quatre états locaux. Chaque état local est formé de deux chemins, l'un reliant ou bien les arêtes pendantes en x_1 et t_1, ou bien les arêtes pendantes en x_3 et t_3, l'autre reliant ou bien les arêtes pendantes en x'_1 et t'_1, ou bien les arêtes pendantes en x'_3 et t'_3, indépendamment de l'autre chemin. Le couple d'arêtes pendantes en x_1 et t_1 (resp. en x_3 et t_3, en x'_1 et t'_1, en x'_3 et t'_3) peut être vu comme une unique arête a (resp. b, a', b'), l'arête a (resp. a') étant maigre ssi l'arête b (resp. b') est grasse.*

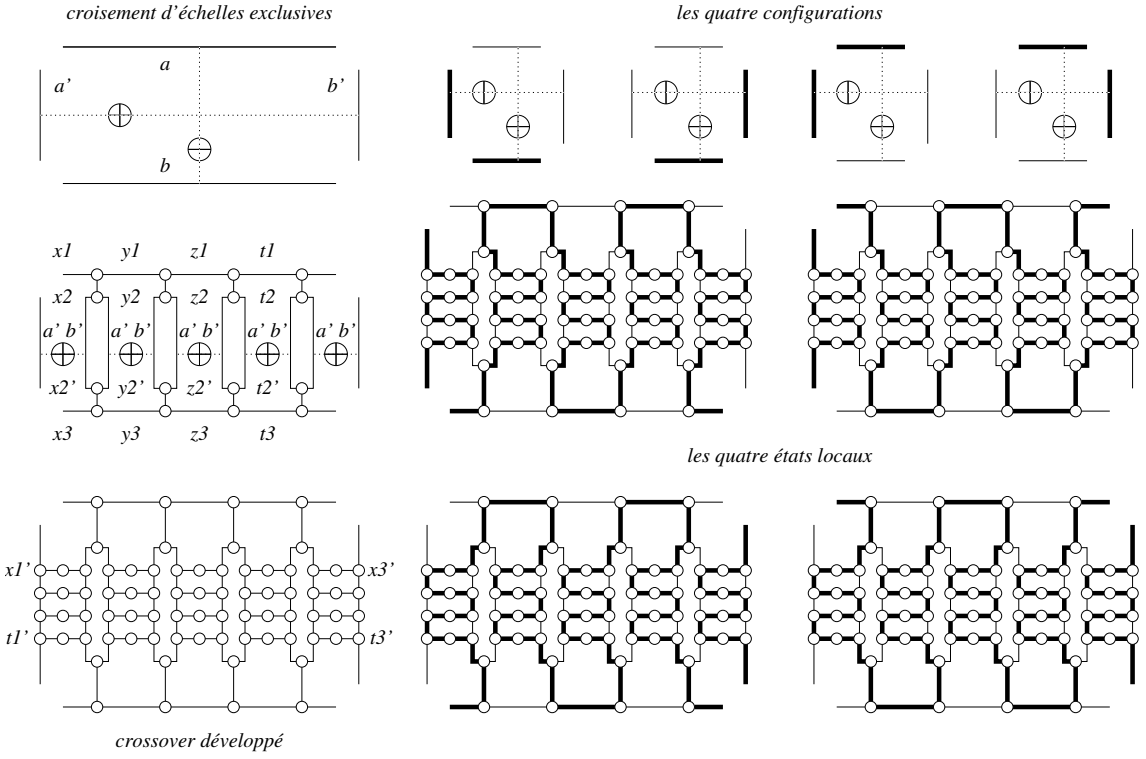


FIG. 4.7 – Le crossover décroisant deux échelles XOR

Preuve On s'appuie sur le schéma intermédiaire à gauche de la Fig. 4.7. Les sommets x_2 et x'_2 sont reliés par deux arêtes nommées a' et b'. L'une est maigre ssi l'autre est grasse car sinon, soit un cycle gras (x_2, x'_2, x_2) se forme, soit x_2 et x'_2 ne sont pas visités. Il en est de même pour toutes les autres arêtes nommées a' et b' reliant y_2 et y'_2, z_2 et z'_2, t_2 et t'_2. De plus, chaque arête nommée a' étant reliée par une échelle XOR à l'arête nommée b' située immédiatement à sa droite, toutes les arêtes

nommées a' – resp. b' – ont même statut, les arêtes a' étant maigres ssi les arêtes b' sont grasses. Les "chemins" constituant les barreaux de l'échelle (x_1, x_2, x'_2, x_3), (y_1, y_2, y'_2, y_3), (z_1, z_2, z'_2, z_3), (t_1, t_2, t'_2, t_3) étant tous gras, le reste de la preuve est identique à celle de la Prop. 4.8. ■

La Fig. 4.8 montre le graphe de l'exemple précédent après la pose des trois crossovers, et la Fig. 4.9 montre le cycle Hamiltonien représentant la solution $(a = d = faux, b = c = vrai)$ dans le nouveau graphe maintenant planaire. La réduction est toujours parcimonieuse mais elle est quadratique, $O(\ell^2)$ crossovers pouvant être requis, où ℓ est la longueur de la formule $\frac{1}{n}$-SAT.

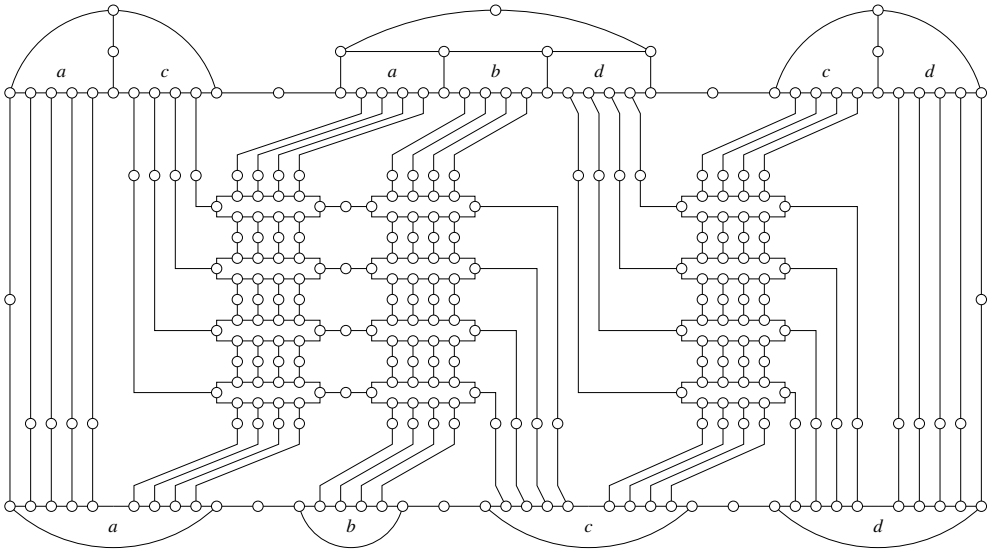


FIG. 4.8 – Résolution des croisements des échelles par pose de crossovers

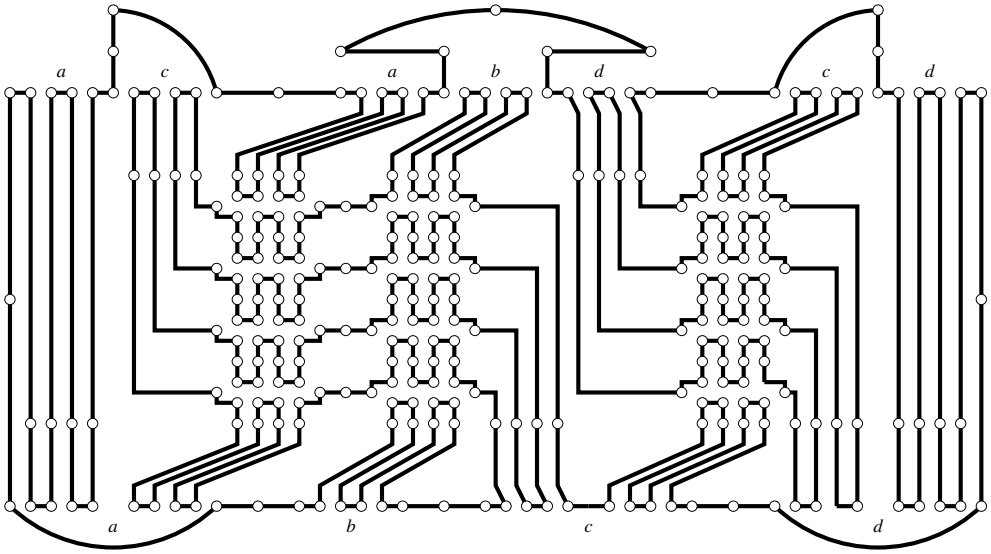


FIG. 4.9 – Le nouveau cycle Hamiltonien planaire associé à la solution

4.2.2 Vers une réduction planaire et linéaire de SAT à UHAM-CYCLE

Notre réduction de SAT à UHAM-CYCLE suit le même principe que la réduction précédente, à savoir des gadgets de clauses et de variables dont les arêtes distinguées sont connectées par des échelles XOR. Nous utilisons cependant la convention inverse pour la représentation des booléens : la valeur vraie, resp. fausse, est représentée par une arête grasse, resp. maigre.

Ce qui rend la réduction précédente quadratique vient du fait qu'elle ne respecte pas le plongement dans le plan du graphe de formule G_φ associé à l'instance SAT $\varphi(V, \overline{C})$ à réduire, en mettant bout à bout gadgets de clauses d'un côté et gadgets de variables de l'autre. Cette stratégie est déterminée par la nature du seul crossover dont nous disposons : il résout uniquement les croisements entre échelles XOR. Notre stratégie est de conserver le plongement dans le plan en adressant les deux problèmes liés à la connexion des arêtes pendantes des gadgets se trouvant plongés dans différentes faces du graphe ainsi construit.

1. Connecter les arêtes pendantes de deux gadgets plongées dans deux faces distinctes signifie qu'une arête doit être capable de traverser les frontières des faces, constituées d'échelles XOR. Il faut donc construire un nouveau crossover, *résolvant les croisements de type arête/échelle*, et non plus de type échelle/échelle. Nous verrons que ce crossover possède trois états locaux, raison pour laquelle nous le baptiserons *flip-flap-flop*.
2. Afin d'assurer un nombre linéaire de crossovers, nous devons organiser la visite des différentes faces intelligemment. A moins de disposer d'un cycle Hamiltonien dans le graphe dual de G_φ, nous ne pouvons pas visiter chaque face une et une seule fois les unes à la suite des autres. Nous visiterons donc les faces en *longeant les arêtes d'un arbre couvrant arbitraire du graphe dual* de G_φ.

Nous construisons donc la première partie du graphe de la façon suivante :

- Pour toute clause $C_i \in \overline{C}$, on créé un gadget de clause $g(C_i)$ associé à C_i, et pour toute variable $v \in V$, on créé un gadget de variable $g(v)$ associé à v. L'implémentation des gadgets de variables est identique à celle de la réduction précédente mais ses deux arêtes non-pendantes sont distinguées, et associées resp. aux littéraux v et $\overline{v}$. Les gadgets de clauses restent à définir mais suivent la même interface que précédemment : deux arêtes pendantes et une arête distinguée par occurrence de littéral dans la clause simulée.
- Pour toute occurrence ℓ_v d'une variable $v \in V$ dans une clause $C_i \in \overline{C}$, on connecte l'arête distinguée de $g(C_i)$ associée à ℓ_v et l'arête distinguée de $g(v)$ associée à $\overline{\ell_v}$ par une échelle XOR, tout en respectant le plongement de G_φ dans le plan. Ceci signifie que l'ordre des échelles XOR autour de chaque gadget dans le sens trigonométrique est le même que celui des arêtes correspondantes dans G_φ. De plus, les arêtes pendantes d'un même gadget doivent être maintenues consécutives afin qu'elles restent toujours dans la même face. Ceci implique que l'une des arêtes distinguées des gadgets de variables ne se trouve pas le long de sa face externe, et que son accès nécessite la pose d'un flip-flap-flop.

Le graphe partiel obtenu à cette étape est appelé le "squelette". Voir Fig. 4.10.

Dans un deuxième temps, il faut connecter les arêtes pendantes de tous les gadgets. Comme nous l'avons dit précédemment, afin d'obtenir un nombre linéaire de crossovers, nous organisons la connexion des arêtes pendantes des gadgets à partir d'un arbre couvrant du dual de G_φ. Soit T un tel arbre et T' l'arbre correspondant dans le squelette. A chaque arête e' de T correspond une échelle XOR duale e dans le squelette (qui se coupent sur la Fig. 4.11). Sur chacune de ces échelles e du squelette, nous posons deux flip-flap-flops consécutifs. Chacun de ces couples est maintenant considéré comme un unique gadget à quatre arêtes pendantes (à raison d'une paire d'arêtes pendantes plongées dans chacune des deux faces adjacentes à e).

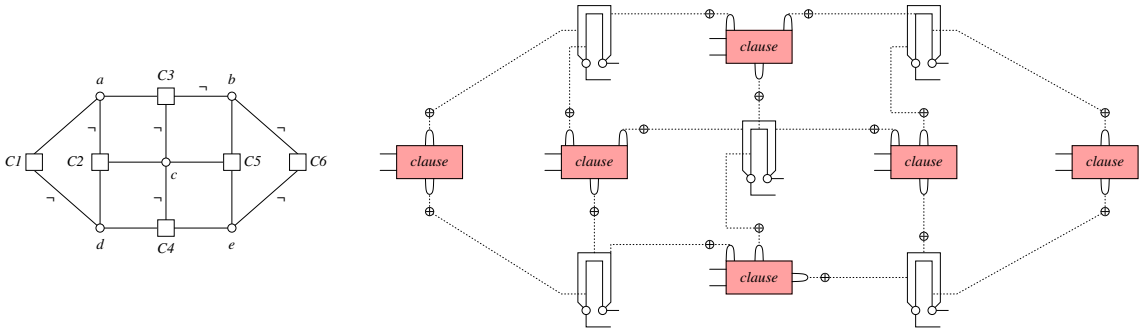


FIG. 4.10 – Squelette obtenu pour un plongement planaire pour la formule $\varphi = C_1 \wedge ... \wedge C_6$ avec $C_1 = a \vee \neg d$, $C_2 = \neg a \vee c \vee d$, $C_3 = a \vee \neg b \vee \neg c$, $C_4 = \neg c \vee e \vee d$, $C_5 = b \vee e \vee c$, $C_6 = \neg b \vee \neg e$

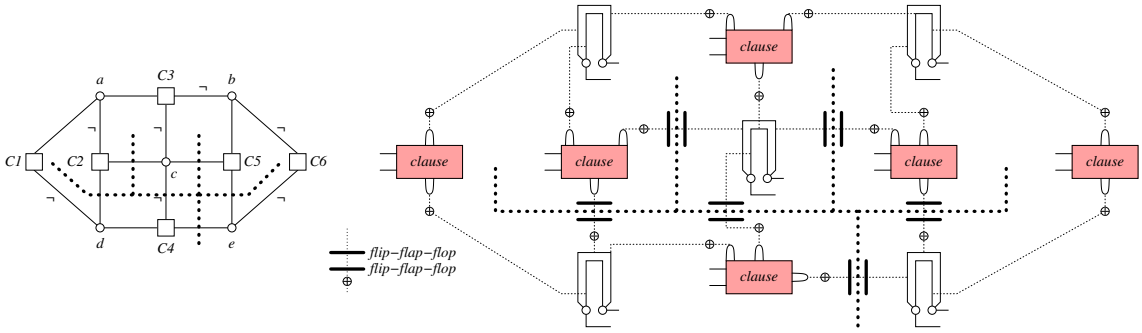


FIG. 4.11 – Pose des crossovers le long d'un arbre couvrant dans le dual

Le tracé des connexions entre gadgets épouse alors T' comme un gant : Pour chaque face F (sans oublier la face externe) de G_φ, soit F' la face correspondante dans le squelette. Soit $(g_0, \cdots, g_{k-1})$ la liste des k gadgets obtenus en balayant la frontière de F dans l'ordre trigonométrique (on peut considérer que cette frontière est toujours connexe si G_φ est lui-même connexe). Soit (e_i, f_i) le couple d'arêtes pendantes de g_i plongées dans F tel que e_i précède f_i pour ce même ordre. Pour tout $0 \leq i < k$, nous fusionnons les arêtes pendantes f_i et e_{i+1} (l'addition étant effectuée modulo k). Ces connexions vont constituer la colonne vertébrale de tout cycle Hamiltonien dans le graphe. Voir Figs. 4.12 et 4.13.

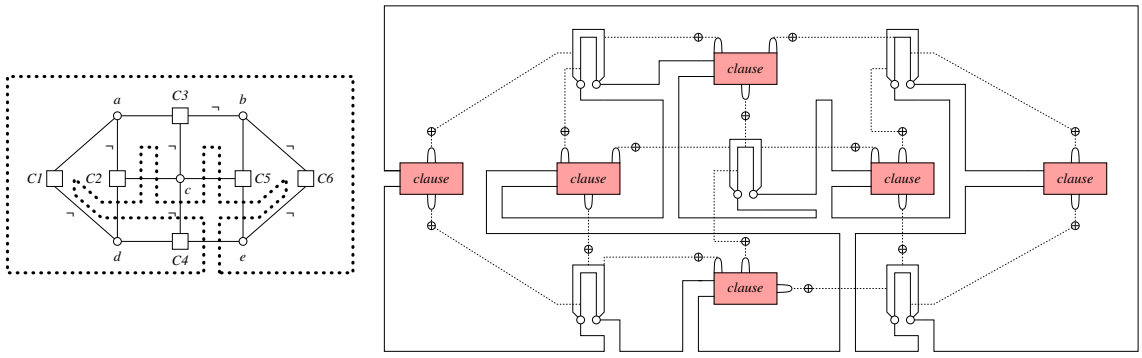


FIG. 4.12 – Ajout de la colonne vertébrale au squelette

La linéarité découle du fait que la construction du dual d'un graphe ainsi que la construction d'un arbre couvrant se font en temps linéaire. La correction et la parcimonie de la transformation dépendent de celles des gadgets qui restent à implémenter.

Nous faisons aussi l'hypothèse que G_φ est connexe lors du balayage de la frontière des faces. Cette hypothèse est facilement supprimée en réduisant les k composantes connexes de G_φ séparément puis en fusionnant les k colonnes vertébrales par leurs faces externes autour d'une étoile imaginaire : Plonger les k composantes connexes autour d'un même sommet imaginaire selon un ordre trigonométrique arbitraire $K_0, \cdots, K_{k-1}$. Pour toute composante K_i, on casse une arête arbitraire de la face externe de sa colonne vertébrale en deux arêtes pendantes e_i et f_i, de telle sorte que f_i précède e_{i+1} pour l'ordre choisi (l'addition étant modulo k). On fusionne f_i et e_{i+1} pour tout $0 \leq i < k$ afin de créer l'étoile réunissant les k colonnes vertébrales.

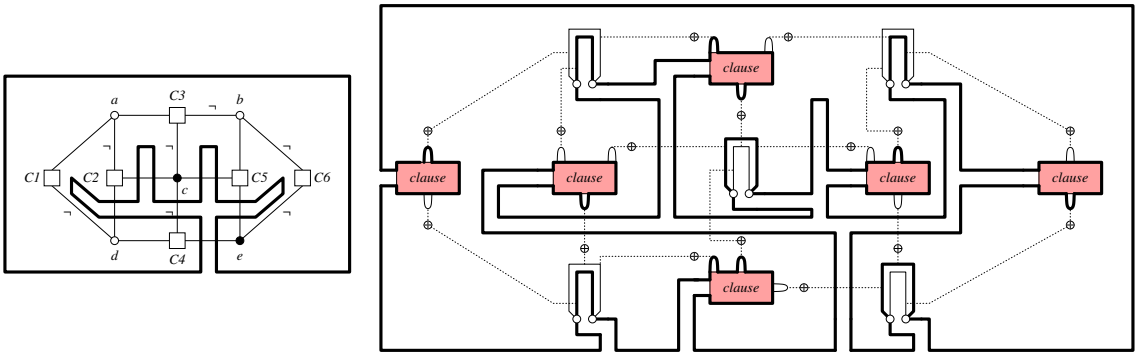


FIG. 4.13 – Un cycle Hamiltonien représentant la valuation $c = e = faux, a = b = d = vrai$

A ce stade, nous présentons maintenant les *gadgets logiques* permettant de construire le simulateur de clause. L'existence d'un flip-flap-flop parcimonieux sera admise et exploitée pour la construction de ces gadgets. La présentation du flip-flap-flop est reportée en Section 4.2.4.

4.2.3 Des gadgets Hamiltoniens logiques et fonctionnels

Les gadgets qui suivent sont construits et composés à la manière des portes d'un circuit électronique. Ces gadgets ont des arêtes distingués jouant le rôle des bits opérandes d'une porte, ainsi que deux arêtes pendantes pour chaque bit de sortie ainsi que le fil d'alimentation. La composition de gadgets élémentaires va permettre l'élaboration de gadgets plus complexes d'une manière complètement transparente.

Notre gadget de clause va être fabriqué par un simple chaînage de gadgets OR, eux-mêmes obtenus par combinaison de gadgets AND et NOT par application de la loi de De Morgan pour les connecteurs logiques : $x \vee y \iff \neg(\neg x \wedge \neg y)$. Cette composition sera d'autant plus facile que nos gadgets AND et OR sont *fonctionnels*, c'est-à-dire qu'ils produisent un (ou plusieurs) bit(s) de sortie à la manière des portes d'un circuit :

Définition 4.10 (gadget fonctionnel) *Un gadget fonctionnel représente une fonction booléenne* $\{0,1\}^k \to \{0,1\}^q$ *en utilisant l'interface suivante :*

- *k arêtes distinguées, représentant chacune un bit opérande, l'arête étant grasse ssi le bit correspondant est allumé, et,*
- *$3q$ arêtes pendantes : q arêtes d'entrée in_i, et q paires d'arêtes de sortie correspondantes $out_i = (out_i^0, out_i^1)$, $0 \leq i < q$. Une paire d'arêtes de sortie out_i code un bit du résultat,*

l'arête out_i^1 (resp. out_i^0) étant grasse ssi le bit i du résultat vaut 1 (resp. 0). Les arêtes d'entrée in_i sont toujours grasses.

De plus, comme on veut représenter une fonction totale sur $\{0,1\}^k$:

- *Les k arêtes distinguées peuvent toujours être grasses ou maigres indépendamment les unes des autres, i.e., des états locaux existent pour chacune de ces 2^k combinaisons.*
- *Un état local est toujours constitué de q chemins, chacun entrant par l'une des q arêtes d'entrée in_i, et sortant par une arête de la paire de sortie correspondante (out_i^0, out_i^1) entièrement déterminée par l'état gras ou maigre des k arêtes distinguées.*

Bien sûr, les chemins sont entièrement déterminés lorsque le gadget est parcimonieux.

Comme dans ce chapitre, nous ne manipulons que des fonctions d'arité $k = 2$ sur $q = 1$ bit (ce ne sera pas le cas dans le chapitre 5 où nous ferons du comptage avec des gadgets arithmétiques), nous noterons simplement *in* et $out = (out_0, out_1)$ les arêtes d'entrée et sortie de nos gadgets. Voici le gadget AND :

Propriété 4.11 *Le gadget planaire à droite de la Fig. 4.14, qui a trois arêtes pendantes in, out_1, out_0 et deux arêtes distinguées (c,d), (b,d) disposées anti-trigonométriquement dans cet ordre, implémente parcimonieusement la fonction AND, i.e., le gadget admet exactement quatre états locaux, tous entrant par l'arête in et sortant par out_1 ssi (b,d) et (c,d) sont grasses, et sortant par out_0 sinon.*

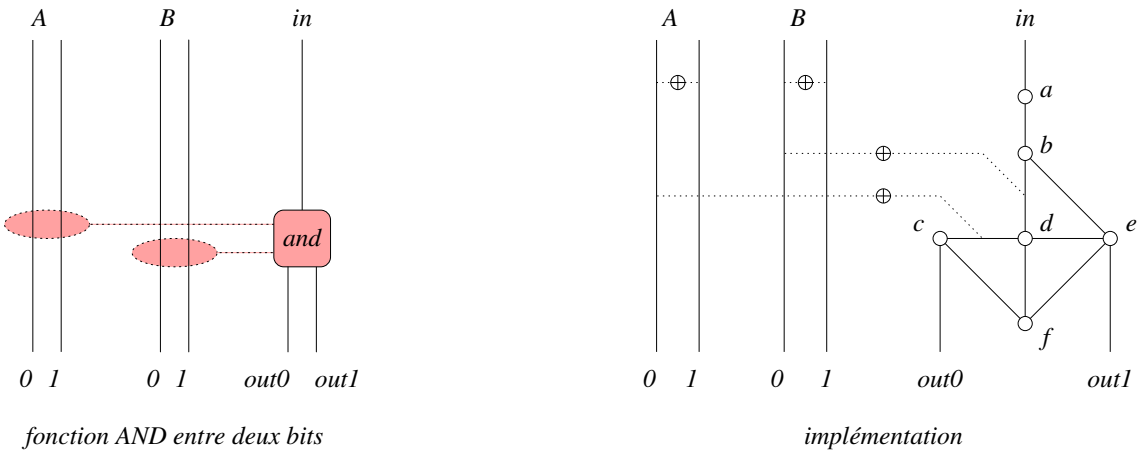


FIG. 4.14 – Le gadget fonctionnel AND

Preuve L'arête *in* est nécessairement grasse à cause du sommet a de degré 2 adjacent. Il ne peut y avoir d'autre chemin que celui passant par l'arête *in* car un tel deuxième chemin entrerait et sortirait par out_0 et out_1, ne laissant aucune sortie libre pour le premier chemin. L'unique chemin sortira donc soit par out_0 soit par out_1. Il y a quatre cas, reproduits sur la Fig. 4.15, selon l'état gras ou maigre des deux arêtes distinguées :

1. Supposons que (b,d) est maigre : Alors (b,e) est nécessairement grasse. On ne peut sortir en out_1, car il faut visiter les sommets c, d et f. Donc out_1 est maigre et out_0 est grasse.
   (a) Si de plus (c,d) est maigre, il ne reste plus que (d,e) et (d,f) pour visiter le sommet d, ainsi que (c,f) pour visiter le sommet c. Ces arêtes sont donc grasses. Le chemin obtenu code le résultat $AND(0,0) = 0$.

(b) Si de plus (c, d) est grasse, alors (d, e) ne peut être grasse car sinon f n'est pas visité. Donc l'arête (d, e) est maigre et les arêtes (e, f) et (d, f) sont grasses. Le chemin obtenu code le résultat $AND(1, 0) = 0$

2. Supposons que (b, d) est grasse :

(a) Si de plus (c, d) est maigre, il ne reste plus que (c, f) et out_0 pour visiter le sommet c. Ces arêtes sont donc grasses. L'arête (d, f) ne peut ensuite être grasse, car sinon le sommet e ne serait pas visité. Donc (d, f) est maigre et les arêtes (d, e) et (e, f) sont grasses. Le chemin obtenu code le résultat $AND(0, 1) = 0$.

(b) Si de plus (c, d) est grasse, on ne peut sortir immédiatement en out_0, car il faut visiter e et f. Par conséquent, (c, f), (e, f) et out_1 sont grasses. Le chemin obtenu code le résultat $AND(1, 1) = 1$

Nous avons quatre états locaux pour quatre configurations, le gadget est donc bien parcimonieux. ■

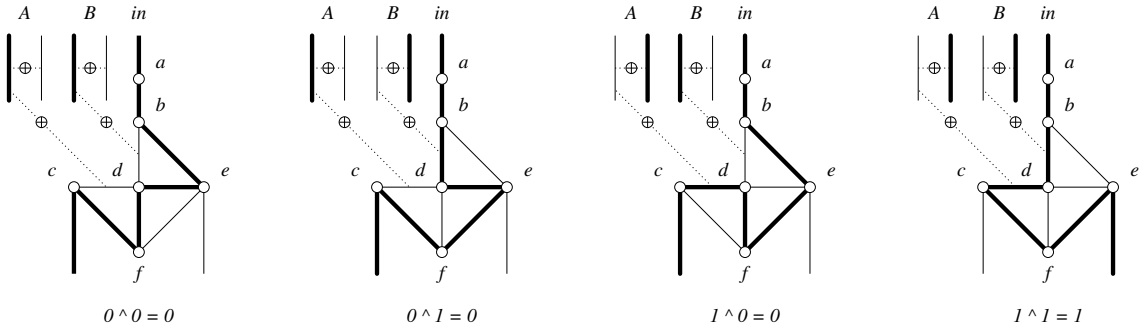


FIG. 4.15 – Les quatre configurations du gadget fonctionnel AND

Remarque 4.12 *Le gadget AND sera noté par la suite par la boite noire présentée à gauche sur la Fig. 4.14. Dans toutes les figures, la boite aura toujours son arête d'entrée en haut, ses arêtes de sortie en bas, et ses opérandes à gauche. Comme toujours, les opérandes sont connectés aux arêtes distinguées du gadget par des échelles XOR. On voit sur la droite de la Fig. 4.14 que plusieurs flip-flap-flops peuvent être nécessaires pour effectuer ces connexions, des croisements entre les arêtes des opérandes et les échelles étant inévitables.*

En ce qui concerne le gadget NOT, il est plus pratique à l'usage de l'implémenter comme un inverseur sur place que comme un gadget fonctionnel : Le gadget planaire de la Fig. 4.16 implémente parcimonieusement un tel inverseur. Il s'agit simplement d'une échelle XOR à laquelle il manque un barreau. Le lecteur peut facilement vérifier que les deux seuls états locaux du gadgets sont les deux chemins en zig-zag au centre et à droite de la Fig. 4.16, et que le bit circulant dans le couple d'arêtes (out_0, out_1) est l'opposé du bit circulant dans le couple d'arêtes (in_0, in_1). Dans les figures suivantes, le gadget NOT sera représenté par le symbole $\otimes$.

Comme il a été annoncé plus haut, l'opérateur OR est simplement obtenu par combinaison des opérateurs AND et NOT en appliquant la loi de De Morgan : voir Fig. 4.17. Les deux couples d'arêtes opérandes $A = (A_0, A_1)$ et $B = (B_0, B_1)$ passent chacun par un premier inverseur, les couples d'arêtes de sortie sont opérandes d'un gadget AND dont le couple d'arêtes de sortie est à son tour inversé, obtenant ainsi le résultat de $OR(A, B)$ dans le couple d'arête (out_0, out_1).

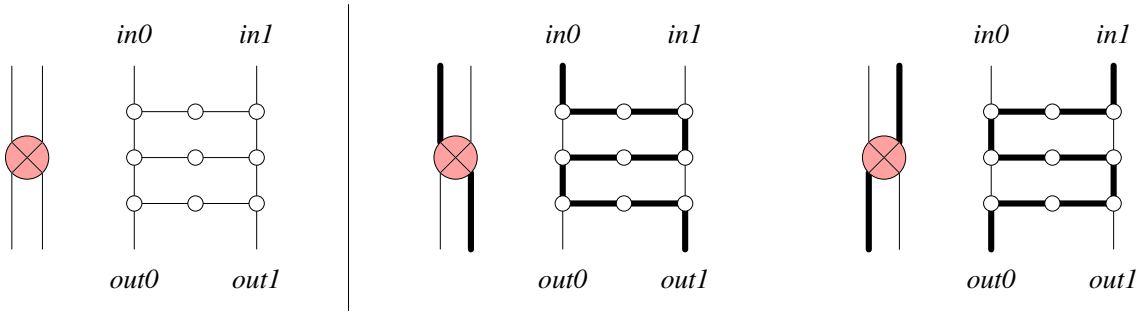


FIG. 4.16 – Le gadget NOT

Enfin, A et B passent chacun dans un deuxième inverseur afin de rétablir les valeurs initiales des opérandes.

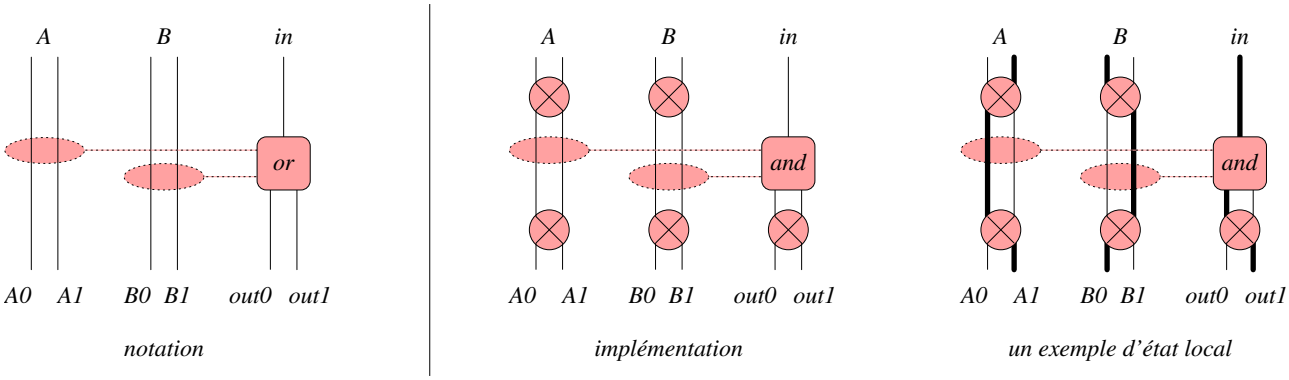


FIG. 4.17 – Le gadget OR

Le gadget de clause simulant une clause à k littéraux $\ell_1 \vee \cdots \vee \ell_k$ est implémenté en accumulant le résultat des opérateurs OR : $a_1 = OR(\ell_1, \ell_2)$, $a_2 = OR(a_1, \ell_3)$, ..., $a_j = OR(a_{j-1}, \ell_{j+1})$, $1 < j < k$. On force le dernier accumulateur a_{k-1} à porter la valeur vraie en faisant passer l'arête out_1 du couple par un sommet c de degré 2. Le gadget est clairement de taille linéaire en k et parcimonieux (sous réserve que le flip-flap-flop le soit aussi). La construction est présentée dans la Fig. 4.18.

4.2.4 Le flip-flap-flop

Nous implémentons maintenant le flip-flap-flop qui nous permet de croiser une échelle XOR et une arête. Le gadget a dix arêtes pendantes : 2 pour la jonction de l'arête qui croise l'échelle et 2×4 pour la jonction des quatre barreaux de l'échelle. Notons qu'une échelle peut croiser plusieurs arêtes, qui peuvent être maigres ou grasses indépendamment les unes des autres. Il faut donc que le jeu de configurations du flip-flap-flop soit compatible avec le montage en série de plusieurs flip-flap-flops F_i. Nous utilisons un jeu de trois configurations (voir Fig. 4.19), nommées resp. *flip*, *flap* et *flop*, d'où le nom du gadget. Soient e_1 et e_2 les arêtes dont l'échelle doit assurer la "relation XOR" et $f_1, \ldots, f_k$, les k arêtes croisant l'échelle XOR en les k flip-flap-flops $F_1, \cdots, F_k$ respectifs (e_1 étant à gauche et e_2 à droite). La signification des configurations est la suivante :

1. *flip* apparaît sur le flip-flap-flop F_i lorsque l'arête f_i est grasse et que l'échelle XOR assure que e_1 est maigre et que e_2 est grasse.

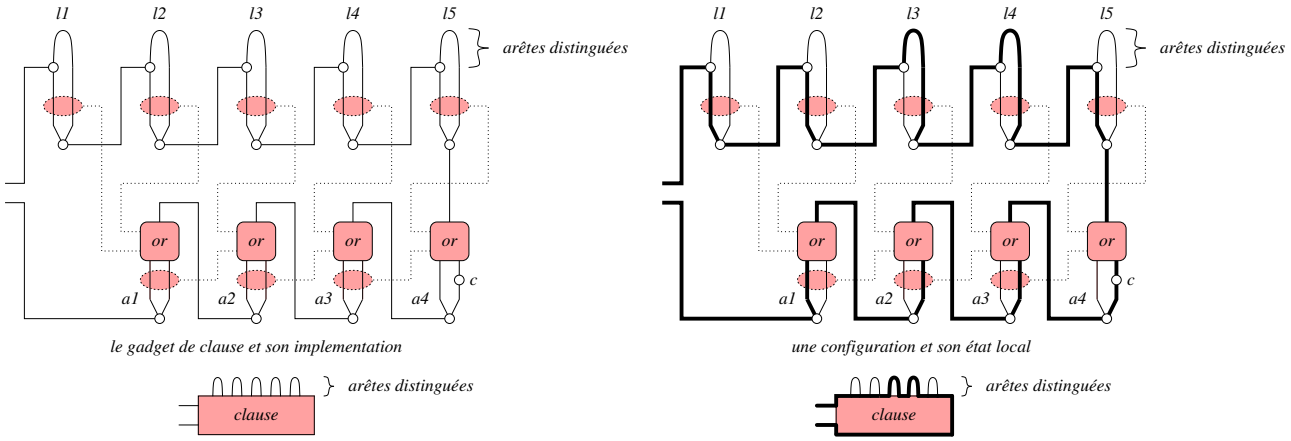


FIG. 4.18 – Le gadget de clause pour $\ell_1 \vee \cdots \vee \ell_5$

2. *flap* est la configuration symétrique de *flip*, e_1 étant grasse et e_2 maigre.
3. *flop* est la configuration qui apparaît sur le flip-flap-flop F_i lorsque f_i est maigre, quel que soit le statut de e_1 et e_2.

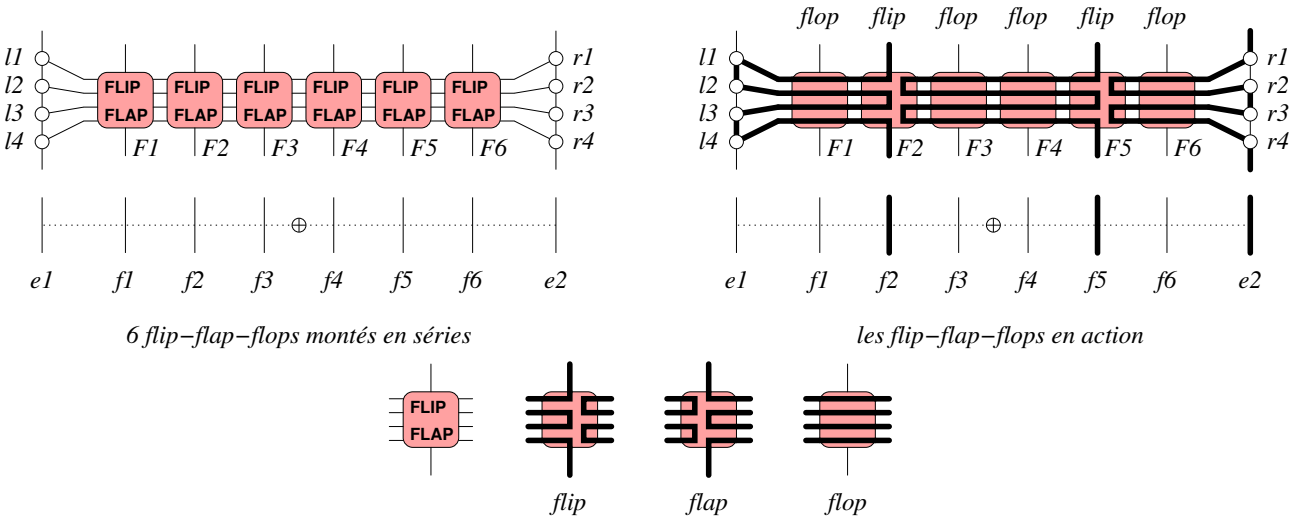


FIG. 4.19 – Le jeu de configurations sur un flip-flap-flop

Vérifions que les combinaisons légales des configurations sur k flip-flap-flops montés en série assurent la "relation XOR" entre e_1 et e_2 ainsi que le libre passage de chemins sur les k arêtes f_i. Interprétons une combinaison de configurations comme un mot w de longueur k sur l'alphabet $\Sigma = \{flip, flap, flop\}$:

1. Si $w = flop^k$, alors la série de gadgets fonctionne comme quatre longs barreaux d'échelle XOR, tous gras. La "relation XOR" entre e_1 et e_2 est assurée de la même manière que l'échelle XOR, et aucun chemin ne passe par les arêtes f_i. Voir cas 1 sur la Fig. 4.20 :
2. Si $w \in (flip|flop)^k$ avec au moins une occurrence de $flip$, alors on vérifie sur le cas 2 de la Fig. 4.20 que tout chemin entrant par l'arête pendante Sud d'un flip-flap-flop ressort

toujours par l'arête Nord de ce même flip-flap-flop. Aucun cycle n'apparaît le long du ruban. De plus, soit F_g, resp. F_d, le flip-flap-flop le plus à gauche, resp. le plus droite, qui admet une configuration $flip$: A gauche, l'arête (l_2, l_3) ne peut être grasse, un cycle central se formant alors le long de $F_1, \cdots, F_g$. Donc (l_2, l_3) est maigre, (l_1, l_2) et (l_3, l_4) sont grasses et les deux arêtes pendantes en l_1 et l_4 sont maigres. A droite, (r_2, r_3) ne peut être maigre car cela impliquerait que (r_1, r_2) et (r_3, r_4) soient grasses, fermant alors deux cycles superposés le long de $F_d, \cdots, F_k$. Donc (r_2, r_3) est grasse, de même que les arêtes pendantes en r_1 et r_4, ce qui achève d'assurer la "relation XOR" entre e_1 et e_2.

3. Si $w \in (flap|flop)^k$ avec au moins une occurrence de $flap$, alors on obtient le cas symétrique du précédent. Voir cas 3 de la Fig. 4.20.
4. Si w possède un facteur $v = w_i \cdots w_j \in flipflop^*flap$, alors deux cycles superposés apparaissent le long de $F_i \cdots F_j$. Cette combinaison est donc illégale. Voir cas 4 de la Fig. 4.20.
5. Si w possède un facteur $v = w_i \cdots w_j \in flapflop^*flip$, alors un cycle central apparaît le long de $F_i \cdots F_j$. Cette combinaison est donc aussi illégale. Voir cas 5 de la Fig. 4.20.

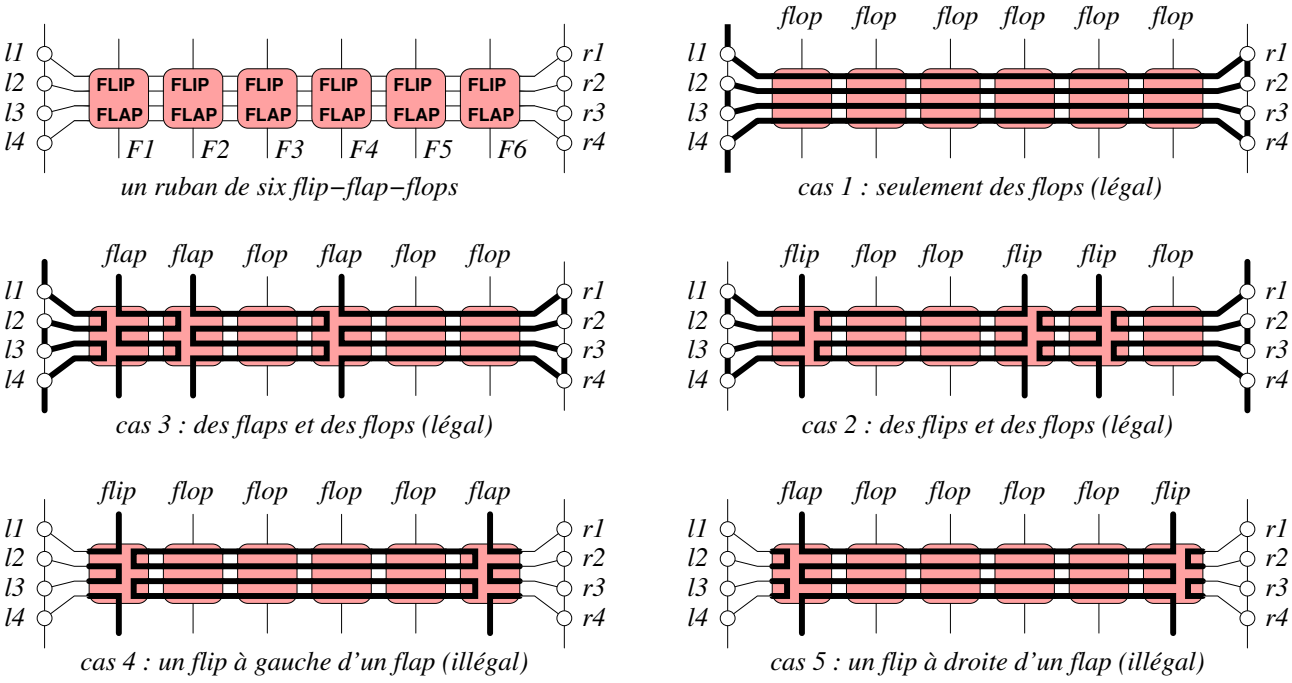


FIG. 4.20 – Les combinaison légales et illégales sur une série de flip-flap-flops

Nous donnons enfin une implémentation parcimonieuse du flip-flap-flop.

Propriété 4.13 *Le gadget planaire de la Fig. 4.21 implémente parcimonieusement un flip-flap-flop, i.e., il admet exactement trois états locaux, correspondant respectivement aux trois configurations flip, flap et flop de la Fig. 4.19.*

Preuve Les sommets de degré 2 impliquent que toutes les arêtes pendantes sont grasses, de même que les arêtes adjacentes (a_i, b_i), (c_i, d_i) pour $1 \leq i \leq 4$ ainsi que les arêtes centrales (u, v), (v, w), (x, y) et (y, z). Il y a trois cas selon le statut des arêtes (w, z) et (u, x) qui ne peuvent pas être grasses en même temps (un cycle central (u, v, w, z, y, x, u) se formerait sinon) :

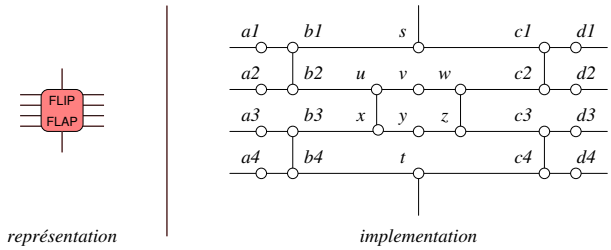


FIG. 4.21 – L'implémentation d'un flip-flap-flop

1. Supposons que (w, z) est grasse (Cas à gauche sur la Fig. 4.22). Les arêtes (w, c_2) et (z, c_3) sont alors maigres, impliquant que (c_1, c_2) et (c_3, c_4) sont grasses. Ceci implique que les arêtes (b_2, u), (b_3, x) sont grasses, de même que (b_1, s) et (b_4, t), ainsi que les arêtes pendantes en s et t. Nous obtenons un état local correspondant à la configuration flip.
2. Supposons que (u, x) est grasse. (Cas au centre sur la Fig. 4.22). C'est le cas symétrique du cas précédent, et il amène à l'état local correspondant à la configuration flap.
3. Supposons que ni (u, x) ni (w, z) ne sont grasses. (Cas à droite sur la Fig. 4.22). Alors à gauche, les arêtes (b_2, u), (b_3, x) sont grasses, ainsi que (b_1, s) et (b_4, t). De même à droite, (w, c_2) et (z, c_3) sont grasses, ainsi que (s, c_1) et (t, c_4). Nous obtenons un état local correspondant à la configuration flop.

■

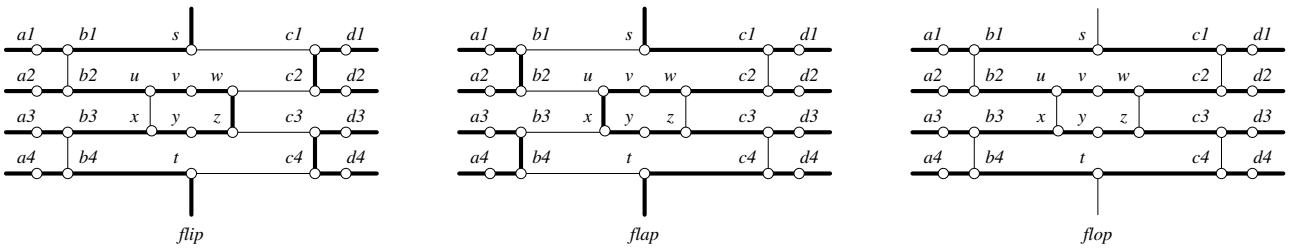


FIG. 4.22 – Les états locaux pour les configurations flip, flap, et flop

Ceci achève la démonstration de la réduction linéaire et parcimonieuse de PLAN-SAT à PLAN-UHAM-CYCLE.

4.3 De l'Hamiltonicité planaire à la satisfaisabilité planaire

Nous présentons maintenant la réduction réciproque, de PLAN-UHAM-CYCLE à PLAN-SAT. En voici l'intuition : Tout d'abord, étant donné un graphe planaire et connexe $G(V, E, F)$, notre système SAT simule une partition $\mathcal{C}$ de V en k cycles disjoints (ce que nous appellerons une k-Ham partition) en contraignant chaque sommet à avoir exactement deux arêtes incidentes dans $\mathcal{C}$ (voir Fig. 4.23). La deuxième idée est de forcer $k = 1$ (i.e. forcer $\mathcal{C}$ à être un cycle Hamiltonien) en s'aidant du graphe dual G' : $\mathcal{C}$ induit une structure $\mathcal{D}$ dans le graphe dual G', qui est le graphe des faces non-déconnectées par $\mathcal{C}$.

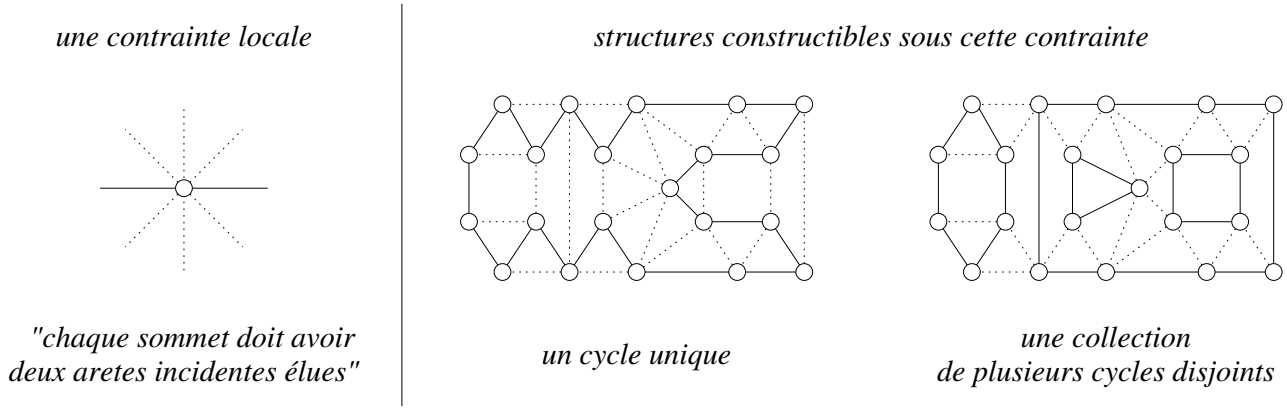


FIG. 4.23 – Idée 1 : Obtenir une collection de cycles simples disjoints

Le Lemme 4.24 montre que $\mathcal{C}$ est un cycle Hamiltonien ssi $\mathcal{D}$ est une forêt constituée d'exactement deux arbres. Donc, notre système SAT cherche naturellement à contraindre toutes les faces (à l'exception de deux) à élire exactement un père parmi leurs faces adjacentes dans une même composante connexe de $\mathcal{D}$. Comme il n'y a que deux racines (faces sans père), $\mathcal{D}$ contient exactement deux arbres mais $\mathcal{D}$ peut toujours contenir aussi des composantes connexes indésirables : les monocycles, qui sont formés d'un cycle dont chaque sommet peut éventuellement être la racine d'un arbre. Comme ces monocycles n'ont pas de racine, notre système SAT se trouve apparemment incapable de les détecter (voir Fig. 4.24).

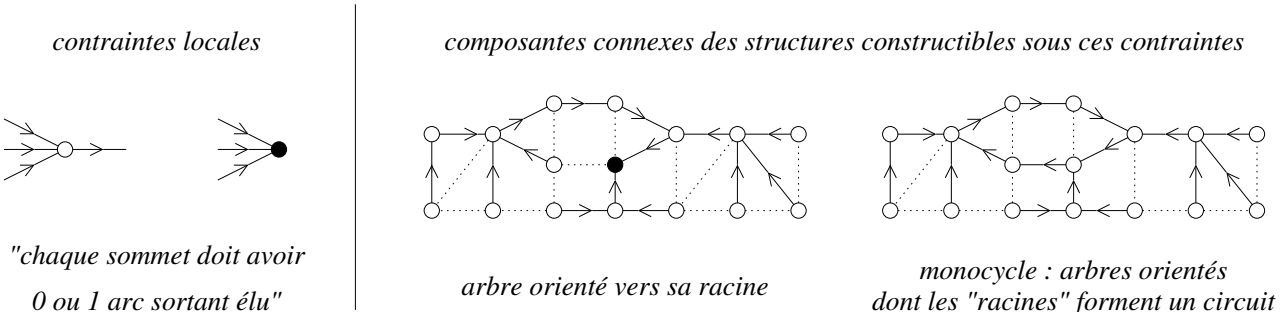


FIG. 4.24 – Idée 2 : Obtenir une structure d'arbres et de monocycles dans le graphe dual

Heureusement, le Lemme 4.26 montre que tout cycle dans $\mathcal{D}$ empêche les deux racines d'être adjacentes. Donc, en forçant simplement nos deux racines à être adjacentes, nous interdisons indirectement la formation de cycles dans $\mathcal{D}$, et nous obtenons une réduction linéaire de PLAN-UHAM-CYCLE à SAT (voir Fig. 4.25). Cette réduction est même parcimonieuse du fait que l'on est capable de choisir les racines *a priori*, indépendamment du cycle Hamiltonien.

Enfin, grâce au crossover parcimonieux pour PLAN-SAT il est facile de faire cohabiter le sous-système exerçant ses contraintes sur le graphe primal avec le sous-système exerçant les siennes sur le graphe dual en n'utilisant qu'un nombre linéaire de ces crossovers. Nous avons ainsi une réduction linéaire et parcimonieuse de PLAN-UHAM-CYCLE à PLAN-SAT.

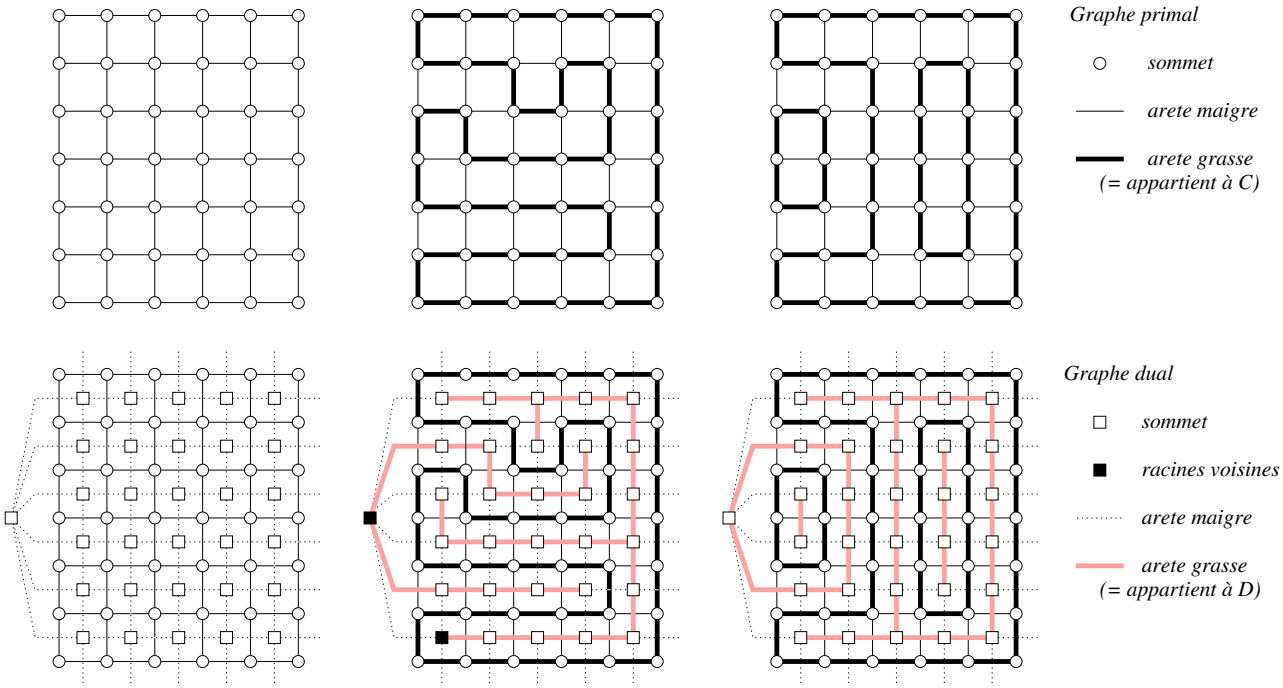


FIG. 4.25 – Idée 3 : Empêcher l'apparition de monocycles dans le dual

4.3.1 Une caractérisation locale de l'Hamiltonicité planaire

Fait 4.14 (Théorème de la courbe fermée de Jordan, généralisé) *Soit $\mathcal{C}$ une collection de k courbes simples fermées $C_1, \cdots, C_k$, disjointes deux à deux, plongées sur une sphère S. Alors $S \setminus \mathcal{C}$ est découpé en exactement $k+1$ régions connexes maximales (ou régions, pour faire court).*

On se donne maintenant un graphe connexe planaire $G(V, E, F)$ plongé sur une sphère S, où V (resp. E, F) est l'ensemble de ses sommets (resp. arêtes, faces).

Définition 4.15 (co-dual, Ham-dual, composantes) *Soit $G' = dual(G)$ et $\mathcal{C}$ un ensemble d'arêtes grasses dans G. Le co-dual $\mathcal{D}$ de $\mathcal{C}$ est défini comme étant G' après la suppression de toutes les arêtes $e' = dual(e)$ telles que e est grasse dans G, i.e. appartient à $\mathcal{C}$. Les arêtes dans $\mathcal{D}$ (resp. dans $G' \setminus \mathcal{D}$) sont appelées les arêtes grasses (resp. maigres) de G'. En d'autres termes, $e' = dual(e)$ est grasse (maigre) ssi e est maigre (grasse). Les* composantes connexes maximales *dans $\mathcal{D}$ sont appelées simplement* composantes. *Le co-dual d'une k-Ham-partition est appelé* Ham-dual.

A partir de maintenant, on suppose donnée une k-Ham partition $\mathcal{C}$ dans G, ainsi que son Ham-dual $\mathcal{D}$ dans $G' = dual(G)$.

Lemme 4.16 *Le Ham-dual $\mathcal{D}$ a exactement $k' = k+1$ composantes.*

Preuve Ceci découle immédiatement du Fait 4.14. ■

Définition 4.17 (C-régions) *Soit C un cycle simple dans G ou G'. D'après le fait 4.14, $S \setminus C$ est constitué de deux régions R_1 et R_2, que l'on appelle* C-régions, *telles que R_1, R_2, et C forment une partition de la sphère S.*

Définition 4.18 (C-cordes, C-arêtes, R-arêtes) *Soit C un cycle simple dans G ou son dual G', R une C-région, et $e = (x, y) \in E$ une arête plongée dans $C \cup R$ telle que $x \in C$. Si $y \in R$, alors e est appelée* R-arête. *Sinon, $y \in C$ et e est appelée une C-arête (resp. une C-corde) si $e \in C$ (resp. si $e \notin C$).*

Notons que si C appartient à $\mathcal{C}$, alors toutes ses C-arêtes sont grasses, et toutes ses R-arêtes et C-cordes sont maigres.

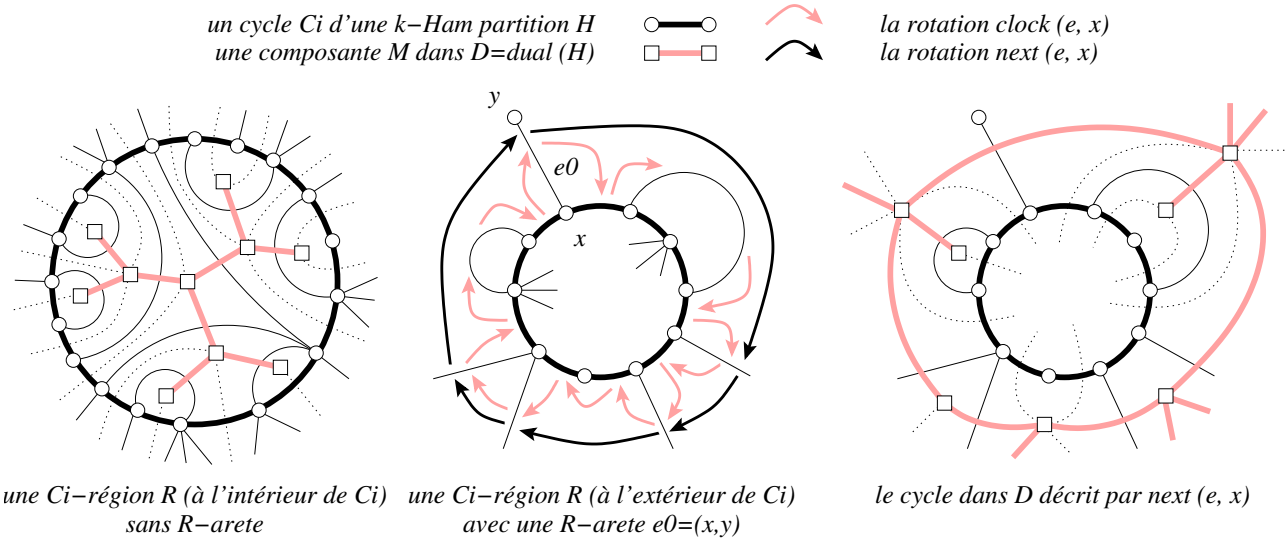


FIG. 4.26 – Composantes cycliques et acycliques dans $\mathcal{D}$

Lemme 4.19 *Soit C_i un cycle dans $\mathcal{C}$ et R une C_i-région telle qu'il n'existe pas de R-arête. Alors l'unique composante M de $\mathcal{D}$ plongée dans R est acyclique, i.e., est un arbre (voir Fig. 4.26 à gauche).*

Preuve Comme il n'existe aucune R-arête et que G est planaire et connexe, le sous-ensemble $V_R \subset V$ de sommets plongés dans R est vide. Ainsi, R ne peut contenir que des C_i-cordes. Maintenant, supposons que M a un cycle C'. Soit e' une arête arbitraire de C', et $e = (x, y) = primal(e')$. Remarquons que x et y sont plongés dans deux C'-régions distinctes. Comme exactement l'une des deux C'-régions ne contient pas C_i, soit x, soit y doit être plongé dans R, ce qui contredit $V_R = \emptyset$. ■

Lemme 4.20 *Soit C_i un cycle dans $\mathcal{C}$, et R une C_i-région telle qu'il existe une R-arête $e_0 \in E$. Alors la composante M de $\mathcal{D}$ contenant $e'_0 = dual(e_0)$ est cyclique (voir Fig. 4.26 au centre et à droite).*

Preuve Pour toute arête $e = (x, y)$, $clock(e, x)$ dénote l'arête immédiatement après e dans le sens anti-trigonométrique (sens des aiguilles d'une montre) autour de x. De plus, $other(e, x)$ dénote y (l'autre sommet de x) et vice-versa. Soit $I(e_0)$ l'itération :

```
e := e0(x,y);
loop {
   e := clock (e, x);
   write (e);
   x := other(e, x);
}
```

Si l'on stoppe $I(e_0)$ aussitôt que e_0 est à nouveau rencontré, alors $I(e_0)$ écrit la frontière d'une face f dans le sens trigonométrique. Cette face f, notée $left(e_0, x)$, est la face à main gauche pour un observateur situé en y et regardant vers x. Maintenant, supposons que e_0 est une R-arête de C_i avec $x \in \mathcal{C}$. Alors, la première R-arête e rencontrée après e_0 dans $I(e_0)$ est notée $e = next(e_0, x)$. Ainsi, e_0 et $next(e_0, x)$ partagent la même face adjacente $f = left(e_0, x)$.

Si C_i est un cycle de $\mathcal{C}$ et si e est une R-arête de C_i, alors on désignera par $common(e, C_i)$ l'unique sommet commun de e et C. Soit $J(e_0, C_i)$ l'itération :

```
e := e0(x,y) ;
repeat {
   write (left (e, x), dual (e)) ;
   e := next (e, x) ;
   x := common (e, Ci) ;
} until (e = e0) ;
```

La boucle termine à cause de l'invariant $x \in C_i$. Comme chaque arête e rencontrée partage une face commune $left(e, x)$ avec $next(e, x)$, $J(e_0, C_i)$ écrit les sommets et arêtes successifs d'un cycle C' dans G' (éventuellement non-simple si G n'est pas biconnexe et qu'un point d'articulation x est rencontré). Puisque toutes les arêtes de C' sont les arêtes duales de R-arêtes, et que ces R-arêtes sont toujours maigres dans G, il suit que les arêtes de C' sont toutes grasses dans G'. Ce qui signifie qu'une composante M plongée dans R et contenant $e'_0 = dual(e_0)$ contient aussi le cycle C'. ■

A partir de maintenant, on choisit arbitrairement une face $f_{out} \in F$, que l'on l'appelle la *face externe.*

Définition 4.21 (intérieur, extérieur) *Soit C un cycle simple dans G, et R_1, R_2 deux C-régions. Exactement l'une d'elles, disons R_2, contient la face externe f_{out}. Alors R_1 (resp. R_2) est appelée* l'intérieur *(resp.* l'extérieur*) de C et on la note $int(C)$ (resp. $ext(C)$).*

Définition 4.22 (cycles emboîtés, min-cycles, max-cycles) *Pour toute paire de cycles C_i, C_j dans $\mathcal{C}$ tels que $int(C_i) \subset int(C_j)$, on note $C_i < C_j$ (et on lit : C_i est emboîté dans C_j, ou C_j emboîte C_i). De plus, s'il n'existe pas de cycle $C_h \in \mathcal{C}$ tel que $C_i < C_h < C_j$, alors on note $C_j = succ(C_i)$ et $brothers(C_i) = \{C_j \in \mathcal{C} : succ(C_j) = succ(C_i)\}$. Enfin, un min-cycle (resp. max-cycle) C_i est un cycle de $\mathcal{C}$ qui n'emboîte (resp. n'est emboîté dans) aucun autre cycle de $\mathcal{C}$.*

Lemme 4.23 *Soit M une composante de $\mathcal{D}$. Alors M est un arbre ssi :*

1. *M est plongé dans l'intérieur d'un min-cycle C_i de $\mathcal{C}$, ou*
2. *M est plongé dans l'extérieur d'un cycle C_j qui est l'unique max-cycle de $\mathcal{C}$.*

Preuve Il y a exactement quatre cas :
- Si M est plongé dans $R = int(C_i)$ et si C_i est un min-cycle de $\mathcal{C}$, alors il n'existe pas de R-arête et le Lemme 4.19 s'applique à C_i et R : M est un arbre.
- Si M est plongé dans $R = ext(C_j)$ et si C_j est l'unique max-cycle de $\mathcal{C}$ alors, de même, il n'existe pas de R-arête et le Lemme 4.19 s'applique encore à C_j et R, et M est un arbre.

- Si M est plongé dans $R = ext(C_i)$ et si C_i est un max-cycle non-unique dans $\mathcal{C}$ alors, comme G est planaire et connexe, il existe $e = (x, y) \in E$ telle que $x \in C_i$ et $y \in C_j$ pour un max-cycle donné $C_j \neq C_i$. Puisque e est une R-arête, Le Lemme 4.20 s'applique à C_i et $R : M$ a un cycle.
- Sinon, M est plongé dans $ext(C_i) \cap int(C_j)$, pour un $C_j = succ(C_i)$ donné. Comme G est planaire est connexe, il existe $e = (x, y) \in E$ telle que $x \in C_i$ et $y \in C_j$ ou $y \in C_h$ avec $C_h \in brothers(C_i)$, $h \neq i$. Dans les deux cas, e est une R-arête pour $R = ext(C_i)$, et donc le Lemme 4.20 s'applique à C_i et $R : M$ a un cycle.

■

Lemme 4.24 *$k = 1$, i.e., $\mathcal{C}$ est un cycle Hamiltonien ssi $\mathcal{D}$ est constitué d'exactement 2 arbres.*

Preuve ($\Longrightarrow$) : Si $k = 1$, alors soit C_1 l'unique cycle dans $\mathcal{C}$: Par le lemme 4.16, $\mathcal{D}$ a $k' = k + 1 = 2$ composantes M_1 et M_2, plongées resp. dans $int(C_1)$ et $ext(C_1)$. Comme C_1 est à la fois un min-cycle et l'unique max-cycle dans $\mathcal{C}$, le Lemme 4.23 s'applique deux fois, et M_1 et M_2 sont tous deux des arbres.

($\Longleftarrow$) : C'est une conséquence immédiate du Lemme 4.16. ■

Définition 4.25 (Composantes jumelles) *Deux composantes M_1 et M_2 dans $\mathcal{D}$ sont dites* jumelles *ssi les deux conditions sont vérifiées :*

1. *M_1 et M_2 sont toutes deux des arbres, et*
2. *Il existe deux sommets $r_1 \in M_1$ et $r_2 \in M_2$, appelées* racines jumelles, *tels que $(r_1, r_2) \in E'$*

Lemme 4.26 *$k = 1$, i.e., $\mathcal{C}$ est un cycle Hamiltonien ssi deux composantes jumelles existent dans $\mathcal{D}$.*

Preuve $\Longrightarrow$: C'est une conséquence immédiate du Lemme 4.24. Il suffit d'exhiber deux racines jumelles r_1 et r_2 : Choisir une arête arbitraire e dans l'unique cycle $C_1 \in \mathcal{C}$. Soit $e' = (f, g) = dual(e)$ tel que f soit plongé dans $int(C_1)$ et g dans $ext(C_1)$. Poser $r_1 = f$ et $r_2 = g$.

($\Longleftarrow$) : Soient M_1 et M_2 deux composantes jumelles de racines respectives r_1 et r_2. Puisque les deux composantes sont des arbres, alors par le Lemme 4.23, chacune doit être plongée soit à l'intérieur d'un min-cycle, soit à l'extérieur d'un unique max-cycle. Supposons que $k > 1$, alors il y a deux cas :

- $M_1 \in int(C_i)$ et $M_2 \in int(C_j)$ où C_i et C_j sont des min-cycles disjoints dans $\mathcal{C}$. Comme G est planaire et C_i est disjoint de C_j, tout chemin de r_1 à r_2 dans G' doit contenir un sommet intermédiaire plongé dans $ext(C_i) \cap ext(C_j)$. Donc, r_1 et r_2 ne sont pas des racines jumelles, une contradiction.
- $M_1 \in int(C_i)$ et $M_2 \in ext(C_j)$, où C_i est un min-cycle et C_j est l'unique max-cycle dans $\mathcal{C}$. Comme $k > 1$, nous avons $C_i \neq C_j$. Puisque G est planaire, C_i est disjoint de C_j, tout chemin de r_1 à r_2 dans G' contient un troisième sommet plongé dans $ext(C_i) \cap int(C_j)$, et donc r_1 et r_2 ne sont pas racines jumelles.

■

4.3.2 La réduction linéaire de PLAN-UHAM-CYCLE à PLAN-SAT

Comme nous souhaitons que notre réduction linéaire soit également parcimonieuse, il nous faut fixer les deux racines jumelles r_1 et r_2 à l'intérieur de deux faces prédéterminées de G. D'après la preuve du Lemme 4.26, r_1 et r_2 peuvent être arbitrairement choisies parmi n'importe quelle paire de faces adjacentes séparées par une arête grasse dans G. Afin de trouver une telle arête, nous choisissons un sommet $v \in V$ de degré d quelconque, et nous l'explosons en le graphe $G(v)$ de la Fig. 4.27 en forme de roue de vélo à d rayons. Une correspondance bijective suggérée sur cette même figure (au centre et à droite) est vérifiée entre les cycles Hamiltoniens dans G avant et après la substitution, et G contient maintenant un sommet t de degré 2 (choisi arbitrairement parmi les d sommets de degré 2 sur la circonférence de la roue), ce qui force ses deux arêtes incidentes à être grasses pour tout cycle Hamiltonien dans G. Ainsi, r_1 et r_2 peuvent être fixées comme étant les deux faces adjacentes de t.

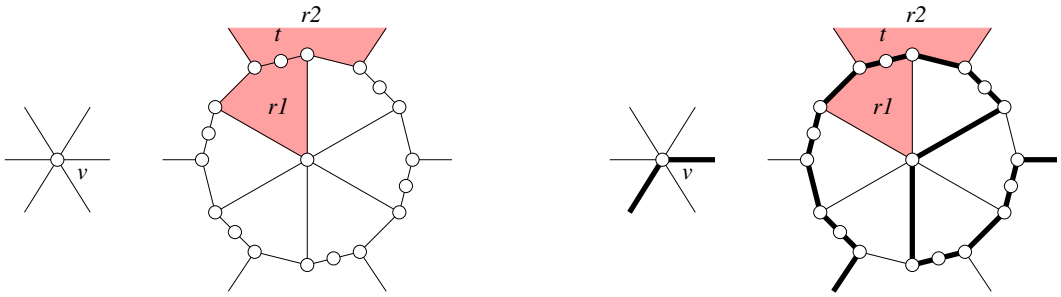


FIG. 4.27 – Explosion d'un sommet de degré $d = 6$ en une roue de vélo à d rayons

Pour des raisons de clarté, nous supposons que nous disposons des deux clauses spéciales $1/N(\ell_1, \cdots, \ell_d)$ et $2/N(\ell_1, \cdots, \ell_d)$ qui sont satisfaites ssi exactement 1 littéral (resp. 2 littéraux) parmi les ℓ_i $(1 \leq i \leq d)$ sont mis à vrai. Implémenter ces deux contraintes parcimonieusement avec un système de $O(d)$ clauses planaires est une tâche assez facile dont le détail est relégué en fin de section. Notre réduction associe à tout graphe planaire $G(V, E, F)$ une formule $\varphi(G)$ construite de la façon suivante :

- *Ensemble de variables de $\varphi(G)$:* Chaque arête $e \in E \cup E'$ a une variable booléenne associée $thick_e$, qui exprime que e est grasse dans $\mathcal{C} \cup \mathcal{D}$. Le choix des racines r_1 et r_2 détermine une direction pour la forêt $\mathcal{D}$ où chaque noeud pointe vers son père : chaque sommet-face $f \in V'$ de degré d possède d variables booléennes associées $father_f^{e'}$, une pour chaque arête $e' = (f, g) \in E'$, et qui est vraie ssi $e' \in \mathcal{D}$ et g est le père de f dans $\mathcal{D}$.

$\varphi(G)$ est la conjonction des formules exprimant les contraintes suivantes :

- *"$\mathcal{C}$ est une k-Ham partition de G" :* Pour chaque sommet $v \in V$ de degré d ayant pour arêtes incidentes $e_1, \cdots, e_d$, on génère la contrainte $2/N(thick_{e_1}, \cdots, thick_{e_d})$.
- *"$\mathcal{D}$ est le Ham-dual de $\mathcal{C}$" :* Pour chaque arête $e \in E$ avec $e' = dual(e)$, on génère les clauses $(thick_e \vee thick_{e'})$ et $(\neg thick_e \vee \neg thick_{e'})$.
- *"Toute face hormis r_1 et r_2 a exactement un père" :* Pour chaque sommet-face $f \in V'$ de degré d, $f \notin \{r_1, r_2\}$, ayant pour arêtes incidentes $e'_1, \cdots, e'_d$, on génère la contrainte

$$1/N(father_f^{e'_1}, \cdots, father_f^{e'_d})$$

– *"Les deux racines jumelles r_1 et r_2 n'ont pas de père"* : Pour chaque arête $e' \in E'$ incidente à une racine $r \in \{r_1, r_2\}$, on génère la clause unitaire

$$(\neg father_r^{e'})$$

– *"L'orientation du Ham-dual $\mathcal{D}$ est cohérente"* : Pour chaque arête $e' = (f, g) \in E'$, on génère les contraintes

$$(father_f^{e'} \Longrightarrow thick_{e'})$$
$$(father_g^{e'} \Longrightarrow thick_{e'})$$
$$(thick_{e'} \Longrightarrow father_f^{e'} \vee father_g^{e'})$$
$$(\neg father_g^{e'} \vee \neg father_f^{e'})$$

La formule $\varphi(G)$ est de taille linéaire en $|G + G'|$ et sa correction est une conséquence immédiate des Lemmes 4.24 et 4.26. La parcimonie découle de la possibilité de fixer r_1 et r_2 dans deux faces voisines choisies. De plus, la Fig. 4.28 montre comment plonger le système SAT dans le plan au niveau de chaque face en utilisant un nombre linéaire de crossovers parcimonieux pour PLAN-SAT, ce qui termine la preuve de la proposition 4.1, sous l'hypothèse qu'on sait implémenter les contraintes $1/N$ et $2/N$ linéairement et parcimonieusement dans le plan.

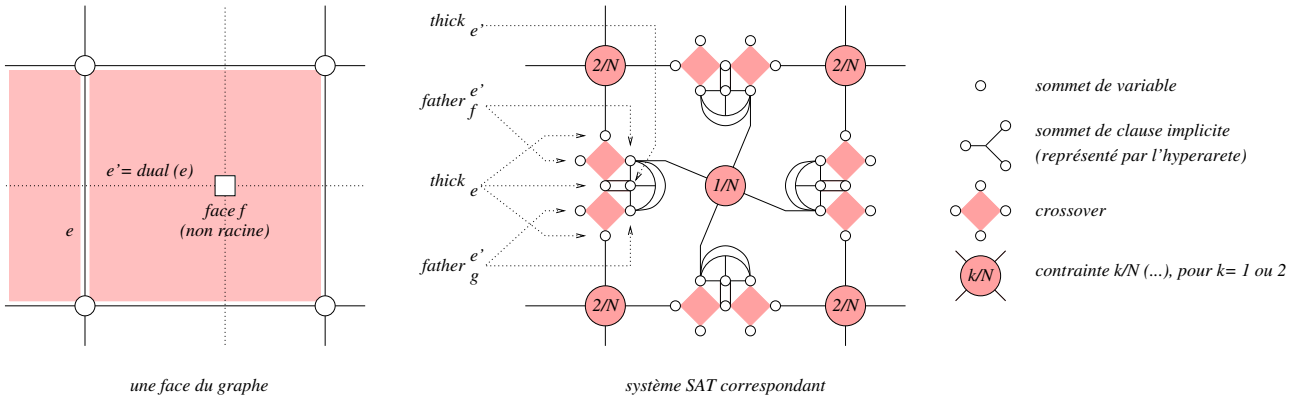


FIG. 4.28 – Plongement du système SAT dans le plan au niveau d'une face

4.3.3 Implémentation des contraintes 1/N et 2/N dans le plan

Nous codons la contrainte $1/N(x_1, \cdots, x_d)$ en simulant une paire de suites booléennes $([a_k], [x_k])$ de longueur d, où $[a_k]$ est une suite monotone non-croissante débutant à $a_0 = 1$, et où $x_k = 1$ ssi k est le dernier rang tel que $a_k = 1$, e.g., $[a_k] = [1, 1, 1, 0, 0, 0]$ et $[x_k] = [0, 0, 1, 0, 0, 0]$. La monotonie $[a_k]$ est codée par la clause unitaire (a_1) ainsi que les clauses $(a_{k+1} \Longrightarrow a_k)$, pour tout $k < d$, qui ne signifie rien d'autre que $a_{k+1} \leq a_k$.

La suite $[x_k]$ est codée par $(x_d \Longleftrightarrow a_d)$ et pour tout $k < d$ par $(x_k \Longleftrightarrow (a_k \wedge \neg a_{k+1}))$ dont le développement en clauses est $(\neg x_d \vee a_d)$, $(\neg a_d \vee x_d)$, $(\neg x_k \vee a_k)$, $(\neg x_k \vee \neg a_{k+1})$, $(\neg a_k \vee a_{k+1} \vee x_k)$. Un plongement planaire de $1/N(\cdots)$ est la chaîne de tétraèdres de la Fig. 4.29.

Le codage de la contrainte $2/N(x_1, \cdots, x_d)$ suit la même idée. Nous simulons deux paires de suites booléennes finies $([a_k], [y_k])$ et $([b_k], [z_k])$ comme présenté plus haut, telles que $[y_k]$ et $[z_k]$ contiennent exactement un "un". Ensuite, nous définissons le vecteur $[x_k]$ comme le "ou bit-à-bit" des vecteurs $[y_k]$ et $[z_k]$, ce qui se code par $(x_k \Longleftrightarrow y_k \vee z_k)$ pour tout $k \leq d$, et se développe en les clauses $(\neg x_k \vee y_k \vee z_k)$, $(\neg y_k \vee x_k)$, $(\neg z_k \vee x_k)$.

Pour que $[x_k]$ contienne deux "un" et non un seul, il faut de plus contraindre que $y_i = 1$ et $z_j = 1$ pour deux indices distincts i, j. L'une des sous-séquences de "uns" initiale dans $[a_i]$ et $[b_i]$ doit être strictement plus longue que l'autre. Afin de rester parcimonieux, nous choisissons que ce soit toujours la sous-séquence de $[a_k]$ qui soit strictement plus courte que celle de $[b_k]$. Ceci se fait par l'ajout des clauses $(\neg a_k \vee b_k)$ et $(\neg y_k \vee \neg z_k)$ pour tout $k \leq d$.

Un plongement "presque" planaire pour la simulation de $2/N(\cdots)$ est la triple chaîne de tétraèdres de la Fig. 4.30. "Presque", parce que la séquence $[x_k]$ ne se trouve pas le long de la face externe comme il se doit pour un gadget planaire. Ceci se corrige en rajoutant d crossovers parcimonieux afin de faire évader la séquence $[x_k]$ vers la frontière de la face externe (voir Fig. 4.31).

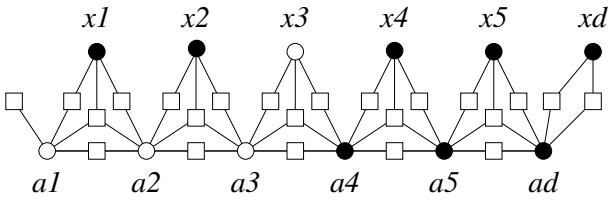


FIG. 4.29 – Codage planaire, linéaire et parcimonieux de la contrainte $1/N(x_1, \cdots, x_d)$

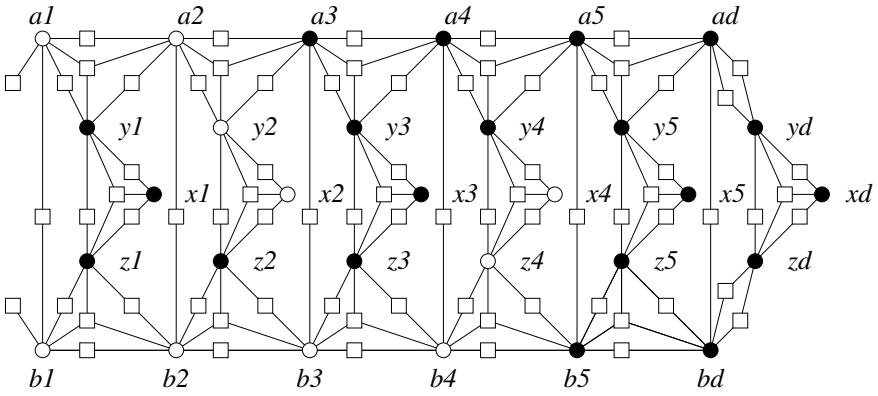


FIG. 4.30 – Codage presque planaire d'une contrainte $2/N(x_1, \cdots, x_d)$

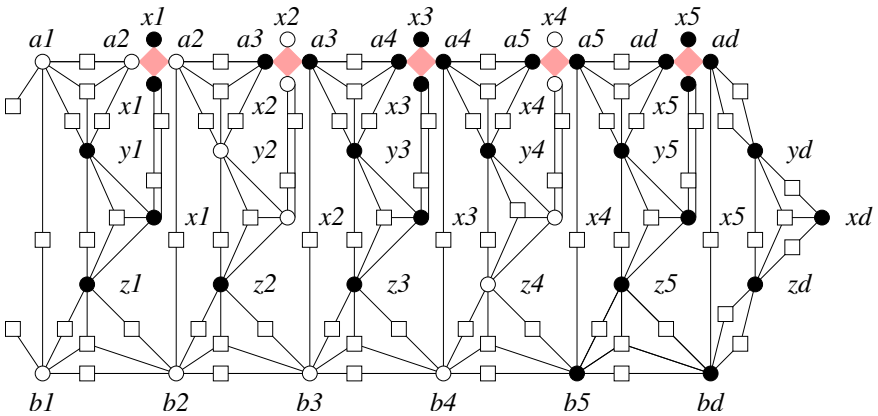


FIG. 4.31 – Codage planaire, linéaire et parcimonieux d'une contrainte $2/N(x_1, \cdots, x_d)$

4.4 SAT et les problèmes de cardinalité

Nous terminons ce chapitre en montrant brièvement deux réductions linéaires et parcimonieuses entre VERTEX-COVER et SAT. Pour la réduction de SAT à VERTEX-COVER, nous partons encore du problème intermédiaire $\frac{1}{3}$-SAT.

Soit $\varphi(V, L)$ le système $\frac{1}{3}$-SAT à réduire à VERTEX-COVER. On construit un graphe G de la manière suivante : Pour chaque variable $v \in V$ de degré $d(v)$ dans le graphe de formule G_φ (i.e., avec $d(v)$ occurrences dans φ), on créé un $4d(v)$-cycle $(v_1^k, v_2^k, v_3^k, \cdots)_{1 \leq k \leq d(v)}$. Pour chaque $\frac{1}{3}$-clause $(a, b, c) \in L$, on choisit trois 4-chemins $(a_1^i, a_2^i, a_3^i, a_4^i)$, $(b_1^j, b_2^j, b_3^j, b_4^j)$, $(c_1^k, c_2^k, c_3^k, c_4^k)$, encore non-utilisés dans les cycles correspondants, et on les connecte avec trois nouveaux sommets A, B, C, comme montré sur la Fig. 4.32.

Lemme 4.27 *Les couvertures dans G de cardinalité $\leq 8|L|$ sont alors en bijection avec les assignements satisfaisant φ.*

> **Preuve** Soit K une couverture de G ; Colorions les éléments de K en blanc, et les autres sommets en noir. Pour couvrir un cycle de longueur paire $2n$, il faut au moins n sommets blancs, et la seule solution pour atteindre cette borne est un bicoloriage alternant blanc/noir. Aussi, pour couvrir un triangle, il faut au moins deux sommets blancs dans ce triangle. Tous ces triangles et cycles sont disjoints. La "couverture" de tous les cycles associés aux variables nécessite donc un nombre total de sommets blancs au moins égal à $\frac{1}{2}\sum_{v\in V} 4d(v) = 2\sum_{v\in V} d(v) = 2\times 3|L| = 6|L|$, et le coloriage des triangles nécessite au moins $2|L|$ sommets blancs, soit au total au moins $8|L|$ sommets dans K. La limite imposée est déjà atteinte, et donc ce schéma de coloriage est le seul possible.
>
> Le gadget simulant une $\frac{1}{3}$-clause (a, b, c) étant symétrique par rapport à a, b, c, on peut supposer que les deux sommets blancs du triangle sont A et C (les deux autres possibilités donneront les deux autres configurations). B étant noir, b_3^j et c_2^k doivent être blancs pour couvrir leur arête commune avec B. La règle du bicoloriage alternant sur les cycles de variables implique ensuite que b_2 et b_4 sont noirs et b_1 est blanc. De même c_1 et c_3 sont noirs et c_4 est blanc. Ensuite, le sommet b_2 étant noir, a_4 doit être blanc pour couvrir leur arête commune. La règle du bicoloriage alternant sur les cycles de variables implique ensuite que a_1 et a_3 sont noirs et a_2 est blanc. On vérifie facilement que toutes les arêtes du gadget simulant la $\frac{1}{3}$-clause sont bien couvertes.
>
> Exactement un cycle de variable parmi les trois, ici le cycle de la variable b a ses sommets d'indices impairs coloriés en blanc, ce qui établit la bijection avec les trois assignements satisfaisant une $\frac{1}{3}$-clause. ■

Par ailleurs, on voit sur la Fig. 4.32 que si G_φ est planaire alors le graphe G construit par notre réduction est aussi planaire, et donc SAT et PLAN-$\frac{1}{3}$-SAT se réduisent bien linéairement et parcimonieusement resp. à VERTEX-COVER et à PLAN-VERTEX-COVER, comme annoncé dans la Prop. 4.5.

Nous n'esquissons que rapidement l'idée de la réduction linéaire et parcimonieuse reciproque de VERTEX-COVER à SAT : Soit $G(V, E)$ le graphe de l'instance d'entrée et k la cardinalité maximale autorisée pour la couverture. Il est clair que si l'on génère la clause $x \vee y$ pour toute arête $(x, y) \in E$, on exprime que toute arête doit être couverte. Il reste à créer un gadget de taille $O(|V|)$ calculant le nombre de variables vraies parmi $|V|$ et stockant le résultat en binaire dans $\log|V|$ variables. Contraindre ensuite ce poids à être inférieur à k se fait facilement en $\log|V|$ en codant un circuit arithmétique avec SAT.

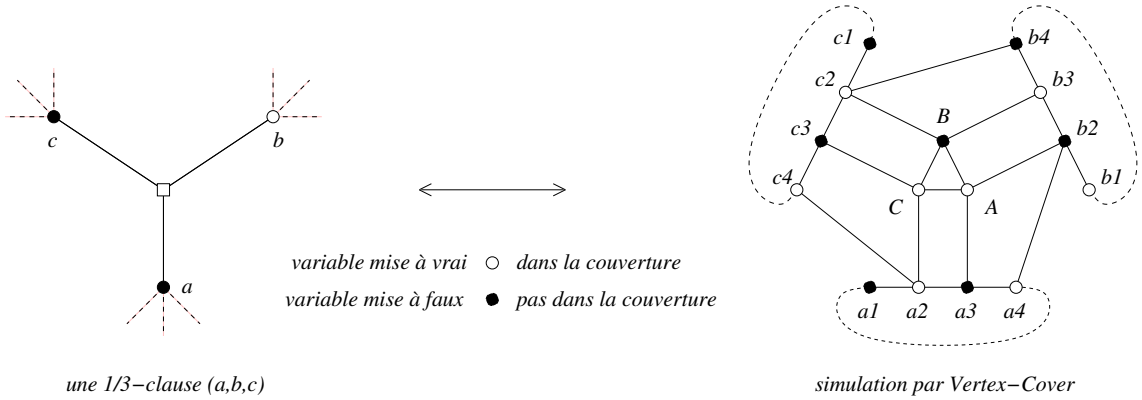


FIG. 4.32 – Réduction linéaire, parcimonieuse et planaire de $\frac{1}{3}$-SAT à VERTEX-COVER

Pour obtenir un additionneur de taille $O(|V|)$, il suffit d'effectuer un diviser-pour-régner sur les additions : Supposons pour simplifier que $n = |V|$ est une puissance exacte de 2, i.e., $n = 2^\ell$. On effectue les additions niveau par niveau, avec $\ell + 1$ niveaux (de 0 à ℓ) : Le niveau 0 consiste en une liste de 2^ℓ nombres de 1 bit $(X_0^0, \cdots, X_0^{2^\ell - 1})$, et qui sont les variables même que l'on veut compter. Pour tout $1 < k \leq \ell$, le niveau k est constitué de $2^{\ell-k}$ nombres $(X_k^0, \cdots, X_k^{2^{\ell-k}-1})$, tels que $X_k^j = X_{k-1}^{2j} + X_{k-1}^{2j+1}$. Comme la somme de deux nombres de b bits tient dans $b + 1$ bits, chaque nombre de niveau k a $k + 1$ bits.

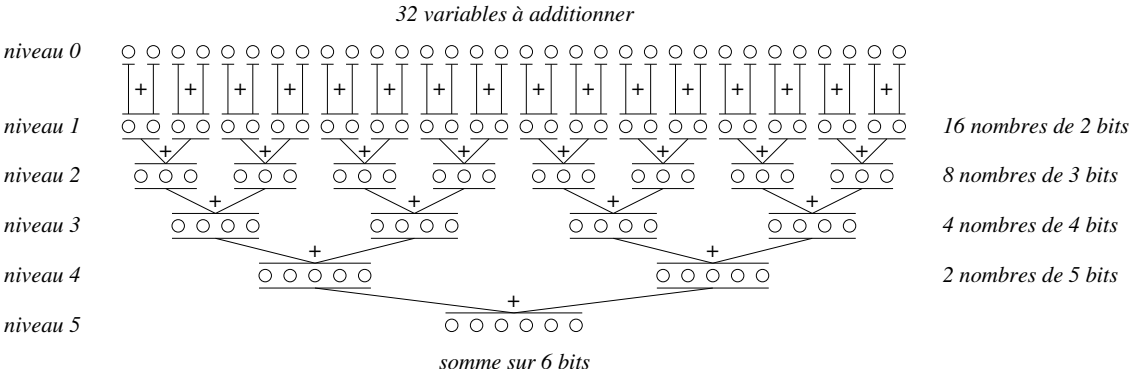


FIG. 4.33 – Un schéma simple d'additionneur linéaire

Ainsi, la liste de niveau ℓ est constituée d'un seul nombre X_ℓ^0 de $\ell+1$ bits contenant le résultat désiré. Coder toutes ces additions binaires dans SAT avec un schéma classique de propagation des retenues prend un temps et un espace $O(s(n))$ où $s(n)$ est le nombre total de bits sur tous les niveaux, or :

$$\begin{aligned} s(n) &= \sum_{k=0}^{\ell}(k+1)2^{\ell-k} \\ &= \sum_{\ell'=0}^{\ell}\sum_{k=0}^{\ell'} 2^k \\ &= \sum_{\ell'=0}^{\ell}(2^{\ell'+1}-1) \\ &= 2\sum_{\ell'=0}^{\ell} 2^{\ell'} - (\ell+1) \\ &= 2(2^{\ell+1}-1) - (\ell+1) \\ &= 4n - (\ell+3) \\ &= O(n) \end{aligned}$$

Le gadget est bien linéaire et est facilement implémentable parcimonieusement, ce qui conclut la description de cette réduction.

De façon générale, ce schéma d'additionneur linéaire permet de réduire linéairement et parcimonieusement à SAT tous les problèmes de contraintes locales soumis à une contrainte globale de cardinalité :

Corollaire 4.28 *Tous les problèmes "de contraintes locales avec cardinalité" tels que* MAX-SAT, DOMINATING-SET, VERTEX-COVER, *etc. sont linéairement et parcimonieusement réductibles à* SAT.

5

Réductions planaires

ou la préservation de la géométrie des instances

Sommaire

5.1 Introduction 94
5.2 L'équivalence des variantes non-planaires de Hamilton 94
5.3 L'équivalence des variantes de Plan-Hamilton 97
5.3.1 Variantes non-orientées : le gadget petit chandelier 98
5.3.2 Variantes orientées : le gadget grand chandelier 98
5.3.3 Gadgets fonctionnels pour l'arithmétique 99
5.3.4 Implémentation du petit chandelier (cadre non-orienté) 100
5.3.5 Implémentation du grand chandelier (cadre orienté) 102
5.4 Variantes non-orientées de Plan-Hamilton à degré borné à 3 104

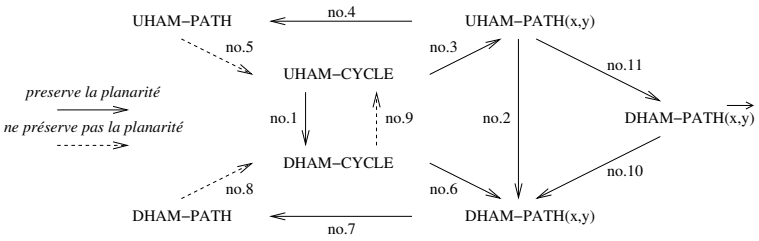


FIG. 5.1 – Schéma de d'inter-réductions entre variantes de HAMILTON

5.1 Introduction

Ce chapitre établit l'équivalence linéaire et parcimonieuse de toutes les variantes du problème PLAN-HAMILTON qui ont été citées dans les préliminaires, renforçant ainsi le résultat principal du chapitre précédent : l'équivalence linéaire et parcimonieuse de PLAN-SAT et PLAN-UHAM-CYCLE.

Proposition 5.1 *Toutes les variantes de* PLAN-HAMILTON *sont linéairement et parcimonieusement équivalentes à* PLAN-SAT.

Par ailleurs, nos réductions aux variantes non-orientées de PLAN-HAMILTON produiront des graphes de degré maximal borné à 4, dans un premier temps, puis à 3, et par conséquent :

Proposition 5.2 *Toutes les variantes non-orientées de* PLAN-HAMILTON *sur les graphes de degré maximal 3 sont linéairement et parcimonieusement équivalentes à* PLAN-SAT.

Nous commençons par présenter les réductions triviales entre les différentes variantes de HAMILTON. Certaines ne préservent pas la planarité. Nous présentons ensuite des réductions plus élaborées qui la préservent tout en restant linéaires et parcimonieuses.

5.2 L'équivalence des variantes non-planaires de HAMILTON

Il est facile d'inter-réduire linéairement et parcimonieusement toutes les variantes de HAMILTON définies lors des préliminaires si l'on n'exige pas que le graphe d'arrivée soit planaire ou à degré borné. La Fig. 5.1 montre le graphe des onze réductions élémentaires établissant l'équivalence des six problèmes UHAM-PATH, UHAM-CYCLE, UHAM-PATH(x, y), et DHAM-PATH, DHAM-CYCLE, DHAM-PATH(x, y). On constate que ce graphe des réductions est fortement connexe comme il se doit pour établir l'équivalence des problèmes, mais qu'il ne l'est plus dès lors que l'on supprime l'un des trois arcs en pointillé qui représentent les réductions ne préservant pas la planarité. Ces réductions simples ne permettent donc pas d'établir l'équivalence linéaire et parcimonieuse des variantes planaires de HAMILTON.

Nous présentons brièvement ces onze réductions dans l'ordre de numérotation du graphe de la Fig. 5.1, et nous proposons ensuite un schéma de réduction qui reste valable dans le plan.

Réduction no.1 : de UHAM-CYCLE **à** DHAM-CYCLE Soit $G(V, E)$ le graphe non-orienté de départ. Pour obtenir le graphe d'arrivée G', il suffit simplement de remplacer chaque arête (u, v) de G par deux arcs anti-parallèles (u, v) et (v, u). En l'état, la réduction est linéaire et préserve la planarité mais elle double le nombre de solutions car à chaque cycle Hamiltonien non-orienté correspondent deux circuits de sens opposés. Pour supprimer les doublons, il suffit de privilégier

un sens sur l'autre. Ceci s'effectue de la façon suivante : on choisit arbitrairement un sommet v de degré 2 dans G. On peut toujours supposer qu'un tel sommet existe grâce au gadget en "roue de vélo" de la Fig. 4.27 présenté dans le chapitre précédent, qui ne modifie pas le nombre de cycles Hamiltoniens dans un graphe non-orienté. Ses deux arêtes incidentes (v, x) et (v, y) ont été remplacées par les quatre arcs (x, v), (v, x), (y, v), (v, y). Il suffit de supprimer un arc entrant et un arc sortant, par exemple (y, v) et (v, x), de sorte que tout cycle Hamiltonien C dans G (qui contient nécessairement le 2-chemin (x, v, y)) est toujours simulé par un seul circuit, qui n'est autre que C orienté dans le sens (x, v, y), le sens opposé ayant été éliminé. Voir Fig. 5.2

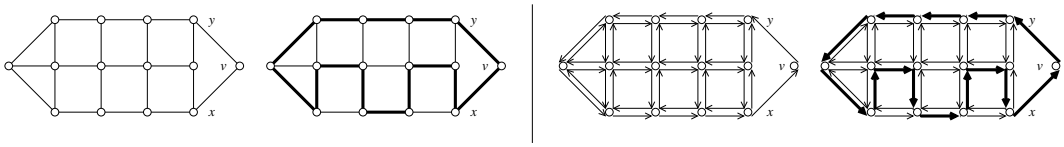


FIG. 5.2 – Schéma simple de réduction de UHAM-CYCLE à DHAM-CYCLE

Réduction no.2 : de UHAM-PATH(x, y) **à** DHAM-PATH(x, y) Le principe est le même que celui de la réduction précédente. On remplace toutes les arêtes par deux arcs anti-parallèles. Là encore, la réduction est linéaire et préserve la planarité, mais double le nombre de solutions puisqu'à chaque chemin non-orienté $(x, \cdots, y)$ correspondent les deux orientations $(x, \cdots, y)$ et $(y, \cdots, x)$. La suppression des doublons est cependant plus simple : il suffit des supprimer tous les arcs entrants en x et tous les arcs sortants en y. Ceci constitue également une réduction à DHAM-PATH.

Réduction no.3 : de UHAM-CYCLE **à** UHAM-PATH(x, y) Là encore, nous pouvons supposer que le graphe G de départ contient un sommet de degré 2 grâce au gadget en "roue de vélo" de la Fig. 4.27 qui préserve le nombre de cycles Hamiltoniens dans un graphe non-orienté. Soit v un tel sommet et u, w ses deux seuls voisins. La réduction consiste simplement à supprimer v et poser $x = u$ et $y = w$.

Réduction no.4 : de UHAM-PATH(x, y) **à** UHAM-PATH Trivialement, il suffit de rajouter deux sommets x' et y' au graphe G de départ, ainsi que les arêtes (x, x') et (y, y').

Réduction no.5 : de UHAM-PATH **à** UHAM-CYCLE Soit $G(V, E)$ le graphe non-orienté de départ. Le graphe d'arrivée G' est simplement obtenu en créant un nouveau sommet x, et une nouvelle arête (x, y) pour tout $y \in V$. Comme tout cycle Hamiltonien C dans G' passe par chaque sommet, il passe en particulier par x via un 2-chemin (u, x, v) avec $u, v \in V$, et les cycles Hamiltoniens C dans G' sont en correspondance bijective avec les chemins Hamiltoniens dans G. La réduction est donc linéaire et parcimonieuse, *mais on voit sur la Fig. 5.3 qu'elle ne préserve pas la planarité si tous les sommets de V ne partagent pas une face commune.*

Réduction no.6 : de DHAM-CYCLE **à** DHAM-PATH(x, y) L'idée de cette réduction est la même que celle de la réduction no.3 : il s'agit de casser les circuits Hamiltonien en un point de passage forcé. On utilise une version orientée du gadget en "roue de vélo" de la Fig. 4.27. Dans cette version, on remplace dans le graphe de départ G un sommet s quelconque de degré $k > 2$ par le gadget illustré sur la Fig. 5.4 : Les arcs pendants de ce gadget correspondent aux arcs incidents de s. Les autres arcs sont des paires d'arcs anti-parallèles simulant les arêtes de

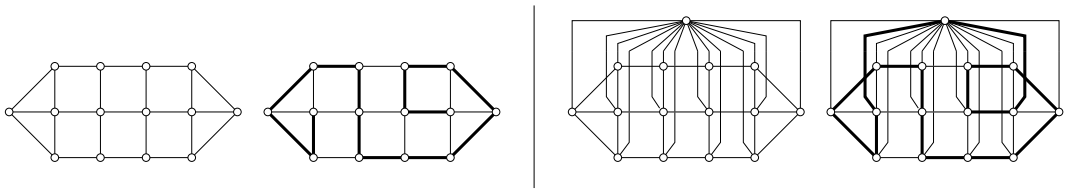


FIG. 5.3 – Schéma simple de réduction de UHAM-PATH à UHAM-CYCLE

la version non-orientée du gadget en "roue de vélo". Le nombre de circuits Hamiltoniens dans G n'est pas modifié par le remplacement de s par ce gadget. Ce gadget fait en outre apparaître un triplet (u, z, v) tel que z n'est traversé que par les deux 2-chemins orientés (u, z, v) et (v, z, u). Tout circuit Hamiltonien dans G doit donc contenir un de ces deux 2-chemins. Notre graphe d'arrivée G' est donc obtenu en supprimant le sommet z, et en posant $x = u$ et $y = v$. Les circuits Hamiltoniens dans G sont alors en bijection avec les chemins Hamiltoniens orientés dans G' allant soit de x à y, soit de y à x. La réduction est linéaire, parcimonieuse et préserve la planarité. Notons que sa correction utilise le fait que x et y peuvent être indifféremment point de départ ou point d'arrivée.

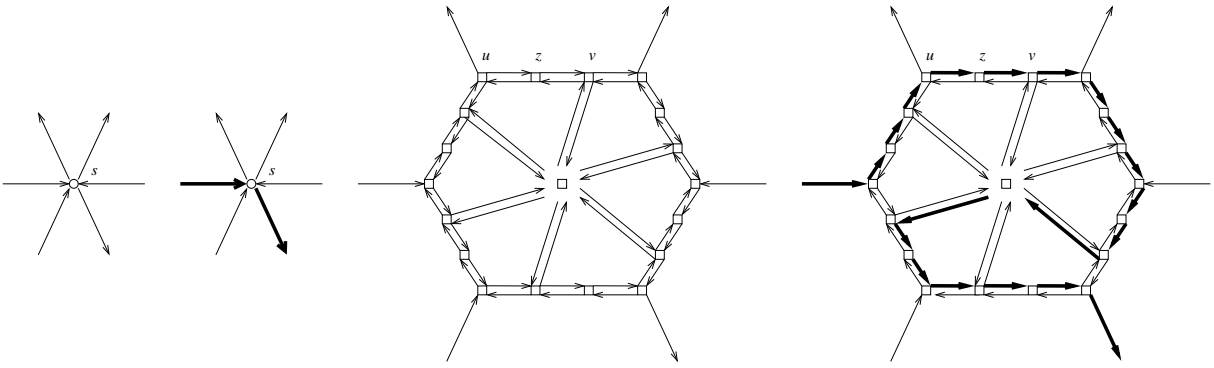


FIG. 5.4 – Roue de vélo, version orientée

Réduction no.7 : de DHAM-PATH(x, y) **à** DHAM-PATH Trivialement, à l'instar de la réduction no.4, il suffit de rajouter deux sommets x' et y' au graphe G de départ, ainsi que les paires d'arcs anti-parallèles (x, x'), (x', x) et (y, y'), (y', y).

Réduction no.8 : de DHAM-PATH **à** DHAM-CYCLE Cette réduction est essentiellement la même que la no.5, sa version non-orientée. Soit $G(V, A)$ le graphe de départ. Pour obtenir le graphe d'arrivée G', on créé un nouveau sommet x, et on créé une paires d'arcs anti-parallèles (x, y) et (y, x) pour tout sommet y de V. Comme la réduction no.5, cette réduction est linéaire et parcimonieuse mais ne préserve pas la planarité.

Réduction no.9 : de DHAM-CYCLE **à** UHAM-CYCLE Soit $G(V, A)$ le graphe orienté de départ. On construit le graphe d'arrivée $G'(V', E')$ de la façon suivante : Pour tout sommet $v \in V$, on créé le 2-chemin (i_v, c_v, o_v) dans G', où i_v, c_v et o_v sont de nouveaux sommets, et

pour tout arc $(x, y) \in E$, on ajoute dans E' l'arête (o_x, i_y). La réduction est clairement linéaire et parcimonieuse, mais comme illustré sur la Fig. 5.5, elle ne préserve pas la planarité.

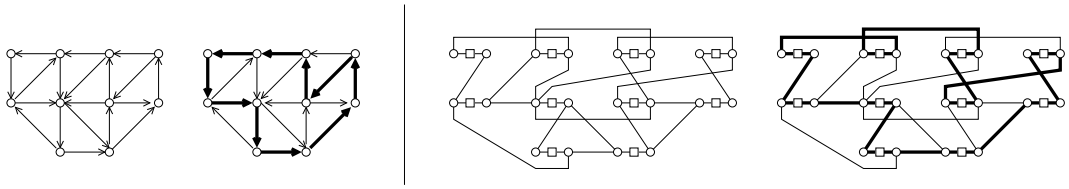


FIG. 5.5 – Schéma simple de réduction de DHAM-CYCLE à UHAM-CYCLE

Réduction no.10 : de DHAM-PATH$(\overrightarrow{x, y})$ **à** DHAM-PATH(x', y') Trivialement, il suffit de créer deux nouveaux sommets x' et y' ainsi que les arcs (x', x) et (y', y).

Réduction no.11 : de UHAM-PATH(x, y) **à** DHAM-PATH$(\overrightarrow{x', y'})$ Il suffit de remplacer chaque arête du graphe de départ par une paire d'arcs anti-parallèles et de créer les arcs (x', x) et (y, y') où x' et y' sont de nouveaux sommets.

5.3 L'équivalence des variantes de PLAN-HAMILTON

On le voit, les deux points de blocage pour la préservation de la planarité sont :
- la simulation de chemins par des cycles (réductions no.5 et no.8),
- la simulation de cycles orientés par des cycles non-orientés. (réduction no.9).

La suite de ce chapitre présente comment remplacer ce schéma par celui de la Fig. 5.6. Dans ce nouveau schéma, les réductions no.5, no.8 et no.9 disparaissent au profit de quatre autres réductions préservant la planarité :
- De UHAM-PATH et UHAM-PATH(x, y) à UHAM-CYCLE, deux réductions essentiellement identiques.
- De DHAM-PATH et DHAM-PATH(x, y) à UHAM-CYCLE, également essentiellement identiques.

Le principe commun de ces réductions est de remplacer chaque sommet v de degré k d'un graphe G par un gadget $g(v)$ à k *portes*, et simuler chaque arête (u, v) par la fusion des portes correspondantes dans $g(u)$ et $g(v)$. Une *porte* est simplement une paire d'arêtes pendantes. Il s'agit de simuler un chemin Hamiltonien H dans G par un cycle Hamiltonien empruntant les portes correspondant aux arêtes empruntées par H, le cycle se refermant au niveau des gadgets

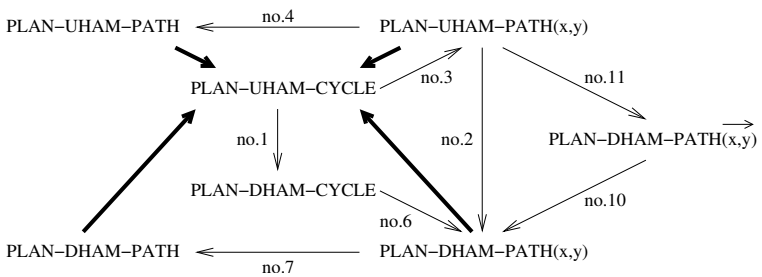


FIG. 5.6 – Schéma d'inter-réductions entre variantes de PLAN-HAMILTON

$g(x)$ et $g(y)$ correspondant aux extrémités x et y de H, comme montré sur la Fig. 5.7. Nous appellerons ces gadget $g(u)$ des *chandeliers*.

5.3.1 Variantes non-orientées : le gadget petit chandelier

Lorsque l'on réduit UHam-Path (ou UHam-Path(x, y)) à UHam-Cycle, on appelle tout chandelier $g(u)$ un *petit chandelier*[5]. Les configurations d'un petit chandelier à k portes se divisent en deux motifs :

- motif 1 : la configuration utilise exactement une porte parmi k. A l'intérieur du gadget, un seul chemin joint les deux arêtes de la porte utilisée.
- motif 2 : la configuration utilise exactement deux portes parmi k, et rien d'autre. A l'intérieur du gadget, il y a deux chemins, joignant chacun l'arête à main gauche d'une porte à l'arête à main droite de l'autre porte.

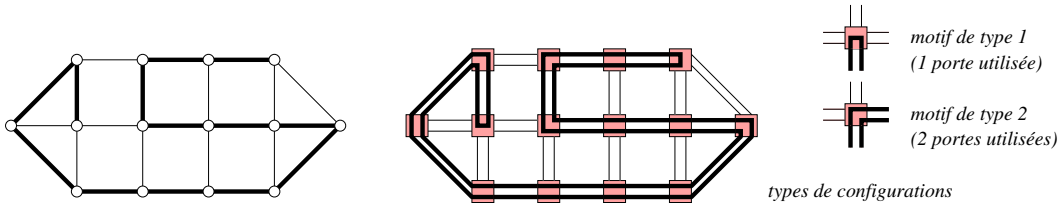


Fig. 5.7 – Notre schéma de réduction de Plan-UHam-Path à Plan-UHam-Cycle

Selon le problème que l'on veut réduire, Un petit chandelier $g(u)$ à k portes fonctionne selon un mode parmi 3, et doit accepter :

- mode 1 : tous les motifs 1 et rien d'autre. C'est le cas pour les gadgets $g(x)$ et $g(y)$ lorsque l'on réduit UHam-Path(x, y).
- mode 2 : tous les motifs 2 et rien d'autre. C'est le cas pour tout gadget $g(u)$, $x \neq u \neq y$, lorsque l'on réduit UHam-Path(x, y).
- mode 3 : tous les motifs 1 et 2 et rien d'autre. C'est le cas pour tout gadget $g(u)$, lorsque l'on réduit UHam-Path.

Notons que tout cycle Hamiltonien H dans un graphe construit à partir de petits chandeliers utilise au moins deux chandeliers selon le motif 1, car sinon le cycle ne peut se refermer. De plus, H ne peut utiliser plus de deux gadgets selon ce même motif, car sinon H ne serait pas un et un seul cycle. Donc H utilise toujours exactement deux chandeliers selon le motif 1. Il découle alors naturellement que :

- si tous les petits chandeliers $g(u)$ fonctionnent en mode 3, la construction est une réduction de UHam-Path à UHam-Cycle, et
- si tous les petits chandeliers $g(u)$, $x \neq u \neq y$, fonctionnent en mode 2, et que $g(x)$ et $g(y)$ fonctionnent en mode 1, la construction est une réduction de UHam-Path à UHam-Cycle.

5.3.2 Variantes orientées : le gadget grand chandelier

Lorsqu'on réduit DHam-Path (ou DHam-Path(x, y)) à UHam-Cycle, on appelle tout chandelier $g(u)$ un *grand chandelier*. Une porte d'un tel gadget $g(u)$ est une *porte d'entrée* ou une *porte de sortie* selon qu'elle simule un arc entrant ou sortant en u. Ses configurations se divisent également en deux motifs analogues à ceux du petit chandelier :

[5] Les chandelles arriveront plus tard...

– motif 1 : la configuration utilise exactement une porte, indifféremment d'entrée ou de sortie.
– motif 2 : la configuration utilise exactement deux portes, une d'entrée et une de sortie.

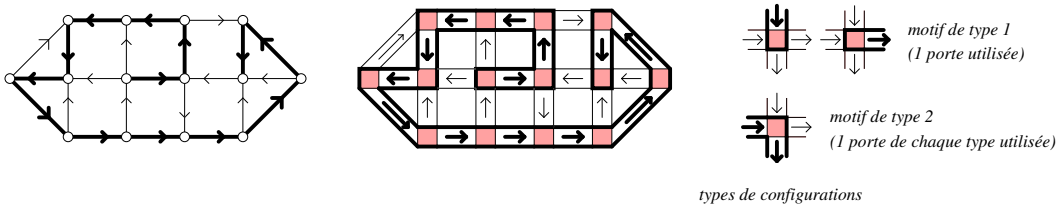


FIG. 5.8 – Notre schéma de réduction de PLAN-DHAM-PATH à PLAN-UHAM-CYCLE

Là encore, selon le problème que l'on veut réduire, un grand chandelier $g(u)$ à k portes doit accepter, selon les modes d'utilisation :

– mode 1 : tous les motifs 1, et rien d'autre. C'est le cas pour les gadgets $g(x)$ et $g(y)$ lorsque l'on réduit DHAM-PATH(x, y).
– mode 2 : tous les motifs 2, et rien d'autre. C'est le cas pour tout gadget $g(u)$, $x \neq u \neq y$, lorsque l'on réduit DHAM-PATH(x, y).
– mode 3 : tous les motifs 1 et 2. C'est le cas pour tout gadget $g(u)$, lorsque l'on réduit DHAM-PATH.

Selon le même raisonnement que ci-dessus, tout cycle Hamiltonien H dans un graphe construit à partir de grands chandeliers doit utiliser exactement deux chandeliers selon le motif 1. De plus, l'un des deux utilise une porte d'entrée et l'autre utilise une porte de sortie, car sinon :

– si les deux chandeliers de motif 1 utilisaient une porte de sortie alors il existerait un chandelier utilisant deux portes d'entrée, un motif interdit, et
– symétriquement, si les deux chandeliers de motif 1 utilisaient une porte d'entrée alors il existerait un chandelier utilisant deux portes de sortie, un motif également interdit.

Il découle alors naturellement que :

– si tous les grands chandeliers $g(u)$ fonctionnent en mode 3, la construction est une réduction de DHAM-PATH à UHAM-CYCLE, et
– si tous les grands chandeliers $g(u)$, $x \neq u \neq y$, fonctionnent en mode 2, et que $g(x)$ et $g(y)$ fonctionnent en mode 1, la construction est une réduction de DHAM-PATH à UHAM-CYCLE.

Ceci établit la correction de nos réductions de substitution dans le schéma 5.6. Il reste à implémenter linéairement et parcimonieusement les gadgets planaires *petit chandelier* et *grand chandelier*. Pour ce faire, il faut savoir faire des additions en binaire avec PLAN-HAMILTON, afin de compter les portes utilisées.

5.3.3 Gadgets fonctionnels pour l'arithmétique

Nous voulons obtenir un gadget planaire fonctionnel (au sens de la définition 4.10 du chapitre précédent) nommé ADD, calculant la somme de deux nombres de n bits sous la forme d'un nombre de $n+1$ bits. Nous utilisons ici un simple schéma de propagation de retenue. Rappelons que la somme de deux nombres a et b d'au plus n bits tels que $bin(a) = (a_{n-1}, \cdots, a_0)$ et $bin(b) = (b_{n-1}, \cdots, b_0)$ est un nombre c d'au plus $n+1$ bits tel que $bin(s) = (s_n, \cdots, s_0)$, $bin(a)$, $bin(b)$ ainsi qu'une liste de retenues $(c_n, \cdots, c_0)$ vérifient les conditions :

– $c_0 = 0$,
– pour tout $0 \leq i < n$, $c_{i+1} = \text{MAJ}(a_i, b_i, c_i)$, défini par $c_{i+1} = (a_i \wedge b_i) \vee (b_i \wedge c_i) \vee (c_i \wedge a_i)$,
– pour tout $0 \leq i \leq n$, $s_i = \text{ODD}(a_i, b_i, c_i)$, défini par $s_i = a_i \oplus b_i \oplus c_i$.

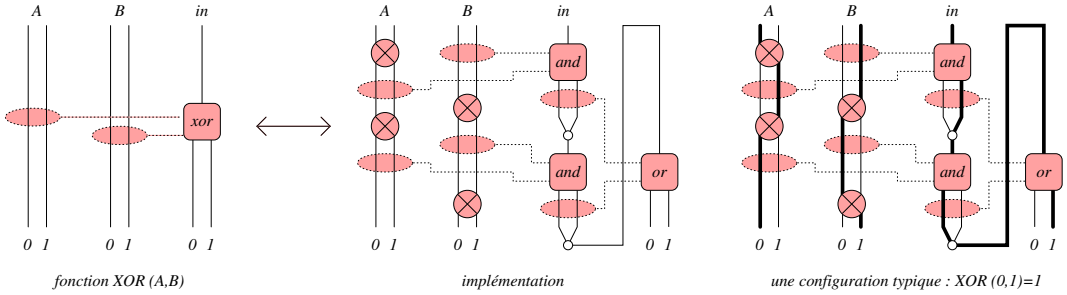


FIG. 5.9 – Le gadget pour la fonction XOR

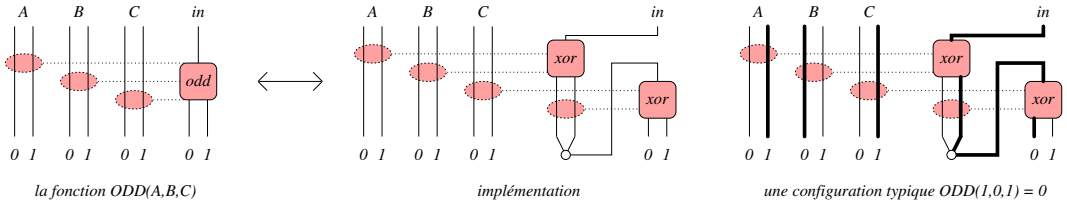


FIG. 5.10 – Le gadget pour la fonction ODD

Nous avons donc besoin des gadgets fonctionnels associés aux fonctions XOR, MAJ et ODD pour implémenter ADD. Nous les implémentons naturellement avec les gadgets fonctionnels NOT, AND et OR déjà obtenus dans le chapitre précédent, à la manière d'un circuit arithmétique. La fonction $\mathrm{XOR}(a,b) = a \oplus b$ est implémentée par son développement en DNF $(a \wedge \neg b) \vee (\neg a \wedge b)$, et les fonction MAJ et ODD sont implémentées par leur expressions données plus haut. Les implémentations de XOR, MAJ et ODD sont représentées sur les Figs. 5.9, 5.10, 5.11. Leurs conventions restent celles des gadgets fonctionnels données au chapitre précédent.

Enfin, l'implémentation du gadget ADD est présentée sur la Fig. 5.12. Nous ne prouvons pas la correction et la parcimonie de ces gadgets planaires, qui vont toutes de soi. Notons que la planarité des gadgets est à nouveau obtenue par l'utilisation d'un certain nombre de crossovers flip-flap-flops définis au chapitre précédent. Dans un gadget ADD sur des opérandes de n bits, le nombre de crossovers est $\Theta(n^2)$, les $\Theta(n)$ fils de l'opérande de droite étant traversés par $\Theta(n)$ flip-flap-flops établissant la connexion de l'opérande de gauche au circuit arithmétique. Le gadget planaire ADD est donc de taille quadratique en n, mais comme toutes les additions que nous ferons dans les chandeliers s'effectueront sur un nombre $O(1)$ de bits, tout gadget planaire ADD utilisé dans un chandelier sera de taille $O(1)$.

5.3.4 Implémentation du petit chandelier (cadre non-orienté)

Le petit chandelier est implémenté sur la Fig. 5.13. Les deux sommets adjecents aux deux arêtes pendantes de chaque porte (en haut) sont connectés par une arête, que l'on appelle le *verrou* de la porte. Les verrous sont reliés entre eux en un cycle alternant verrous et 2-chemins contenant des sommets de degré 2, de sorte que pour toute porte, toute configuration Hamiltonienne emprunte soit son verrou, soit ses deux arêtes pendantes simultanément, mais jamais son verrou et ses arêtes pendantes en même temps. Un verrou est choisi arbitrairement comme premier verrou pour l'ordre suivant le sens anti-trigonométrique.

On souhaite qu'il y ait une porte ou deux portes utilisées, i.e., soit 1 soit 2 verrous maigres. Pour ce faire, le chandelier associe à chaque verrou v_i deux *petites chandelles*, une *chandelle*

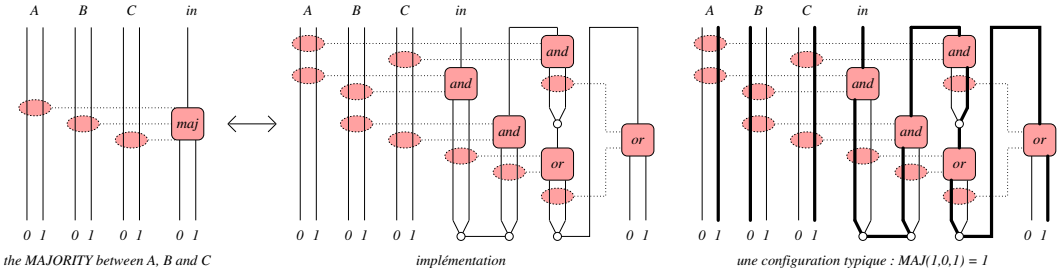


Fig. 5.11 – Le gadget pour la fonction MAJ

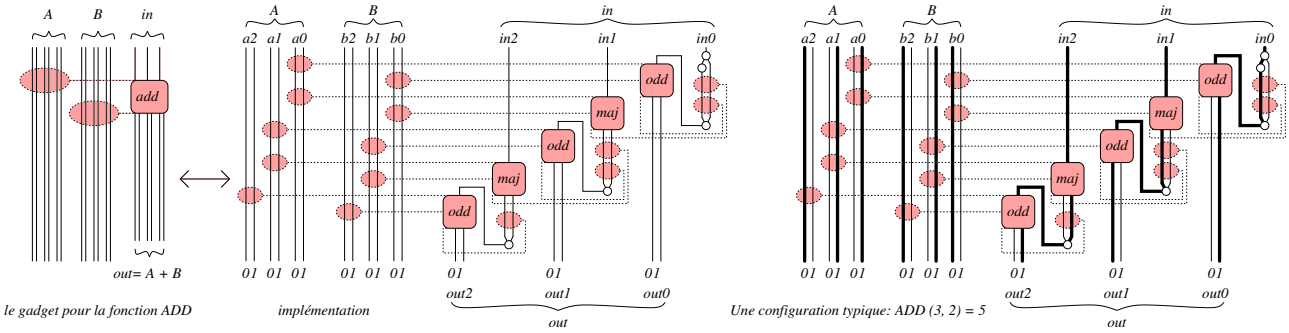


Fig. 5.12 – Le gadget pour la fonction ADD

supérieure, et (sauf pour le premier verrou) une *chandelle inférieure*. Une petite chandelle est un ensemble de 3 paires d'arêtes parallèles, représentant un nombre de 3 bits :

- Pour une chandelle supérieure, le bit de poids faible (le plus à droite) témoigne de la maigreur du verrou associé. Ceci est assuré par une échelle exclusive reliant le verrou à l'arête droite (celle codant le 1) de ce bit. Les deux autres bits sont forcés à 0 par un sommet de degré 2 posé sur les arêtes gauches (celles codant le 0) de ces bits. Par conséquent, le nombre porté par une chandelle supérieure est 1 si son verrou est maigre et 0 sinon.
- Pour une chandelle inférieure, les trois bits comptent le nombre de verrous maigres parmi ceux situés à gauche de son verrou (inclus). Il s'agit donc de l'accumulation additive de la chandelle supérieure associée et de la chandelle inférieure précédente, implémenté par un gadget ADD. Le troisième bit n'est qu'un bit de contrôle d'overflow et est toujours forcé à 0 par un sommet de degré 2 posé sur son arête de gauche. Ainsi les sommes accumulées ne dépassent jamais le nombre 3.

La dernière chandelle inférieure est connectée à une boîte noire X qui détermine le mode de fonctionnement du petit chandelier. Les trois implémentations possibles de cette boîte noire sont représentées sur la Fig. 5.14 :

- mode 1 : on souhaite exactement un verrou maigre donc le bit de poids faible est forcé à 1. Les autres ont forcés à 0.
- mode 2 : on souhaite exactement deux verrous maigres donc le bit de poids moyen est forcé à 1. Les autres sont forcés à 0.
- mode 3 : on souhaite un ou deux verrous maigres, donc on force l'exclusion des deux bits de poids faible (car $1 = bin(0,1)$ et $2 = bin(1,0)$), et on force le bit de poids fort à 0.

Le gadget est complété par un chemin enfilant en zig-zag les bits des chandelles et les fils d'alimentation des gadgets ADD afin que le tout s'intègre au collier extérieur.

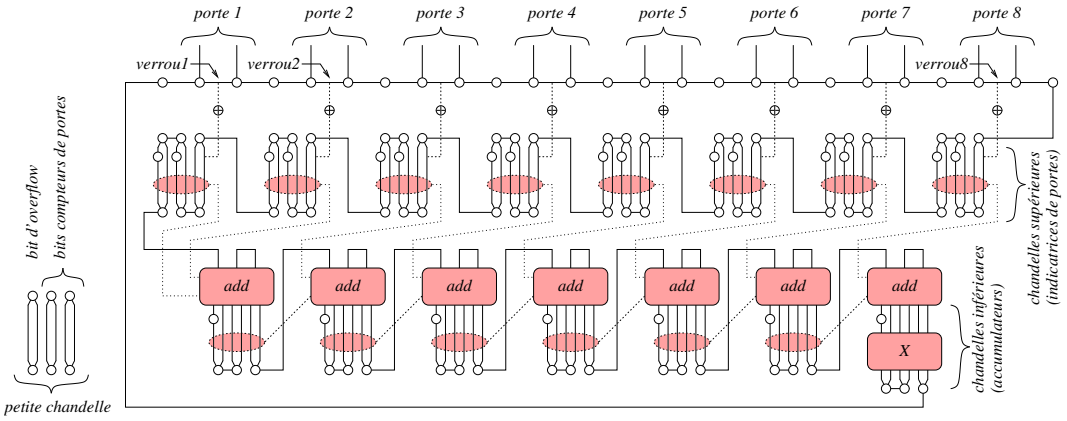


FIG. 5.13 – Le petit chandelier simulant un sommet de degré 8 dans un graphe non-orienté

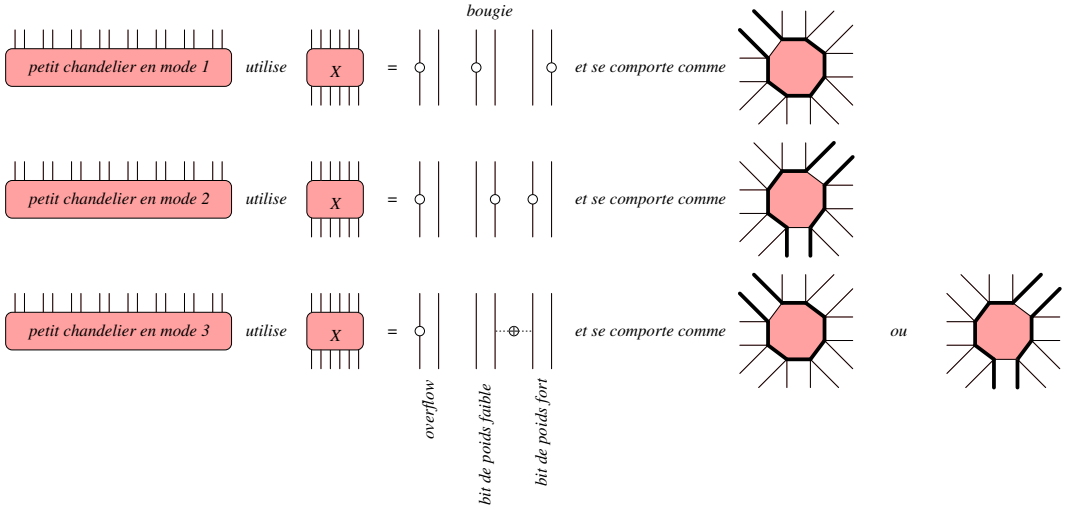


FIG. 5.14 – Les modes de fonctionnement du petit chandelier

5.3.5 Implémentation du grand chandelier (cadre orienté)

Le grand chandelier suit le même principe que le petit chandelier. Seules changent les chandelles et la boîte noire X. Voir Fig 5.16. Les grandes chandelles codent un nombre de 4 bits :

- Pour une grande chandelle supérieure, le bit 0 (resp. le bit 2) témoigne de la maigreur du verrou associé si celui-ci est connecté à une porte d'entrée (resp. à une porte de sortie). Les bits d'overflow 1 et 3 sont forcés à 0.
- Pour une grande chandelle inférieure, les deux bits de poids faible (resp. fort) accumulent le nombre de verrous maigres connectés aux portes d'entrée (resp. de sortie). Les bits 1 et 3 sont des bits de contrôle d'overflow, toujours forcés à 0. Il ne peut donc jamais y avoir plus d'un verrou maigre de chaque type (entrée ou sortie).

Remarque 5.3 *Du fait de la présence des bits d'overflow toujours forcés à 0, un seul gadget ADD suffit pour chaque chandelle inférieure, puisqu'il ne peut y avoir aucune retenue propagée entre les deux bits de poids faible (comptant les entrées utilisées) et les deux bits de poids fort (comptant les sorties utilisées).*

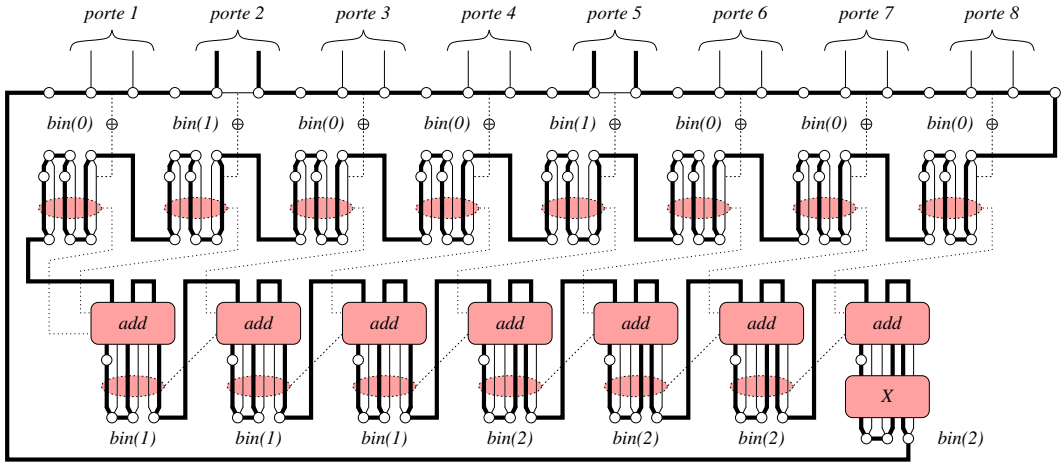


Fig. 5.15 – Un état local typique pour le petit chandelier en mode 2 ou 3

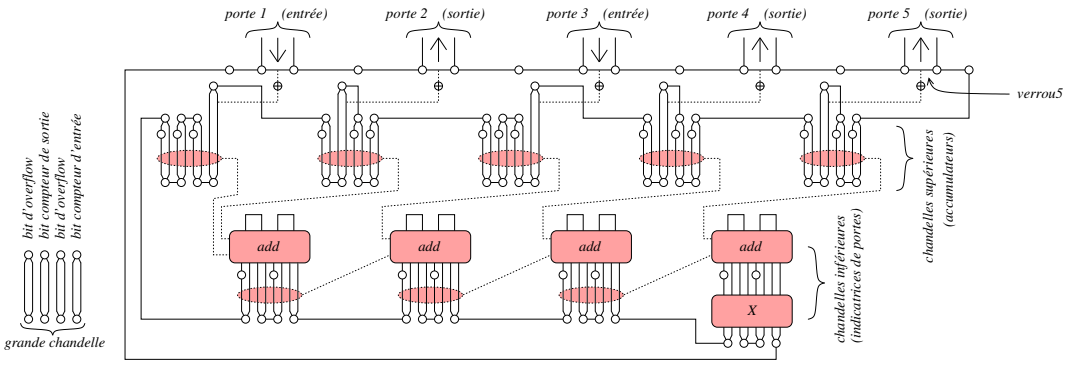


Fig. 5.16 – Le grand chandelier simulant un sommet de degré 5 dans un graphe orienté

Là encore, la boîte noire terminale X détermine le mode de fonctionnement du grand chandelier :

- mode 1 : on veut exactement un verrou maigre, indifféremment de son type (entrée ou sortie). On connecte donc les bits 0 et 2 par une échelle exclusive. Les bits d'overflow sont forcés à 0.
- mode 2 : on veut exactement deux verrous maigres, un de chaque type (entrée et sortie). Les bits 0 et 2 sont donc forcés à 1, et les bits d'overflow sont forcés à 0.
- mode 3 : on veut la réunion des deux modes précédents. Il y a juste à forcer les bits d'overflow à 0. Le gadget devant nécessairement être visité, les deux autres bits ne peuvent en effet être tous deux nuls.

Ceci termine la présentation des réductions linéaires et parcimonieuses de Plan-Uham-Path (Plan-Uham-Path(x, y)) et Plan-Dham-Path (Plan-Dham-Path(x, y)) à Plan-Uham-Cycle. Remarquons que si l'on développe les gadgets de ces réductions, tous se ramènent finalement aux gadgets AND, NOT, flip-flap-flop, et échelle exclusive. Tous ces gadgets sont au plus de degré 4, et par conséquent nos réductions sont également des réductions à Plan-Uham-Cycle sur les graphes de degré maximal 4. Dans la section suivante nous faisons chuter ce degré à 3, évidemment optimal.

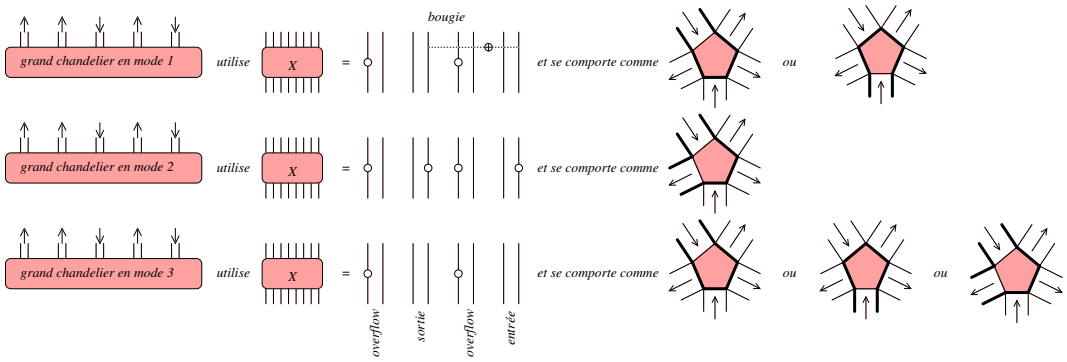


FIG. 5.17 – Les modes de fonctionnement du grand chandelier

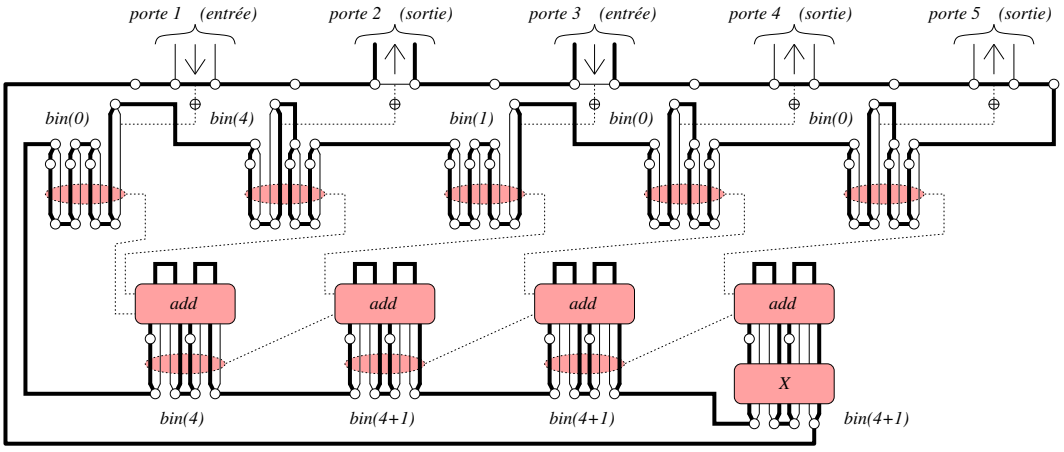


FIG. 5.18 – Un état local typique pour le grand chandelier en mode 2 ou 3

5.4 Variantes non-orientées de PLAN-HAMILTON à degré borné à 3

En fait, parmi les gadgets cités ci-dessus, seul le gadget AND présente effectivement deux sommets de degré 4. Donc si nous trouvons un substitut au gadget AND qui soit de degré au plus 3, notre réduction devient optimale pour le degré du graphe de sortie.

L'idée est de garder le gadget AND tel qu'il est, mais en simulant juste les deux sommets de degré 4 (le sommet central et le sommet Est) par deux gadgets de degré inférieur, nommés resp. CENTRAL et EST, comme montrés sur la Fig. 5.19. Comme on peut le voir sur cette figure, le gadget EST est en fait le gadget CENTRAL auquel deux sommets de degré 2 ont été rajoutés.

Nous laissons au lecteur le soin de vérifier que le gadget CENTRAL admet les huit états locaux montrés sur la Fig. 5.21. Il n'est pas parcimonieux puisqu'il n'y a que six configurations : les deux configurations décrivant un chemin en ligne droite Nord-Sud et Est-Ouest ayant chacune deux états locaux. Cependant le sommet central du gadget AND *n'étant jamais traversé en ligne droite* (voir Fig. 5.20), ces configurations n'apparaîtront jamais et *ne seront jamais source de non-parcimonie.*

Ce n'est pas le cas du sommet Est qui, lui, peut être traversé par une ligne droite Nord-Sud. Le gadget central est donc impropre pour le simuler parcimonieusement. En revanche, le sommet Est n'est jamais traversé par un coude Nord-Est. Cette remarque justifie l'implémentation du gadget EST par une modification du gadget CENTRAL consistant en la pose de deux sommets de degré 2 comme montré sur la Fig. 5.19. On constate facilement sur la Fig. 5.21 que la pose

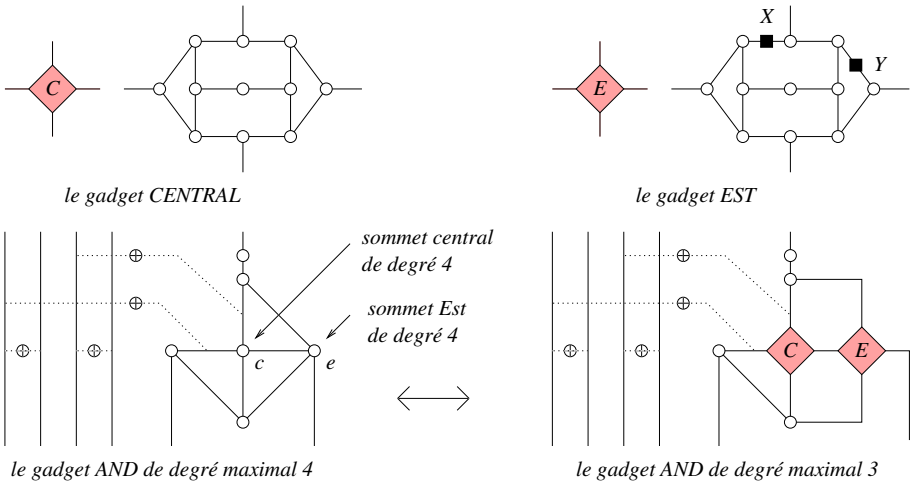


FIG. 5.19 – Comment réduire le degré maximal du gadget AND à 3

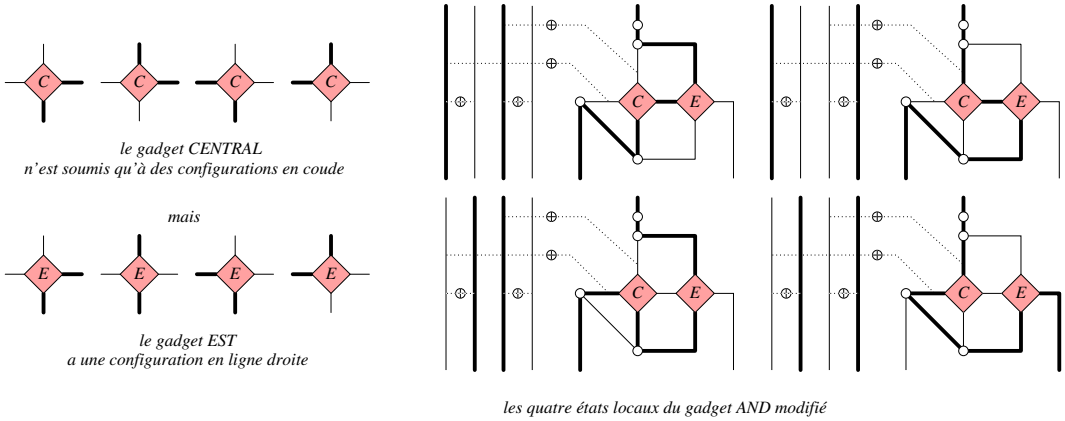


FIG. 5.20 – Gadgets EST et CENTRAL en action

de ces sommets élimine les états locaux doublons pour les configurations rectilignes Nord-Sud et Est-Ouest ainsi que la configuration en coude Nord-Est. Le gadget EST est donc parcimonieux et convient à la simulation du sommet Est du gadget AND.

Ceci conclut la preuve que toute variante de PLAN-HAMILTON se réduit linéairement et parcimonieusement à PLAN-UHAM-CYCLE sur les graphes de degré maximal 3.

Remarque 5.4 *Notons que si l'on veut que la réduction soit parcimonieuse, on ne peut éliminer les sommets de degré 2 et obtenir un graphe complètement cubique. En effet, le nombre de cycles Hamiltoniens dans un graphe cubique passant par une arête fixée est toujours pair [68] : il n'existe donc pas de graphes cubiques avec un unique chemin Hamiltonien, par exemple.*

Enfin, les variantes de PLAN-HAMILTON sont également toutes réductibles à PLAN-UHAM-PATH et PLAN-UHAM-PATH(x, y) de degré $<= 3$: En effet, la réduction à PLAN-UHAM-CYCLE introduisant de nombreux sommets de degré 2 dans le graphe de sortie G, il suffit de choisir un sommet v de degré 2 et de "casser" les cycles Hamiltoniens en v : Soient u et w ses sommet adjacents. On supprime v et on créé deux nouveaux sommets x et y et deux nouvelles arêtes (x, u) et (y, w). Les cycles Hamiltoniens dans G avant cette transformation sont alors en bijection

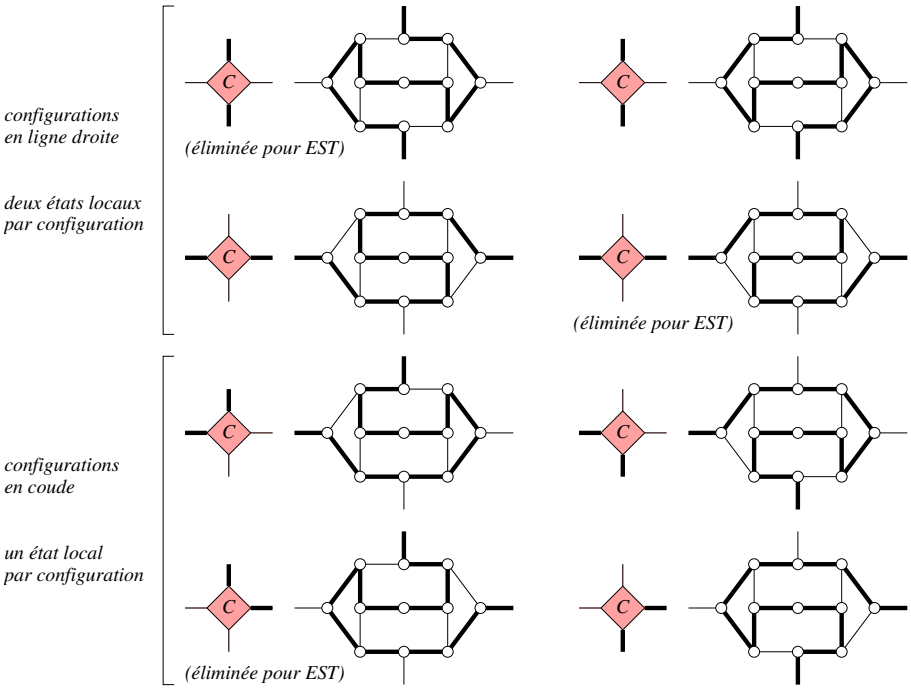


FIG. 5.21 – Configurations des gadgets CENTRAL et EST

avec les chemins (aboutissant nécessairement en x et y) dans G après la transformation. Ce qui conclut la preuve de la Proposition 5.2.

6

Logique et NP-complétude minimale

une caractérisation logique de la classe de SAT

Sommaire

6.1 Introduction 108
6.2 Problèmes locaux et classe LIN-LOCAL 109
6.3 Un problème NP-complet sur des structures bijectives 113
6.4 Un seul prédicat unaire dans la signature et au second ordre 115
6.5 Elimination totale des prédicats unaires dans la signature 117
6.6 L'unique problème minimalement NP-complet 120
6.7 Equivalence parcimonieuse à SAT et NP-complétude minimale . . . 125

6.1 Introduction

Ce chapitre présente des résultats de recherches menées en collaboration avec E. Grandjean, et qui répondent à deux préoccupations liées à la logique, et plus précisément, à la complexité descriptive[6] :

1. La recherche d'une caractérisation intrinsèque (logique) de la classe des problèmes linéairement réductibles à SAT (resp. PLAN-SAT), classe à laquelle cette thèse contribue largement à la suite des travaux de A. Dewdney [25], N. Creignou [20] et E. Grandjean [45].
2. La recherche de(s) problèmes NP-complets(s) défini(s) par une (des) formule(s) logique(s) minimale(s).

Le deuxième point mérite d'être explicité. Le théorème bien connu de Fagin [28] établit que la logique existentielle du second ordre (ESO) "capture" exactement la classe NP, autrement dit, un problème est NP ssi il est défini par une formule ESO. La question naturelle que nous nous sommes posée à la suite de la caractérisation de Fagin est la suivante : Quelle(s) est (sont) la (les) plus simple(s) formule(s) ESO qui défini(ssen)t un (des) problème(s) NP-complet(s) ? Cette question est quelque peu reliée à deux articles récents [27, 35] qui classent complètement les classes préfixielles de la logique ESO portant respectivement sur les mots et les graphes (et plus généralement sur les *structures relationnelles*) selon leur capacité à exprimer des problèmes NP-complets ou à n'exprimer que des problèmes polynomiaux. Par exemple, le problème 3-COL s'exprime en logique existentielle monadique du second ordre (EMSO) avec 2 variables du premier ordre par la formule de signature $\{E\}$ (où E est la relation binaire d'arête) :

$$\exists H\ \exists L\ \forall x\ \forall y \quad \bigwedge \begin{cases} Hx \vee Lx \\ Exy \Longrightarrow ((Hx \oplus Hy) \vee (Lx \oplus Ly)) \end{cases}$$

La formule ci-dessus dit essentiellement qu'une couleur (choisie parmi trois) se code sur un numéro de deux bits (un bit de poids fort H et un bit de poids faible L), que le numéro zéro ne code aucune couleur (et donc au moins un des bits doit être allumé), et que deux sommets adjacents doivent avoir deux couleurs différentes, i.e, leurs numéros doivent différer sur au moins un bit.

A l'inverse, il est facile de vérifier que, sur des *structures relationnelles* (comme les graphes représentés comme ci-dessus par des structures de signature $\{E\}$, où E est un prédicat binaire), les formules qui n'utilisent que des *symboles relationnels* et seulement *une variable du premier ordre*, c'est-à-dire de la forme $\exists \overline{R}\ \forall x\ \psi$ (où ψ est sans quantificateur), ne définissent que des problèmes dégénérés.

La situation change complètement si l'on autorise des symboles de *fonctions* soit dans la signature σ des σ-structures données en entrée, soit dans les symboles EMSO. Par exemple, [47] montre qu'on peut exprimer des problèmes NP-complets, tel le problème du cycle Hamiltonien, avec des formules ESO avec 1 variable du premier ordre x, de la forme

$$(*) \quad \exists \overline{f}\ \forall x \quad \psi(x, \overline{f}, E),$$

où ψ est sans quantificateur, $\overline{f}$ est une liste de symboles de fonctions unaires et E est un prédicat binaire. Plus précisément, [47] montre qu'un problème est défini par une formule de la forme $(*)$ ssi ce problème est reconnu en temps non-déterministe $O(n)$ où n est le nombre de sommets du graphe concerné.

Etudions maintenant les problèmes exprimés par des formules EMSO avec une seule variable du premier ordre sur des *stuctures fonctionnelles unaires.*

[6] Nous ne redéfinissons pas ici les notions logiques utilisées, toutes classiques, telles les formules du premier ordre, du second ordre, etc., qui sont définies dans les livres de logique et de complexité descriptive tels [26, 56].

6.2 Problèmes locaux et classe LIN-LOCAL

Définition 6.1 (structure unaire) *On appelle* structure unaire *toute σ-structure finie $\langle D, \overline{f}, \overline{L}\rangle$ de signature unaire $\sigma = \{\overline{f}, \overline{L}\}$, c'est-à-dire formée d'une liste $\overline{f}$ de symboles de fonctions unaires, encore appelées* fonctions de voisinage *et d'une liste $\overline{L}$ de prédicats monadiques, encore appelés* labels.

Définition 6.2 (formule locale) *On appelle* formule locale *une formule EMSO à une seule variable du premier ordre, de la forme*

$$(**) \quad \exists \overline{U}\ \forall x \quad \psi(\overline{U}, \overline{f}, \overline{L}, x),$$

où ψ est sans quantificateur, et de signature unaire $\sigma = \{\overline{f}, \overline{L}\}$. Par analogie avec un problème de coloriage de graphe, on appelle aussi couleurs *les prédicats unaires $\overline{U}$.*

Définition 6.3 (problèmes locaux) *On appelle* LOCAL, *la classe des problèmes définis par une formule locale sur des structures unaires, encore appelés* problèmes locaux.

Remarque 6.4 *On voit facilement que l'on ne change pas la notion de problème local en autorisant dans la formule $(**)$, à la place de $\overline{U}$, une liste de prédicats $\overline{R}$ d'arités quelconques. L'essentiel est de n'autoriser* qu'une seule *variable du premier ordre.*

La classe LOCAL est comparable, sans être identique, à la classe notée Monadic-NLIN, définie et étudiée dans [61]. Nous estimons qu'aucune de ces deux classes ne peut être considérée comme une classe de complexité car aucune ne possède de caractérisation intrinsèque en termes de machines. Plus précisément, on vérifie facilement que tout problème local A est clos pour la réunion disjointe de ses structures, autrement dit, dès que deux structures $\mathcal{S}$ et $\mathcal{S}'$ sont dans A, leur réunion disjointe $\mathcal{S} + \mathcal{S}'$ appartient également à A : ce fait interdit que LOCAL soit une classe de complexité. C'est la raison pour laquelle nous introduisons la classe de complexité suivante :

Définition 6.5 *LIN-LOCAL désigne la classe des problèmes linéairement réductibles à des problèmes locaux (via des réductions DLIN).*

La définition par clôture de LIN-LOCAL est comparable à celle de la classe LOGCFL [2, 88] des problèmes réductibles aux langages algébriques (CFL= "Context Free Language") par réductions LOGSPACE.

On montre que le problème SAT est LIN-LOCAL-complet, autrement dit :

Proposition 6.6 *Pour tout problème A, on a l'équivalence entre les deux points suivants :*

1. *A se réduit linéairement à* SAT.
2. *$A \in$ LIN-LOCAL.*

Preuve

- $(2 \Longrightarrow 1)$: Il suffit de "dérouler" sur le domaine D de x, donc en temps linéaire, la partie du premier ordre $\forall x\ \psi(x)$ de la formule locale $(**)$ qui devient $\bigwedge_{a \in D} \psi(a)$ et se traduit facilement en une donnée de SAT.
- $(1 \Longrightarrow 2)$: voir ci-dessous. Il s'agit essentiellement de coder le problème SAT en un problème local. Nous le ferons de quatre façons différentes. Dans les quatre cas, les fonctions unaires de la structure obtenue pour le problème local seront des bijections.

■

Dans les chapitres précédents de cette thèse, nous nous sommes aussi intéressé à PLAN-SAT et aux problèmes linéairement réductibles à PLAN-SAT. Come nous le verrons, nos notions et résultats "passent bien" dans le cas planaire.

Définition 6.7 *On appelle PLAN-LOCAL la classe des problèmes définis par une formule locale sur des structures unaires planaires $\mathcal{S} = \langle D, \overline{f}, \overline{L}\rangle$, c'est à dire dont le graphe associé $G(\varphi) = (V, E)$ où $V = D$ et $E = \{(x, y) : \exists f \in \overline{f}, y = f(x)\}$ est planaire.*

Définition 6.8 *LIN-PLAN-LOCAL désigne la classe des problèmes linéairement réductibles à un problème PLAN-LOCAL.*

Là encore, le problème PLAN-SAT est LIN-PLAN-LOCAL-complet, autrement dit :

Proposition 6.9 *Pour tout problème A, les deux points suivants sont équivalents :*

1. *A se réduit linéairement à* PLAN-SAT *;*
2. *$A \in$ LIN-PLAN-LOCAL.*

La preuve de $(1 \Longrightarrow 2)$ est identique à celle de la proposition analogue pour SAT et LIN-LOCAL, nos quatre réductions ci-dessous préservant la planarité. En revanche, pour la preuve de $(2 \Longrightarrow 1)$, on ne peut pas simplement dérouler la formule car alors, la planarité n'est pas préservée. Un premier obstacle est la possibilité qu'il existe des compositions fonctionnelles dans la formule φ caractérisant un problème PLAN-LOCAL auquel se réduit A. Le lemme suivant affirme qu'on peut toujours éliminer les compositions fonctionnelles, au prix d'une augmentation du nombre de prédicats unaires au second ordre. L'intuition est que si φ contient des compositions fonctionnelles, c'est que de l'information utile se trouve à distance 2 ou plus de x : il suffit alors d'utiliser des prédicats unaires supplémentaires pour *rapatrier* l'information à distance 1 de x.

Lemme 6.10 *Toute formule locale $\varphi = \exists\overline{U}\forall x\ \psi(\overline{U}, \overline{f}, \overline{L}, x)$ est équivalente à une formule locale $\varphi' = \exists\overline{U'}\forall x\ \psi'(\overline{U'}, \overline{f}, \overline{L}, x)$ en CNF et sans composition de fonctions.*

> **Preuve** Afin d'éliminer toutes les compositions fonctionnelles de φ, i.e., les formules atomiques de la forme $P(g_n \circ g_{n-1} \circ \cdots \circ g_1(x))$, avec $n \geq 2$, $\{g_1, \cdots, g_n\} \subseteq \overline{f}$, et $P \in \overline{L} \cup \overline{U}$, il suffit d'augmenter le coloriage $\overline{U}$ en un nouveau coloriage $\overline{U'}$ par le procédé suivant : On commence par poser $\psi' = \psi$, et $\overline{U'} = \overline{U}$. Puis, pour chaque formule atomique α dans ψ de la forme $P(g_n \circ g_{n-1} \circ \cdots \circ g_1(x))$ avec $n \geq 2$ et $P \in \overline{L} \cup \overline{U}$:
> - On ajoute à $\overline{U'}$ les prédicats $P^{g_n}, P^{g_n \circ g_{n-1}}, \cdots, P^{g_n \circ \cdots \circ g_2}$;
> - On remplace α dans ψ' par $P^{g_n \circ \cdots \circ g_2}(g_1(x))$;
> - Pour tout $2 \leq j < n$, on ajoute dans ψ' les clauses correspondant à la contrainte $P^{g_n \circ \cdots \circ g_j}(x) \iff P^{g_n \circ \cdots \circ g_{j+1}}(g_j(x))$;
> - On ajoute dans ψ' les clauses correspondant à la contrainte $P^{g_n}(x) \iff P(g_n(x))$.
>
> ■

Notons que les couleurs nouvellement ajoutées dans $\overline{U'}$ sont complètement déterminées par les anciennes sous φ', et donc une structure $\mathcal{S}$ a toujours le même nombre de solutions pour φ (valeurs de $\overline{U}$) que pour φ' (valeurs de $\overline{U'}$).

Muni de ce lemme, nous pouvons maintenant supposer que ϕ est sans composition fonctionnelle, et prouver la partie $(2 \Longrightarrow 1)$ de la Prop. 6.9. Il s'agit de construire un sous-système PLAN-SAT de taille $O(d(x))$ simulant la contrainte ψ autour d'un élément x de degré total $d(x)$. On utilise abondamment le crossover parcimonieux de Lichtenstein [62] pour PLAN-SAT, présenté lors des préliminaires :

Preuve (de la partie $(2 \Longrightarrow 1)$ de la Prop. 6.9) Il nous faut plusieurs gadgets intermédiaires pour construire notre simulateur, que l'on nommera VAR :

– Etant données n variables booléennes plongées sur le côté Ouest d'un carré, il est facile de les dupliquer sur le côté Sud et le côté Est de ce carré en utilisant $O(n^2)$ crossovers : voir le gadget DUP de la Fig. 6.1. Si $n = O(1)$, alors la taille de ce gadget est aussi $O(1)$.

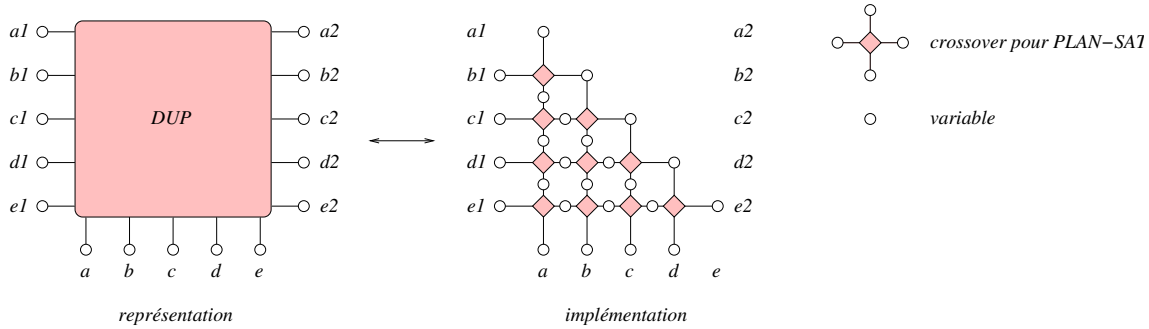


FIG. 6.1 – Le gadget DUP

– Etant donnée une formule propositionnelle ψ en CNF de m clauses portant sur n variables, on peut facilement simuler ψ dans le plan, en chaînant m gadgets DUP de côté n, connectés à leurs voisins par les côtés Ouest et Est, et en laissant les n côtés Sud libres pour y brancher les m clauses de ψ : voir le gadget PSI de la Fig. 6.2. Là encore, le gadget est de taille $O(1)$ si $m = O(1)$ et $n = O(1)$.

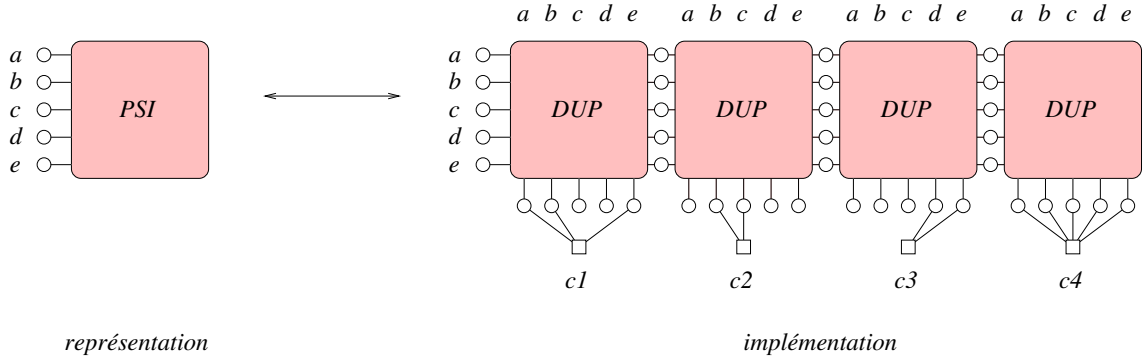


FIG. 6.2 – Le gadget PSI

– Le gadget VAR est construit ainsi : Pour une structure $\mathcal{S} = \langle D, \overline{f}, \overline{L} \rangle$, on choisit $n = (k+1) \times (p+q)$, où $k = |\overline{f}|$, $p = |\overline{L}|$ et $q = |\overline{U}|$. Pour un élément x donné de D, les variables booléennes plongées le long d'un côté Ouest d'un gadget DUP représentent, dans l'ordre, du Nord au Sud : un premier bloc portant les $p+q$ valeurs de $\overline{L}(x)$ et $\overline{U}(x)$ pour un ordre fixé sur $\overline{L} \cup \overline{U}$, suivi de k blocs analogues

portant chacun les $p+q$ valeurs de $\overline{L}(f_i x)$ et $\overline{U}(f_i x)$, pour $1 \leq i \leq k$, mais pour l'ordre inverse de celui fixé pour le premier bloc. Ces $k+1$ blocs forment un "slot". Si $d = d(x)$ est le degré total de x, on chaîne $d-1$ gadgets DUP connectés par les slots Ouest et Est, le dernier gadget DUP étant connecté par son slot Est au côté Ouest d'un gadget PSI. On fixe les labels $\overline{L}$ de x par des clauses unitaires sur le premier bloc du premier slot. Les $d-1$ slots Sud des gadget DUP plus le slot Ouest du premier gadget DUP constituent les d slots libres $s_i(x)$, $1 \leq i \leq d$, du gadget VAR qui vont permettre les connections aux gadgets VAR des d voisins de x : voir Fig. 6.3. Le gadget VAR est bien de taille $O(d)$.

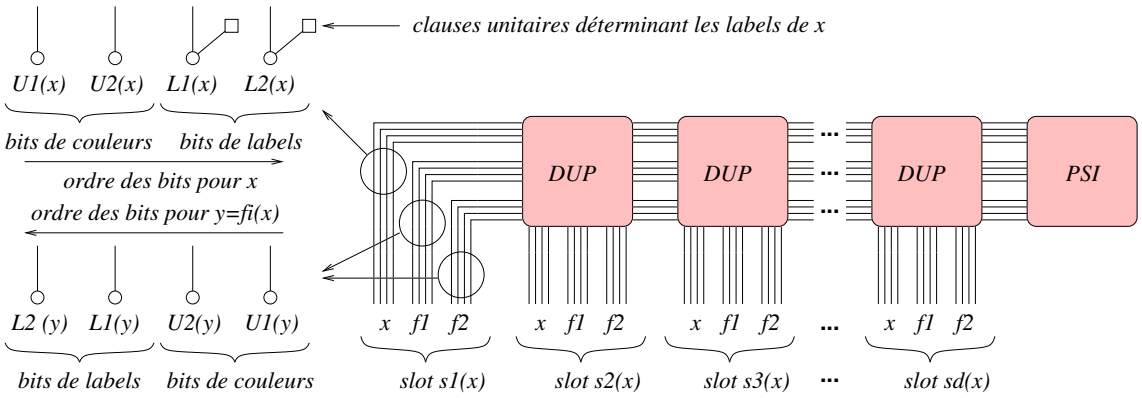


FIG. 6.3 – Le gadget VAR

– La réduction est la suivante : Pour tout élément x de degré total $d = d(x)$ dans D, construire un gadget VAR(x) muni de d slots libres. Chaque flèche $(x, y = f_i(x))$ est associée à une paire de slots libres, disons $s_a(x)$ dans VAR(x) et le slot $s_b(y)$ dans VAR(y), en respectant le plongement planaire de $\mathcal{S}$: on fusionne simplement les variables du bloc f_i de $s_a(x)$ avec les variables correspondantes du bloc y de $s_b(y)$, afin de rapatrier dans VAR(x) les bits d'informations stockés dans VAR(y) qui sont nécessaires au gadget PSI contenu dans VAR(x). Comme on peut le voir sur la Fig. 6.4, cela s'effectue sans croisement dans le plan, grâce à l'ordre inverse choisi pour les blocs f_i à l'intérieur de chaque slot.

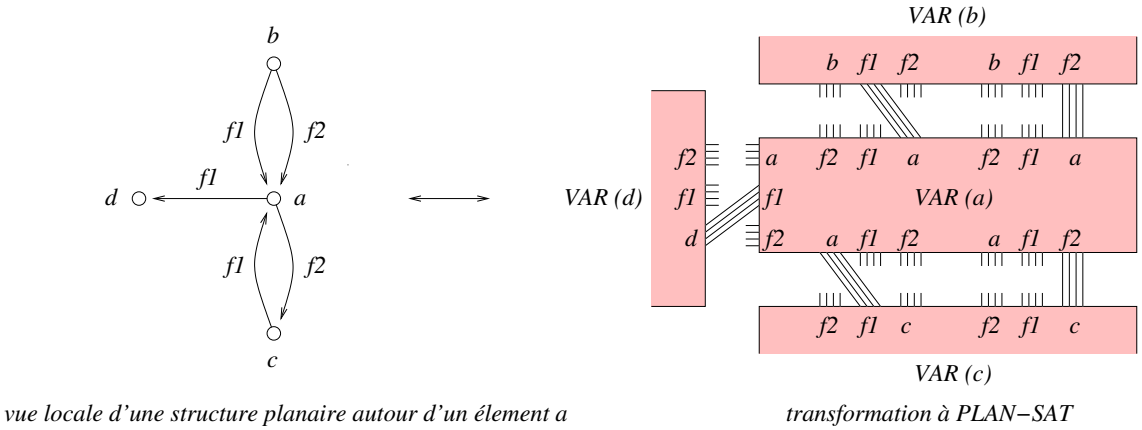


FIG. 6.4 – Les gadgets VAR connectés ensemble

■

Ceci termine la preuve de la LIN-LOCAL-difficulté de SAT et de la LIN-PLAN-LOCAL-difficulté de PLAN-SAT. Combinées avec la LIN-LOCALITE de SAT et la LIN-PLAN-LOCALITE de PLAN-SAT (qui vont être démontrées de plusieurs manières dans les quatre sections suivantes), les équivalences linéaires, obtenues dans les deux derniers chapitres, de PLAN-HAMILTON et PLAN-SAT d'un côté, et de VERTEX-COVER et SAT d'un autre côté, montrent :

Corollaire 6.11 *Toutes les variantes du probleme* PLAN-HAMILTON *sont LIN-PLAN-LOCAL-complètes.*

Corollaire 6.12 VERTEX-COVER *est LIN-LOCAL-complet.*

La réduction de SAT à VERTEX-COVER étant planaire, on a également :

Corollaire 6.13 PLAN-VERTEX-COVER *est LIN-PLAN-LOCAL-dur.*

En fait, le simple gadget SAT de taille $O(n)$ additionnant $O(n)$ bits, exposé à la fin du chapitre 4, montre que la classe LIN-LOCAL reste inchangée si l'on augmente la formule locale avec des contraintes globales de cardinalité :

Corollaire 6.14 *Les problèmes linéairement réductibles aux problèmes sur structures unaires caractérisés par une formule locale "augmentée" de la forme suivante restent des problèmes de classe LIN-LOCAL :*

$$\exists \overline{U}\ \forall x \quad \psi(\overline{U}, \overline{f}, \overline{L}, x) \wedge \psi'(\overline{U}, \overline{L})$$

où ψ' est une combinaison booléenne de "formules atomiques" de la forme $\#P_1 \perp \#P_2$, avec $P_1, P_2 \in \overline{L} \cup \overline{U}$, et $\perp \in \{\leq, \geq, <, >, =, \neq, \}$.

Dans les quatre sections qui suivent, La LIN-LOCALITE de SAT et la LIN-PLAN-LOCALITE de PLAN-SAT vont être prouvées de quatre façons différentes, par des transformations linéaires et planaires de SAT (resp. PLAN-SAT) à des problèmes locaux différents, tous sur structures bijectives. A chaque fois, nous raffinerons notre stratégie afin d'obtenir la forme la plus pure possible pour le problème local d'arrivée, tant au niveau de la signature de la structure que de la formule locale. Au final, nous obtiendrons l'*unique* problème NP-complet logiquement défini sur structures fonctionnelles qui soit *minimal* sous plusieurs critères sur la signature de la structure et sur la formule locale.

6.3 Un problème NP-complet sur des structures bijectives

Par un simple codage de SAT, il est facile d'obtenir un problème NP-complet sur des structures bijectives n'utilisant que deux bijections si l'on ne s'impose pas d'autres restrictions sur la signature et sur la formule.

L'encodage est le suivant. Soit I une instance de SAT . I est encodée en une $\langle D, \overline{f}, \overline{L}\rangle$-structure $\mathcal{S}$ munie de deux bijections $\overline{f} = (e, n)$ et deux labels $\overline{L} = (N, F)$ où :

- Le domaine D est l'ensemble des occurrences des variables booléennes de I.
- Les occurrences d'une même variable sont reliées ensemble par un e-circuit (e comme "équivalence") qui assure la cohérence de leur valuation.
- Les occurrences de variables dans une même clause sont reliées par un n-circuit pour un ordonnancement arbitraire de celle-ci (n comme "next").

– Le label N (N pour "Negative") indique si une occurrence est négative.
– Le label F (F pour "First") indique si une occurrence est la première dans sa clause pour un ordonnancement arbitraire de celle ci.

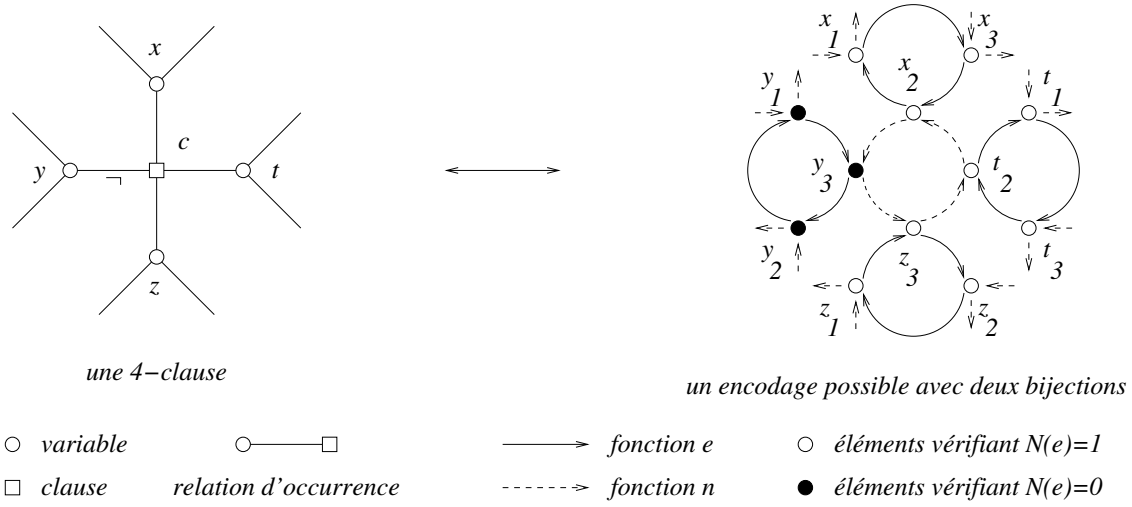


FIG. 6.5 – Encodage de la clause $x \vee \neg y \vee z \vee t$ pour PLAN-Π_1

La formule utilise deux prédicats quantifiés existentiellement : A et T. Soit $c = \ell_1 \vee \cdots \vee \ell_k$ une clause de longueur k. Pour $1 \leq j \leq k$, on note $\text{prefix}_j(c)$ la clause partielle $\ell_1 \vee \cdots \vee \ell_j$. L'idée est de coder l'assignement des variables par le prédicat T (T comme "True") et de maintenir dans A (A comme "Accumulator") les valeurs de vérités accumulées de $\text{prefix}_j(c)$ pour j croissant, d'où l'intérêt du marqueur $First$ qui permet de repérer à la fois le début et la fin de la clause ($next$ décrivant un circuit le long de C, x est la dernière occurrence dans la clause ssi Fnx, i.e., son suivant est la première occurrence dans la clause). On doit forcer A à s'aligner sur la valeur de T en début de clause et forcer A à être vrai en fin de clause. Voir Fig. 6.5. Le problème, que l'on nommera Π_1, est décrit par la formule φ_1 suivante :

$$\Pi_1 : \quad \mathcal{S} = \langle D, (e,n), (N,F) \rangle \quad \models \quad \varphi_1$$

$$\varphi_1 : \quad \exists T, A \; \forall x \quad \bigwedge \begin{cases} (Nx \iff Nex) \iff (Tx \iff Tex) & (i) \\ Fx \implies (Ax \iff Tx) & (ii) \\ \neg Fnx \implies (Anx \iff (Tnx \vee Ax)) & (iii) \\ Fnx \implies Ax & (iv) \end{cases}$$

La linéarité, la correction et la parcimonie de la transformation sont immédiates (remarquons également sur la Fig. 6.5 qu'elle préserve la planarité) :
– La règle (i) assure que toutes les occurrences d'une même variable ont des valuations cohérentes.
– La règle (ii) initialise l'accumulateur du premier littéral de chaque clause c.
– La règle (iii) accumule les valeur de vérité des préfixes de c pour tous les accumulateurs qui ont un prédécesseur.
– La règle (iv) force le dernier accumulateur à témoigner que c est satisfaite.
Nous pouvons conclure que :

Propriété 6.15 SAT *et* PLAN-SAT *sont resp. réductibles à* Π_1 *et* PLAN-Π_1, *linéairement et parcimonieusement.*

Ce premier encodage semble facilement améliorable sur un point : on doit pouvoir se débarrasser du label N encodant le signe des occurences en doublant la taille du domaine à raison d'un élément a_v par occurrence de variable v, plus son successeur $e(a_v)$ représentant son opposé, et en forçant la valeur de T à alterner le long des e-circuits. Il semble moins évident de se débarrasser de l'un des deux prédicats quantifiés existentiellement.

6.4 Un seul prédicat unaire dans la signature et au second ordre

Dans cette section, nous montrons qu'il existe un problème NP-complet sur les structures bijectives n'utilisant que deux bijections, un seul label et un seul prédicat au second ordre, toujours par encodage de SAT. La première idée est d'exploiter la remarque précédente pour se débarrasser du label de signe N. La seconde idée est de se débarrasser du marqueur de début de clause F en codant cette information implicitement dans l'agencement des fonctions. Mais pour cela, nous avons besoin de séparer les éléments "occurrences" des éléments "accumulateurs".

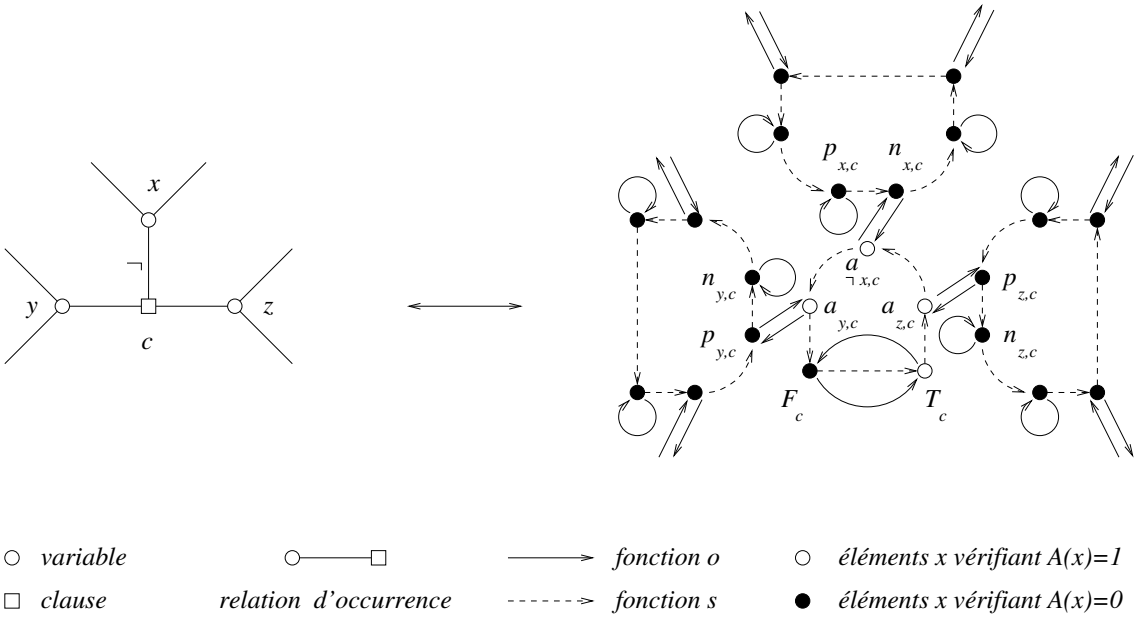


FIG. 6.6 – Encodage de la clause $\neg x \vee y \vee z$ pour PLAN-Π_2

L'encodage est le suivant. Soit I une instance de SAT. I est encodée par une $\langle D, \overline{f}, \overline{L}\rangle$-structure $\mathcal{S}$ munie de deux bijections $\overline{f} = (s, o)$ et d'un label $\overline{L} = (A)$ où :

- Le domaine D se divise en deux parties. Premièrement, D comprend l'ensemble des occurrences de variables et de leurs opposés : Pour chaque occurrence v_c d'une variable v dans une clause c de I, on note $p_{v,c}$ et $n_{v,c}$ les éléments associés à v_c, resp. positifs et négatifs. Deuxièment, D comprend un élément $a_{\ell,c}$ pour tout littéral ℓ apparaissant dans une clause c. Cet élément a pour rôle d'accumuler les valeurs de vérité sur le préfixe correspondant de la clause c (disjonction partielle). Enfin, chaque clause c a deux élements associé F_c et T_c (F_c pour "False" et "First", T_c pour "True" et "Terminal"), que l'on forcera à représenter resp. les constantes fausse et vraie, et qui nous permettront de marquer le début et la fin de la clause c.
- s est la fonction successeur à la fois pour les occurrences et les accumulateurs : Les éléments $p_{v,c}$ et $n_{v,c}$ associés à une même variable v décrivent un n-circuit tel que $n_{v,c} = s(p_{v,c})$ afin

d'avoir un circuit alternant les signes ; De plus, pour chaque clause $c = \ell_1 \vee \cdots \vee \ell_k$ ainsi ordonnée, on construit le s-circuit $(T_c, a_{\ell_k,c}, \cdots, a_{\ell_1,c}, F_c)$. Notons que l'accumulation des valeurs le long des prefixes de c se fera à rebours de s.

- o est principalement la fonction d'occurrence : Pour toute occurrence d'une variable v dans une clause c sous la forme du littéral ℓ, soit $a_{\ell,c}$ l'élément accumulateur associé, $y = p_{v,c}$ ou $n_{v,c}$ l'élément occurrence associé du signe de ℓ, et $z = n_{v,c}$ ou $p_{v,c}$ l'élément occurrence de signe opposé, inutilisé. On construit le o-circuit $(a_{\ell,c}, y)$, ainsi que la o-boucle (z).
- L'astuce forcant le bon comportement de T_c et F_c consiste à construire le o-circuit (F_c, T_c) pour toute clause c, ainsi qu'à considérer T_c comme un élement accumulateur, mais pas F_c : Les éléments vérifiant le prédicat A (A comme "Accumulateur") sont exactement les $a_{c,v}$ et les T_c.

Voir la Fig. 6.6 pour la construction. La formule utilise cette fois un seul prédicat quantifié existentiellement, T (T pour $True$) qui est censé porter :

- la valeur de vérité des occurrences pour les $p_{v,c}$ et les $n_{v,c}$,
- la valeur de vérité des préfixes de clauses pour les $a_{\ell,c}$,
- la valeur vraie (resp. fausse) pour les T_c (resp. les F_c).

Le problème Π_2 est décrit par la formule φ_2 suivante :

$$\Pi_2 : \quad \mathcal{S} = \langle D, (s, o), (A) \rangle \quad \models \quad \varphi_2$$

$$\varphi_2 : \quad \exists T \; \forall x \quad \bigwedge \begin{cases} \neg Ax \Longrightarrow (Tx \oplus Tsx) & (i) \\ Ax \Longrightarrow (Tx \iff (Tsx \vee Tox)) & (ii) \end{cases}$$

La linéarité de la construction et sa préservation de la planarité sont immédiates. En ce qui concerne la correction, remarquons tout d'abord que les s-flèches $(F(c), T(c))$, $(y, a_{\ell,c})$ et la o-boucle (z) pour $z = p_{v,c}$ ou $n_{v,c}$ sont "muettes" : elles ne véhiculent aucune contrainte puisque :

- seule la règle (ii) de la formule utilise la fonction o, et
- cette règle est gardée par la prémisse Ax, qui est fausse pour les antécédents de ces s-flèches.

La cohérence de la valuation des éléments occurrences d'une même variable découle trivialement de la règle (i) le long des s-circuits reliant les occurrences. Ensuite, pour toute clause $c = \ell_1 \vee \cdots \vee \ell_k$ ainsi ordonnée :

- Comme $s(F_c) = T_c$ et $A(F_c) = 0$, la règle (i) force $T(F_c) = \neg T(T_c)$.
- De plus, comme les éléments x de la s-chaine $(T_c, a_{\ell_k,c}, \cdots, a_{\ell_1,c}, F_c)$ vérifient tous $A(x) = 1$ à l'exception de F_c, la règle (ii) force la séquence de valeurs booléennes $T(x)$ à être monotone décroissante le long de cette chaine (rappelons que l'accumulation se fait à rebours des s-flèches).

Par conséquent, on a à la fois $T(T_c) \geq F(T_c)$ et $T(T_c) \neq T(F_c)$, c'est-à-dire $T(F_c) = 0$ et $T(T_c) = 1$. La séquence monotone croissante le long de la s-chaine est donc ultimement non nulle, et il existe un unique j tel que $T(a_{\ell_j,c}) = 1$ et $T(s^{-1}a_{j,c}) = 0$, ce qui, par la règle (ii), implique $T(y) = 1$, pour $y = o(a_{j,c})$, c'est-à-dire l'occurrence $n_{v,c}$ ou $p_{v,c}$.

Toute clause c est donc satisfaite, et il y a une bijection entre les valuations V satisfaisant I et les prédicats T satisfaisant $\mathcal{S}$:

- Pour toute variable v et toute clause c de I, $V(v) = T(p_{v,c}) = 1 - T(n_{v,c})$
- Pour toute clause $c = \ell_1 \vee \cdots \vee \ell_k$ de I, et pour tout $1 \leq j \leq k$, $T(a_{\ell_j}) = V(\text{prefix}_j(c))$.
- Pour toute clause c, $T(T_c) = 1$ et $T(F_c) = 0$.

Ceci démontre la correction et la parcimonie de la transformation :

Propriété 6.16 SAT *et* PLAN-SAT *sont resp. réductibles à* Π_2 *et* PLAN-Π_2, *linéairement et parcimonieusement.*

Jusque là, nos deux transformations ont en commun d'exploiter les prédicats-labels des éléments comme prémisses à gauche d'une implication afin de court-circuiter les règles selon la nature des éléments. Ce dernier label peut être éliminé : Dans la section suivante, nous montrons qu'il existe un problème NP-complet sur les structures bijectives n'utilisant que deux bijections, aucun label et toujours un seul prédicat au second ordre.

6.5 Elimination totale des prédicats unaires dans la signature

Afin d'éliminer le dernier prédicat unaire (label) de la signature, nous exploitons les deux idées développées au cours des deux transformations précédentes, à savoir :

- pour chaque variable, un circuit pour ses occurrences dédoublées, en alternant les signes positifs et négatifs.
- l'opposition des valeurs de vérité de deux éléments spéciaux et contigus dans le circuit d'accumulation de la valeur de vérité des préfixes de clauses, combinée avec la monotonie décroissante de la valeur de vérité circulant le long de ce circuit.

La difficulté est que les éléments occurrences et les éléments accumulateurs "ne connaissent pas leur identité" puisqu'ils ne disposent pas du prédicat "label" pour les renseigner.

Notre formule va principalement exploiter les propriétés de l'opérateur logique NAND, et en particulier le fait qu'il soit complet pour la logique propositionnelle, c'est-à-dire que tous les autres opérateurs booléens s'expriment en fonction de NAND :

- NOT(x) = NAND (x, x) = NAND $(x, 1)$, et
- OR(x)= NAND (NOT (x),NOT (x)).

Dans l'encodage de SAT que nous allons construire, les deux bijections de la structure n'ont pas vraiment une sémantique particulière (elles sont même interchangeables, la formule étant symétrique à leur égard) : nous les appellerons donc simplement f et g. Le problème, que l'on nommera Π_{nand} est décrit par la formule suivante :

$$\Pi_{\text{nand}} : \quad \mathcal{S} = \langle D, (f, g), \emptyset \rangle \quad \models \quad \varphi_{\text{nand}}$$

$$\varphi_{\text{nand}} : \quad \exists U \ \forall x \quad Ux \iff \neg(Ufx \wedge Ugx)$$

qu'il sera intéressant par la suite d'écrire sous la forme clausale équivalente :

$$\varphi_{\text{nand}} : \quad \exists U \ \forall x \quad \bigwedge \begin{cases} \neg Ux \vee \neg Ufx \vee \neg Ugx & (i) \\ Ux \vee Ufx & (ii) \\ Ux \vee Ugx & (iii) \end{cases}$$

Les propriétés du NAND nous permettront de construire nos accumulateurs. En ce qui concerne la première propriété, forcer y = NOT (x) par la construction y = NAND (x, x) n'est pas très pratique pour l'exploitation future de x dans une structure avec deux bijections, car la construction monopolise les deux seules flèches disponibles vers x. La construction y = NAND $(x, 1)$ semble plus viable, mais il faut être capable de générer la constante vraie de telle façon qu'une flèche soit toujours disponible pour pointer sur cette constante vraie. C'est le rôle du gadget suivant :

Propriété 6.17 *Le gadget True, montré sur la Fig. 6.7 et décrit par le g-circuit (α, β), le f-circuit (β, γ) et la f-boucle (α), n'admet qu'une valuation U satisfaisant φ_{nand} : $U(\alpha) = 1$, $U(\beta) = 0$, $U(\gamma) = 1$. Une g-flèche reste disponible pour pointer sur γ et consulter la constante $U(\gamma) = 1$. De plus, la g-flèche sortant du gadget est "muette", i.e., $U(g\gamma)$ est à priori libre.*

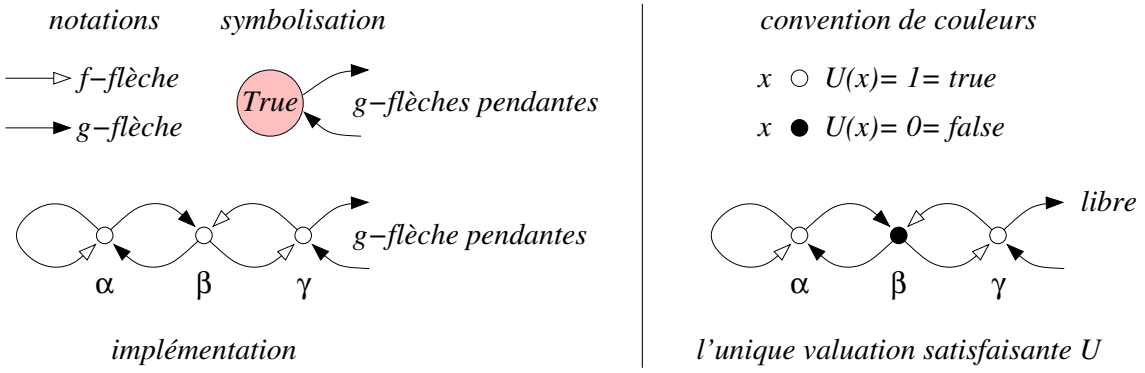


FIG. 6.7 – Le gadget True pour le problème Π_{nand}

Preuve La règle (ii) appliquée à α implique $U(\alpha) = 1$ à cause de la f-fléche (α). Ensuite, la règle (i) appliquée à α implique $U(\beta) = 0$ à cause de la g-flèche (α, β) Enfin, la règle (i) appliquée à β implique $U(\gamma) = 1$ à cause de la f-flèche (β, γ). $U(g\gamma)$ reste libre car quel que soit $U(g\gamma)$:
la règle (i) appliquée à γ est de toute façon satisfaite par $U(f\gamma) = U(\beta) = 0$.
– les règles (ii) et (iii) appliquées à γ sont de toute façon satisfaites par $U(\gamma) = 1$.
Un seul assignement est donc possible, et on vérifie facilement qu'il satisfait les trois règles $(i) - (iii)$ en chacun des trois points α, β, γ. ■

Remarque 6.18 *La règle (iii) est inutile à la fois pour la correction et la parcimonie du gadget True.*

Par la suite, on notera $g(True) = x$ lorsque $g(\gamma) = x$ et $g(y) = True$ lorsque $g(y) = \gamma$, pour le sommet γ d'un gadget True quelconque. De même un g-circuit $(s, \cdots, \gamma, \cdots, t)$ sera noté $(s, \cdots, True, \cdots, t)$.

L'encodage de SAT est le suivant (voir Fig 6.8). Soit I une instance de SAT :

- Pour chaque variable x_i de I ayant k occurences, on crée $2k$ éléments x_i^j, nx_i^j, pour $1 \leq j \leq k$, que l'on relie par le f-circuit $(x_i^1, nx_i^1, \cdots, x_i^k, nx_i^k)$. Les x_i^j et nx_i^j représentent resp. x_i et $\neg x_i$.
- Pour chaque clause $c_i = \ell_1 \vee \cdots \vee \ell_k$ de I et ainsi ordonnée, on crée un élément nc_i^0 et $2k$ éléments c_i^j, nc_i^j, pour $1 \leq j \leq k$, que l'on relie par le f-circuit $(nc_i^k, c_i^k, nc_i^{k-1}, c_i^{k-1}, \cdots, nc_i^1, c_i^1, nc_i^0)$. Pour chacun des $k+1$ nc_i^j, $(0 \leq j \leq k)$, on créé un gadget True, et l'on crée le g-circuit $(nc_i^j, True)$. Un élément c_i^j représente l'accumulateur pour $\text{prefix}_j(c_i)$ et nc_i^j représente son opposé. L'élement nc_j^0 marque le début et la fin de la clause.
- Pour la $j^{\text{ème}}$ occurrence d'une variable v_i apparaissant sous la forme du littéral $\ell_{j'}$ dans une clause $c_{i'} = \ell_1 \vee \cdots \ell_k$, on crée deux gadgets True et les deux g-circuits $(c_{i'}^{j'}, y, True)$ et $(z, True)$, où :
  - $y = nx_i^j$ et $z = x_i^j$ si $\ell_{i'}^{j'}$ est positif.
  - $y = x_i^j$ et $z = nx_i^j$ si $\ell_{i'}^{j'}$ est négatif.

Propriété 6.19 *La construction exposée ci-dessus est une transfomation linéaire et parcimonieuse de* SAT, *resp.* PLAN-SAT, *au problème* Π_{nand}, *resp.* PLAN-Π_{nand}.

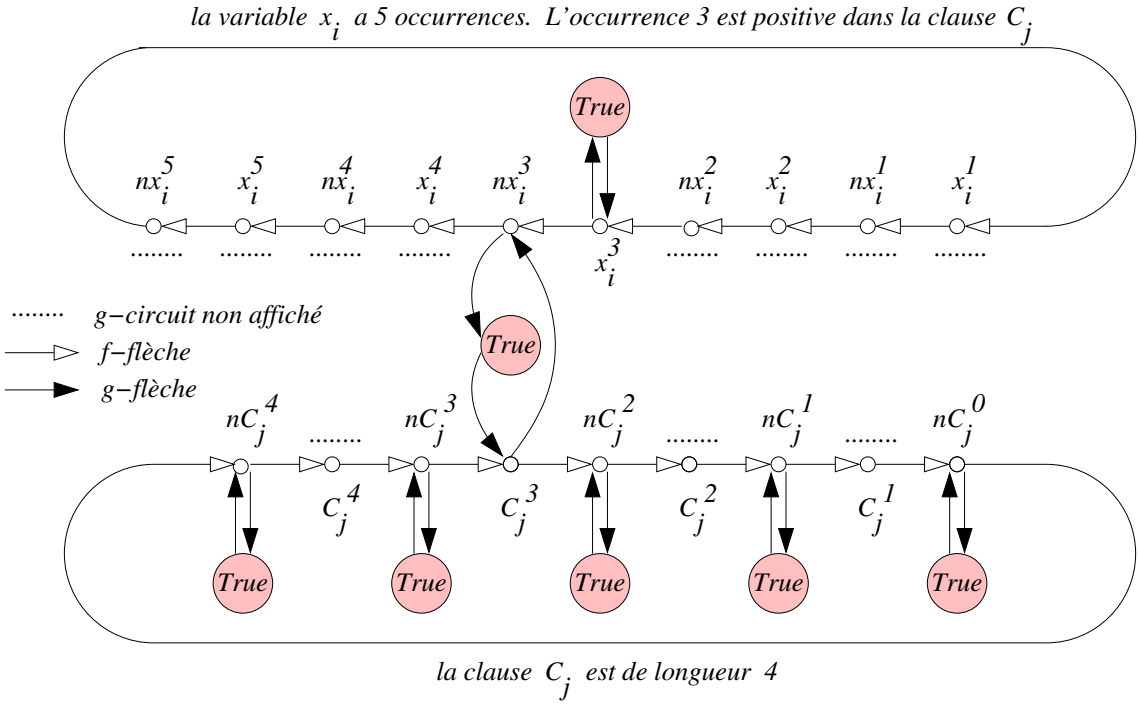


FIG. 6.8 – Encodage d'une clause pour PLAN-Π_{nand}

Pour des raisons de présentation, nous découpons la preuve en deux parties. La première sera réutilisable pour établir un résultat de la section suivante.

Preuve (première partie) Soit U un prédicat sur D satisfaisant φ_{nand}. Notons que quels que soit les booléens a et b, $a = \text{NAND}\ (b, 1)$ implique $a = 1 - b$ par les règles (i) et (ii). Ce qui signifie que si l'on a $f(x) = y$ et $g(x) = True$, la f-flèche se comporte comme un inverseur. Ceci établit que :

– L'assignement des occurrences est cohérente :

$$U(nx_i^1) = \cdots = U(nx_i^k) = 1 - U(x_i^1) = \cdots = 1 - U(x_i^k)$$

pour toute variable x ayant k occurrences. En effet : chacun des x_i^j a sa g-flèche vers un gadget True et sa f-flèche vers nx_i^j, donc $nx_i^j = 1 - x_i^j$ pour tout j ; De plus, chacun des nx_i^j a aussi sa g-flèche vers un gadget True et sa f-flèche vers x_i^{j+1} (en posant $j + 1 = 1$ si $j = k$), donc $U(x_i^{j+1}) = 1 - x_i^j$ pour tout j. D'où le résultat.

– Pour toute clause c_i de longueur k et tout $1 \leq j \leq k$, $U(nc_i^j)$ est bien l'opposé de $U(c_i^j)$ à cause de la g-flèche $(nc_i^j, True)$ et de la f-flèche (nc_i^j, c_i^j). De plus $U(c_i^0) = 1 - U(nc_i^k) = U(c_i^k)$, à cause de la g-flèche $(nc_i^0, True)$ et de la f-flèche (nc_i^0, c_i^k).

Remarque 6.20 *Notons que la règle (iii) n'intervient pas dans cette première partie de preuve.*

Preuve (deuxième partie) Pour toute clause c_i de longueur k et tout $1 \leq j \leq k$, on a $U(c_i^j) = \text{NAND}\ (U(y), U(nc_i^{j-1}))$ où $U(y)$ est l'opposé de la valeur du $j^{\text{ème}}$ littéral ℓ_j dans c_i, et $U(nc_i^{j-1}) = 1 - U(c_i^{j-1})$. Donc $U(c_i^j) = \text{OR}\ (U(\ell_j), U(c_i^{j-1}))$:

les c_i^j sont bien des accumulateurs et la séquence des $U(c_i^j)$ est monotone croissante pour j croissant, à rebours des f-flèches.
Il découle que $U(nc_i^0) = 1$: en effet, si l'on avait $U(nc_i^0) = 0$, on aurait immédiatement $U(c_i^1) = 1$ et la séquence monotone croissante serait constante, égale à 1, avec en particulier $U(c_i^k) = 1$, une contradiction puisque $U(nc_i^0) = U(c_i^k)$. Le rôle de nc_i^0 est donc double :

- Il initialise la séquence d'accumulation : $U(c_i^1) = \text{NAND}\,(U(nc_i^0), U(y))$ où $U(y)$ est l'opposé de la valuation du littéral ℓ_1, et donc $U(c_i^1)$ représente bien l'assignement de ℓ_1 comme voulu : $U(c_i^j)$ est bien la U-valuation de $\text{prefix}_j(c_i)$ pour tout $1 \leq j \leq k$.
- Il termine la séquence d'accumulation en forçant $U(c_i^k) = U(nc_i^0) = 1$. La séquence monotone croissante des $U(c_i^j)$ initialisée à $U(\ell_0)$ doit donc être ultimement non nulle, ce qui signifie que la clause doit être validée.

■

On voit que jusqu'à une étape avancée de la preuve, on n'utilise pas entièrement les propriétés de l'opérateur NAND, mais uniquement les règles (i) et (ii). On peut donc se demander si l'on peut encore appauvrir la formule, ici en termes de longueur et de nombre de clauses, et toujours obtenir une formule capturant des problèmes NP-complets. Dans la section suivante, nous montrons que l'on peut effectivement se débarrasser de la règle (iii).

6.6 L'unique problème minimalement NP-complet

Soit $\Pi_{\min}$ le problème LIN-LOCAL défini sur les structures bijectives $\mathcal{S} = \langle D, \overline{f}, \overline{L}\rangle$ utilisant deux bijections $\overline{f} = (f, g)$ et aucun prédicat "label" (i.e., $L = \emptyset$), et décrit par la formule $\varphi_{\min}$ suivante n'utilisant qu'un prédicat quantifié existentiellement :

$$\Pi_{\min} : \quad \mathcal{S} = \langle D, (f, g), \emptyset\rangle \quad \models \quad \varphi_{\min}$$

$$\varphi_{\min} : \quad \exists U\ \forall x \quad \bigwedge \begin{cases} \neg Ux \vee \neg Ufx \vee \neg Ugx & (i) \\ Ux \vee Ufx & (ii) \end{cases}$$

dont la forme DNF équivalente est $\delta_{\min}$:

$$\delta_{\min} : \quad \exists U\ \forall x \quad \bigvee \begin{cases} Ux \wedge \neg Ufx \\ Ux \wedge \neg Ugx \\ Ufx \wedge \neg Ux \\ Ufx \wedge \neg Ugx \end{cases}$$

Il s'agit de φ_{nand} débarrassée de sa règle (iii). Nous allons montrer que non seulement le problème $\Pi_{\min}$ est NP-complet, mais aussi que la formule $\varphi_{\min}$ qui le définit est la formule minimale, sous plusieurs critères, et unique (à quelques symétries près) qui définit un problème NP-complet. Nous établissons tout d'abord que :

Propriété 6.21 *La même réduction linéaire et parcimonieuse de* SAT*, resp.* PLAN-SAT*, au problème* Π_{nand}*, resp.* PLAN-Π_{nand}*, (cf. Propriété 6.19) est également une réduction linéaire mais cette fois non parcimonieuse de* SAT*, resp.* PLAN-SAT*, au problème* $\Pi_{\min}$*, resp.* PLAN-$\Pi_{\min}$*.*

La formule $\varphi_{\min}$ étant φ_{nand} débarrassée de sa règle (*iii*), la remarque 6.18 nous permet d'affirmer que le gadget *True* continue de fonctionner correctement et parcimonieusment avec $\varphi_{\min}$. De plus, par la remarque 6.20, il suffit juste de réécrire la deuxième partie de la preuve de la Propriété 6.19. Ce que l'on perd dans cette partie est la correspondance bijective entre la valuation du prefixe $\text{prefix}_j(c_i)$ d'une clause c_i et $U(c_i^j)$:

Preuve A ce stade, nous savons que toute prédicat U satisfaisant $\varphi_{\min}$ sur la structure $\mathcal{S}$ représente une valuation cohérente des occurrences de chaque littéral (à i fixé, les $U(x_i^j)$ sont identiques pour tout j et les $U(nx_i^j)$ leur sont opposés). De plus, pour toute clause c_i de longueur k, les $U(c_i^j)$ sont opposés aux $nC(c_i^j)$ pour $1 \leq j \leq k$, et $U(nc_i^0) = U(c_i^k)$.

La règle (*ii*) de $\varphi_{\min}$ force $U(cx_i^{j+1}) \vee U(ncx_i^j)$ pour $0 \leq j < k$, c'est-à-dire $U(cx_i^{j+1}) \vee \neg U(cx_i^j)$, que l'on peut écrire comme $U(cx_i^j) \Longrightarrow U(cx_i^{j+1})$, et on obtient à nouveau que la séquence des $U(c_i^j)$ est monotone croissante pour j croissant, à rebours des f-flèches. On réobtient aussi $U(nc_i^0) = 1$: en effet, si l'on avait $U(nc_i^0) = 0$, la règle (*i*) impliquerait $U(c_i^1) = 1$ et la séquence monotone croissante serait constante, égale à 1, avec en particulier $U(C_i^k) = 1$, une contradiction puisque $U(nc_i^0) = U(c_i^k)$. Comme $U(c_i^k) = U(nc_i^0)$, la séquence est ultimement non nulle.

Nous montrons maintenant que pour toute clause $c_i = \ell_1 \vee \cdots \vee \ell_k$, l'élément c_i^j, $j > 1$, joue bien son rôle d'accumulateur lorsque $I(\ell_j) = 0$ mais qu'il peut partiellement ignorer un assignement $I(\ell_j) = 1$ en ne faisant pas basculer la séquence des $U(c_i^j)$ de 0 à 1 :

- Pour $j = 1$, la règle (*i*) force $\neg U(c_i^1) \vee \neg U(nc_i^0) \vee \neg U(y)$, où $U(y) = 1 - I(\ell_1)$ et $U(nc_i^0) = 1$. On a donc $\neg U(c_i^1) \vee I(\ell_1)$, que l'on peut réécrire comme $U(c_i^1) \Longrightarrow I(\ell_1)$.
- Pour $j > 1$, la règle (*i*) force $\neg U(c_i^j) \vee \neg U(nc_i^{j-1}) \vee \neg U(y)$, où $U(y) = 1 - I(\ell_j)$ et $U(nc_i^{j-1}) = 1 - U(c_i^{j-1})$. On a donc $\neg U(c_i^j) \vee U(c_i^{j-1}) \vee I(\ell_j)$ que l'on peut réécrire comme $U(c_i^j) \Longrightarrow I(\ell_j) \vee U(c_i^{j-1})$.

Cependant, la séquence étant monotone croissante, ultimement non nulle, et initialisée à $I(\ell_1)$ ou 0, exactement un des accumulateurs associés à un littéral satisfaisant la clause devra fonctionner correctement en faisant basculer la séquence de 0 à 1. ■

Notons toutefois que ce genre de non parcimonie peut être transformée en parcimonie faible avec un facteur exponentiel sur le nombre de solutions, en utilisant la composition de réductions :

$$\text{SAT} \leq \tfrac{1}{3}\text{-SAT} \leq \text{SAT} \leq \Pi_{\min}$$

Les deux premières réductions, qui ont été présentées dans les préliminaires, sont parcimonieuses. Rappelons que la réduction de $\frac{1}{3}$-SAT à SAT associe à toute $\frac{1}{3}$-clause (x, y, z) le sous-système de clauses :

$$\bigwedge \begin{cases} x \vee y \vee z \\ \neg x \vee \neg y \\ \neg y \vee \neg z \\ \neg z \vee \neg x \end{cases}$$

Tout assignement I satisfiant ce sous-système satisfait la 3-clause et deux 2-clauses *en un unique littéral*. La 2-clause restante est satisfaite *par ses deux littéraux*. Par conséquent, pour l'encodage de chacun de ces sous-systèmes dans la troisième réduction $\text{SAT} \leq \Pi_{\min}$, seule cette dernière 2-clause est source de non parcimonie : le mécanisme d'accumulation pouvant se déclencher sur

deux accumulateurs au choix. Pour un système de k $\frac{1}{3}$-clauses ayant s solutions, il y a donc $2^k \times s$ solutions pour l'instance de $\Pi_{\min}$ obtenue. De plus, comme toutes les réductions de la chaîne préservent la planarité, on peut conclure que :

Corollaire 6.22 *Il existe une réduction linéaire, planaire et faiblement parcimonieuse de* SAT, *resp.* PLAN-SAT, *au problème* $\Pi_{\min}$, *resp.* PLAN-$\Pi_{\min}$.

Nous allons maintenant montrer que $\varphi_{\min}$ est *l'unique* formule sur structures fonctionnelles qui soit minimale pour plusieurs critères tout en exprimant un problème NP-complet. Ces critères sont résumés dans la Table 6.1. L'unicité suppose de prendre en compte un certain nombre de symétries syntaxiques résumées dans la Table 6.2.

signature	2 de fonctions unaires	CNF ($\varphi_{\min}$)	nombre de clauses	2
nombre de symboles EMSO	1		longueur	5
nombre de variables FO	1	DNF ($\delta_{\min}$)	nombre d'anticlauses	3
nombre d'atomes distincts	3		longueur	6

TAB. 6.1 – Critères de minimalité sur la formule EMSO

Théorème 6.23 *Supposons $P \neq NP$. Parmi les problèmes logiquement définis sur des* structures fonctionnelles *par une formule EMSO de la forme :*

$$\exists \overline{U}\ \forall \overline{x} \quad \psi(\overline{U}, \overline{f}, \overline{L}, \overline{x})$$

(où ψ est sans quantificateur, sans composition fonctionnelle, et sans égalité) $\Pi_{\min}$ *est un problème* minimalement *NP-complet pour les critères de la Table 6.1, dans la mesure ou une contrainte plus forte sur l'un de ses critères rend le problème décidable en temps déterministe polynomial même si tous les autres critères sont relachés.*

De plus, $\varphi_{\min}$ *est* l'unique *formule EMSO minimale en CNF qui décrit un problème NP-complet sur* structures bijectives *et sur* structures bijectives planaires *(aux symétries près de la Table 6.2).*

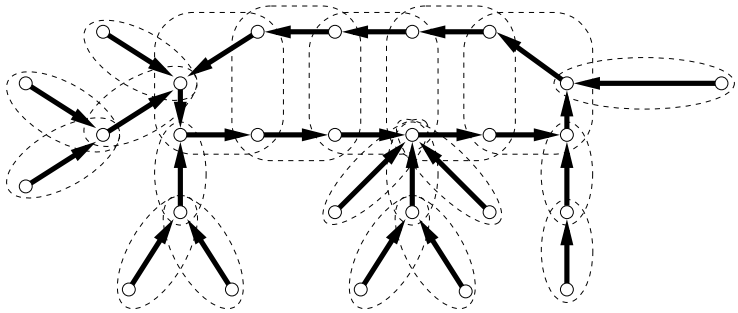
Nous prouvons séparément la minimalité et l'unicité.

Preuve (Minimalité) Examinons chacun des huit critères de minimalité :

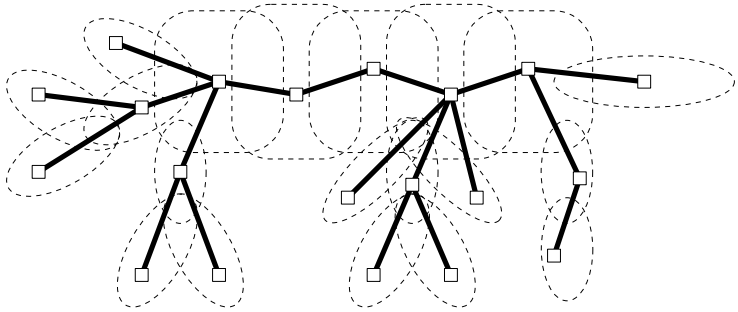
Minimalité de la signature (= 2 fonctions, 0 label) Si l'on n'a qu'une seule fonction et un nombre illimité de labels dans la signature σ, alors les composantes connexes d'une σ-structure $\mathcal{S}$ sont des monocycles, i.e., des arbres dont les racines sont connectées par un circuit. Une telle composante a alors une décomposition arborescente de largeur bornée, i.e., on peut couvrir l'ensemble de ses élements et de ses flèches par un nombre linéaire de sous-ensembles $V_1, \cdots, V_n$ de tailles bornées

symétrie 1	permutation de U et $\neg U$
symétrie 2	permutation de x, fx et gx

TAB. 6.2 – Symétries considérées



Une composante connexe d'une structure fonctionnelle à une seule fonction munie d'une couverture de l'ensemble de ses éléments et de ses flèches par des sous−ensembles de taille maximale 4



Une décomposition arborescente de la structure selon la couverture ci−dessus

FIG. 6.9 – Les monocycles ont une décomposition arborescente de largeur bornée

et construire un arbre T sur les noeuds $V_1, \cdots, V_n$ tel que pour tout élément x de la composante, l'ensemble des V_i contenant x sont connectés dans T.

La Fig. 6.9 montre comment construire une décomposition arborescente de largeur 4 sur un monocycle en temps linéaire : Chaque couple (x, fx) appartenant à un arbre autour du circuit forme un sous-ensemble V_i de taille 2. Un circuit central de longueur $\geq$ 4 est décomposé en sous-ensembles de taille maximale 4 de la façon suivante : Choisir une chaîne de quatre éléments pour former un premier sous-ensemble, puis former un second sous-ensemble avec les deux extrémités de la chaine et leur deux voisins situés sur le circuit mais hors de la chaîne. Continuer ainsi jusqu'à épuisement des éléments du circuit. Le dernier sous-ensemble peut être de taille 3 ou 4 selon la parité de la longueur du circuit. (On peut obtenir une largeur arborescente de 3 mais c'est inutile).

Courcelle [18] a montré que toute propriété monadique du second ordre (MSO), et donc en particulier les propriétés EMSO, est vérifiable en temps déterministe linéaire sur les structures admettant une décomposition arborescente de largeur bornée. Nous ne donnons ici qu'une preuve limitée au cas d'une EMSO avec une seule variable quantifiée universellement au premier ordre.

La décomposition arborescente T de chaque composante connexe d'une telle structure $\mathcal{S}$ nous permet de décider en temps linéaire déterministe si celle-ci est le modèle d'une formule de la forme $\exists \overline{U}\ \forall x\ \psi$. Il suffit de choisir une racine arbitraire dans T, et de procéder de façon "bottom-up" : on traite d'abord les feuilles, puis on remonte dans l'arbre vers la racine, niveau par niveau. En considérant $\overline{U}$ comme un vecteur de

bits indexé sur les éléments de la structure et en notant $\overline{U}[V]$ sa restriction sur un sous-ensemble V, le traitement d'une feuille V_i de père V_j consiste à :

1. générer tous les vecteurs $\overline{U}[V_i \cup V_j]$ possibles. V_i et V_j étant de taille $O(1)$ et le nombre de prédicats étant $O(1)$, il n'y a que $O(1)$ vecteurs à générer, et ceci prend donc un temps $O(1)$.
2. ne conserver que les vecteurs qui vérifient φ développée sur V_i ;

Le traitement d'un noeud interne V_j de père V_k consiste à :

1. trier la liste des vecteurs $\overline{U}[V_i \cup V_j]$ obtenus pour tous les fils V_i de V_j sur leur restriction commune $\overline{U}[V_j]$. Cette restriction étant $O(1)$, le tri s'effectue en temps linéaire, i.e., en temps proportionnel au degré du noeud.
2. ne conserver que les restrictions $\overline{U}[V_j]$ qui possèdent au moins une extension $\overline{U}[V_i \cup V_j]$ pour tout fils V_i. Les vecteurs étant triés sur $\overline{U}[V_j]$, les extensions d'une restriction $\overline{U}[V_j]$ se trouvent en position contiguë dans la liste triée et on peut donc effectuer le filtrage en une passe sur la liste, i.e., en temps proportionnel au degré du noeud.
3. générer toutes les extensions $\overline{U}[V_j \cup V_k]$ possibles à partir des vecteurs $\overline{U}[V_j]$ obtenus.
4. ne conserver que les vecteurs qui vérifient φ développée sur V_j.

Le processus s'arrête lorsqu'un ensemble de vecteurs se vide pour un noeud quelconque. La structure est un modèle de φ si et seulement on peut remonter jusqu'à la racine. Le traitement d'un noeud coûte un temps linéaire en son degré, et donc le coût total pour la structure est $O(|\mathcal{S}|)$.

Minimalité du nombre de symboles EMSO (= 1) S'il n'y a pas de symbole au second ordre, alors la formule est du premier ordre (FO), et décrit une propriété AC_0, donc décidable en temps polynomial (ou en temps linéaire s'il n'y a qu'une variable au premier ordre).

Minimalité du nombre de symboles FO (=1) Trivial.

Minimalité du nombre d'atomes distincts (= 3) S'il n'y a qu'au plus deux atomes distincts dans φ, alors, une fois écrite en CNF, ses clauses sont trivialement de longueur au plus 2, i.e., la formule se déplie linéairement sur la variable x en une donnée du problème 2-SAT, donc décidable en temps linéaire.

Minimalité du nombre de clauses de la CNF (= 2) Toute formule ESO qui n'a qu'une clause en CNF définit un problème trivialement satisfaisable.

Minimalité de la longueur de la CNF (= 5) Si la CNF a une longueur au plus 4, alors seuls trois cas se présentent :

(*i*) Toutes les clauses sont de longueur au plus 2. La formule est alors décidable en temps linéaire, comme il a déjà été mentionné ci-dessus.

(*ii*) Il n'y a qu'une clause de longueur 3 ou 4. Voir paragraphe ci-dessus.

(*iii*) Il y a exactement une 3-clause et une 1-clause. La 3-clause a alors soit au plus un littéral positif, soit au plus un littéral négatif. La formule est donc soit ESO-Horn, soit ESO-Anti-Horn, et donc décidable en temps déterministe polynomial [37].

Minimalité du nombre d'anticlauses dans la DNF (= 3) Toute DNF ne contenant qu'au plus deux anticlauses est équivalente à une CNF contenant uniquement de clauses de longueur au plus 2.

Minimalité de la longueur de la DNF (= 6) Comme il doit y avoir au moins 3 anticlauses d'après l'argument ci-dessus, une DNF de longueur au plus 5 a au moins une anticlause unitaire, ce qui rend le problème trivialement satisfaisable. ■

Nous prouvons maintenant que $\varphi_{\min}$ est l'unique formule EMSO en CNF qui définit un problème minimalement NP-complet sur structures bijectives et sur structures bijectives planaires.

Preuve (Unicité de $\varphi_{\min}$) Soit φ une formule minimalement EMSO en CNF définissant un problème NP-complet. On sait qu'elle doit être de la forme

$$\varphi : \quad \exists U\ \forall x \quad \psi(f, g, U, x)$$

où ψ est la conjonction de deux clauses C_1 et C_2 telles que $|C_1| + |C_2| = 5$, avec $|C_1| < |C_2|$. On peut remarquer deux points :

(i) Une des deux clauses doit être monotone positive, et l'autre doit être monotone négative, sinon, la formule serait toujours trivialement satisfaisable, soit par $U(x) = 1$ pour tout x, soit par $U(x) = 0$ pour tout x.

(ii) $|C_1| = 2$ et $|C_2| = 3$, car sinon C_1 serait unitaire, et la formule serait toujours insatisfaisable. En effet, les fonctions étant des permutations, la formule $\forall x\ C_1$ serait équivalente soit à $\forall x\ Ux$, soit à $\forall x\ \neg Ux$. Ce qui contredirait $\forall x\ C_2$ par le point (i).

Par les points (i) et (ii), φ ne peut être que l'un des deux candidats φ_1 et φ_2, aux permutations près de f avec g, et de U avec $\neg U$:

$$\begin{array}{lllllllllll} \varphi_1(f,g): & \exists U\ \forall x & (Ux & \vee & Ufx) & \wedge & (\neg Ux & \vee & \neg Ufx & \vee & \neg Ugx) \\ \varphi_2(f,g): & \exists U\ \forall x & (Ugx & \vee & Ufx) & \wedge & (\neg Ugx & \vee & \neg Ufx & \vee & \neg Ux). \end{array}$$

La formule φ_1 n'est autre que $\varphi_{\min}$. Les formules φ_1 et φ_2 définissent essentiellement le même problème sur les structures bijectives (resp. sur les structures bijectives planaires) $\langle D, f, g\rangle$: Car en remplaçant x par $g^{-1}x$ dans la matrice de la formule φ_1, on obtient immédiatement l'équivalence :

$$\langle D, f, g\rangle \models \varphi_1(f, g) \iff \langle D, f', g'\rangle \models \varphi_2(f', g'),$$

où $f' = g^{-1}$ et $g' = fg^{-1}$. Cela a également du sens sur les structures bijectives planaires lorsque l'on remarque que le graphe $G(D, f, g)$ est planaire ssi le graphe $G(D, f', g')$ est lui-même planaire. ■

6.7 Equivalence parcimonieuse à SAT et NP-complétude minimale

Nous avons vu que notre réduction de SAT au problème minimal $\Pi_{\min}$ n'est pas parcimonieuse, alors que notre réduction de SAT à Π_{nand} l'est. On peut naturellement se demander s'il existe une formule minimale décrivant un problème parcimonieusement équivalent à SAT, et s'il est unique. Nous répondons partiellement à cette question par la proposition suivante :

Proposition 6.24 *Supposons que $P \neq NP$ et qu'il n'existe pas de réduction parcimonieuse de* SAT *à* $\Pi_{\min}$. *Alors, parmi les $\{f, g\}$-formules EMSO en CNF de la forme*

$$\exists U\ \forall x\ \psi(x)$$

(où ψ est sans quantificateur, sans égalité, sans composition de fonctions, et uniquement construite sur les atomes Ux, Ufx et Ugx), φ_{nand} est l'unique formule minimale (aux permutations près de x, fx, gx et de U, $\neg U$) qui définit un problème NP-complet sur structures à deux bijections et qui soit parcimonieusment équivalent à SAT. *Plus précisément, φ_{nand} a un nombre minimal de clauses ($= 3$), et une longueur minimale ($= 7$).*

On commence par prouver la minimalité de φ_{nand}.

Preuve (minimalité)

Minimalité du nombre de clauses $(= 3)$ Comme il a déjà été remarqué, toute formule EMSO de la forme requise et definissant un problème NP-complet avec seulement deux clauses a exactement une clause monotone négative et une clause monotone positive, parmi lesquelles on a au moins une clause de longueur ≥ 3 et aucune clause unitaire ; Donc l'autre clause est de longueur 2 ou 3. Il n'y a alors que deux formes possibles : Notre formule minimale $\varphi_{\min}$ (et ses variantes symétriques), et la formule φ_{nae} définie comme suit :

$$\begin{array}{ll} \varphi_{\text{nae}} : & \exists U\ \forall x \quad \psi_{\text{nae}}(x) \quad \text{, où } \psi_{\text{nae}} \text{ est la formule "not-all-equal"} \\ \psi_{\text{nae}} : & (Ux \vee Ufx \vee Ugx) \wedge (\neg Ux \vee \neg Ufx \vee \neg Ugx). \end{array}$$

Or, on sait que $\#\{U : (\mathcal{S}, U) \models \forall x\ \psi_{\text{nae}}(x)\}$ est *pair* car $\{\mathcal{S} : \mathcal{S} \models \varphi_{\text{nae}}\}$ est invariant par inversion de U et $\neg U$. Donc, aucune réduction de SAT au problème Π_{nae} (s'il en existe une) ne peut être parcimonieuse (avec la façon standard de compter le nombre de solutions).

Minimalité de la longueur $(= 7)$ C'est une conséquence directe du fait qu'il doit toujours y avoir deux clauses de longueur ≥ 2, et au moins une autre de longueur 3. ■

Enfin, voici la preuve de l'unicité de φ_{nand} :

Preuve (Unicité) Clairement, toute formule qui vérifie nos conditions de minimalité, i.e., qui a trois clauses et est de longueur 7, a exactement une 3-clause et deux 2-clauses. On remarque de plus que :

(*i*) Il y a au moins une clause monotone positive et au moins une clause monotone négative ;

(*ii*) Aucune 2-clause "ne subsume" (i.e., "n'est incluse dans") la 3-clause ;

(*iii*) Chaque 2-clause doit être en désaccord avec la 3-clause sur le signe de tous leur littéraux communs. Sinon, en notant la 3-clause $(\ell_1 \vee \ell_2 \vee \ell_3)$, soit la 2-clause est de la forme $(\ell_1 \vee \ell_2)$ et alors elle subsume la 3-clause, soit la 2-clause est de la forme $(\overline{\ell_1} \vee \ell_2)$ et alors une étape de résolution sur ℓ_1 induit la 2-clause $(\ell_2 \vee \ell_3)$ qui subsume à son tour la 3-clause, en contradiction avec le point (*ii*) ;

(*iv*) Les 2-clauses ont exactement un atome en commun : Elles en ont clairement au moins un puisqu'il n'y a que trois atomes disponibles. Maintenant, si elles en ont deux, elles sont en désaccord sur le signe de soit un, soit deux littéraux. Si nous avons $(\ell_1 \vee \ell_2) \wedge (\ell_1 \vee \overline{\ell_2})$, alors une étape de résolution sur ℓ_2 induit la 1-clause (ℓ_1). Et si nous avons $(\ell_1 \vee \ell_2) \wedge (\overline{\ell_1} \vee \overline{\ell_2})$, alors $\ell_1 \iff \overline{\ell_2}$, et la 3-clause se réduit soit à une 2-clause ou à "vrai" en remplaçant ℓ_1 par $\overline{\ell_2}$;

(*v*) La 3-clause doit être monotone. Sinon, par le point (*i*), les deux 2-clauses doivent être monotones et de signes opposés : Soit alors ε le signe majoritaire de la 3-clause. La 2-clause de signe ε ne peut être en désaccord sur le signe de chaque littéral de la 3-clause, car cette dernière n'a qu'un littéral de signe $\overline{\varepsilon}$. Ceci contredit le point (*iii*) ;

(*vi*) Les deux 2-clauses sont monotones, de même signe, et de signe opposé à celui de la 3-clause : C'est une conséquence directe des points (*iii*) et (*v*).

Clairement, les points (*iv*), (*v*) et (*vi*) mis ensemble ne laissent comme formules candidates que φ_{nand} et ses variantes symétriques. ■

Conclusion

Nous avons abordé dans cette thèse plusieurs problèmes liés de façon très étroite au problème de la Satisfaisabilité, i.e., des problèmes équivalents à Sat sous réductions vérifiant plusieurs critères :

- le temps linéaire, qui permet la préservation de la complexité des problèmes,
- la parcimonie, préservation du nombre de solutions entre problèmes, qui, en pratique, permet également de conserver la structure de l'espace des solutions des problèmes. Couplée au temps linéaire, elle permet ainsi de conserver la complexité de l'énumération des problèmes. Les réductions parcimonieuses de Sat permettent de prouver de façon unifiée des résultats de NP-complétude, #P-complétude, et DP-complétude.
- la préservation de la planarité des instances. La variante planaire d'un problème étant généralement de complexité plus basse que le cas général, les réductions à la fois linéaires et planaires permettent de conserver la complexité des problèmes planaires.

Les contributions de cette thèse sont à la fois combinatoires et logiques. Du point de vue combinatoire, nous avons amélioré des réductions de la littérature pour lesquelles au moins un des critères cités plus haut – linéarité, parcimonie, planarité – manquait.

Du point de vue de la parcimonie, nous avons unifié les preuves de NP-complétude et de #P-complétude de la 3-colorabilité – resp. de la 3-colorabilité planaire – en établissant l'équivalence de ce problème à celui de la Satisfaisabilité – resp. de la Satisfaisabilité planaire – sous réductions linéaires et *parcimonieuses*. Nous avons également montré comment planariser parcimonieusement une instance 3-Col en temps quadratique en exhibant un crossover parcimonieux pour Plan-3-Col. Ces réductions montrent également la DP-complétude de la 3-colorabilité unique et de la 3-colorabilité planaire unique sous réductions polynomiales aléatoires, des résultats nouveaux à notre connaissance. Seules des réductions faiblement parcimonieuses avec multiplication par un facteur exponentiel existaient à ce jour.

Du point de vue du temps linéaire et de la préservation de la planarité, nous avons montré l'équivalence de la satisfaisabilité planaire et de l'Hamiltonicité planaire sous réductions *linéaires* et parcimonieuses, et ce pour de nombreuses variantes de l'Hamiltonicité planaire : circuits ou cycles Hamiltoniens, chaînes ou chemins Hamiltoniens d'extrémités libres ou fixées, cycles ou chemins dans les graphes de degré maximal 3. Comme il semble a priori ne pas exister de réduction de l'Hamiltonicité générale à la satisfaisabilité générale qui soit de complexité sensiblement moindre que $O(|E|\log|V|)$, l'existence de réductions linéaires dans le plan est à souligner. Ceci montre que l'Hamiltonicité n'est plus un problème de connexité dans le plan, mais un simple problème de coloriage sous contraintes locales. Il suit qu'on peut appliquer à n'importe quelle variante de Plan-Hamilton la stratégie de diviser-pour-régner déjà connue pour les problèmes de coloriages locaux dans le plan et obtenir des algorithmes sous-exponentiels de même complexité $2^{O(\sqrt{n})}$.

Enfin, dans l'aspect logique de cette thèse, nous avons défini les classes LIN-LOCAL et LIN-PLAN-LOCAL, classes définies comme l'ensemble des problèmes linéairement réductibles au

problème Π_φ une formule EMSO φ exprimant une contrainte uniforme et locale sur des structures fonctionnelles (ou bijectives) et sur les structures fonctionnelles (ou bijectives) planaires. Ces classes capturent exactement les problèmes linéairement équivalents à la Satisfaisabilité et à la Satisfaisabilité planaire. Cette démarche nous a conduit à explorer les plus petites formules EMSO qui définissent un problème NP-complet. Nous avons montré qu'il existe une formule EMSO $\varphi_{\min}$, *minimale* sous plusieurs critères syntaxiques, définissant un problème $\Pi_{\min}$ sur structures fonctionnelles – resp. sur structures fonctionnelles planaires – qui est NP-complet et en fait linéairement équivalent à Sat – resp. Plan-Sat. De plus, la formule $\varphi_{\min}$ a été prouvée *unique*, à quelques symétries près, sur structures bijectives et sur les structures bijectives planaires. Nos réductions de Sat et Plan-Sat à $\Pi_{\min}$ ne sont pas parcimonieuses, mais restent cependant faiblement parcimonieuses. Dans un cadre syntaxique plus restreint, nous montrons que s'il n'existe pas de réduction parcimonieuse de Sat à $\Pi_{\min}$ sur structures bijectives et bijectives planaires, alors une variante φ_{nand} jouit des mêmes propriétés de minimalité et d'unicité pour la définition d'un problème Π_{nand} sur structures bijectives et bijectives planaires qui est (linéairement et) *parcimonieusement* équivalent à Sat.

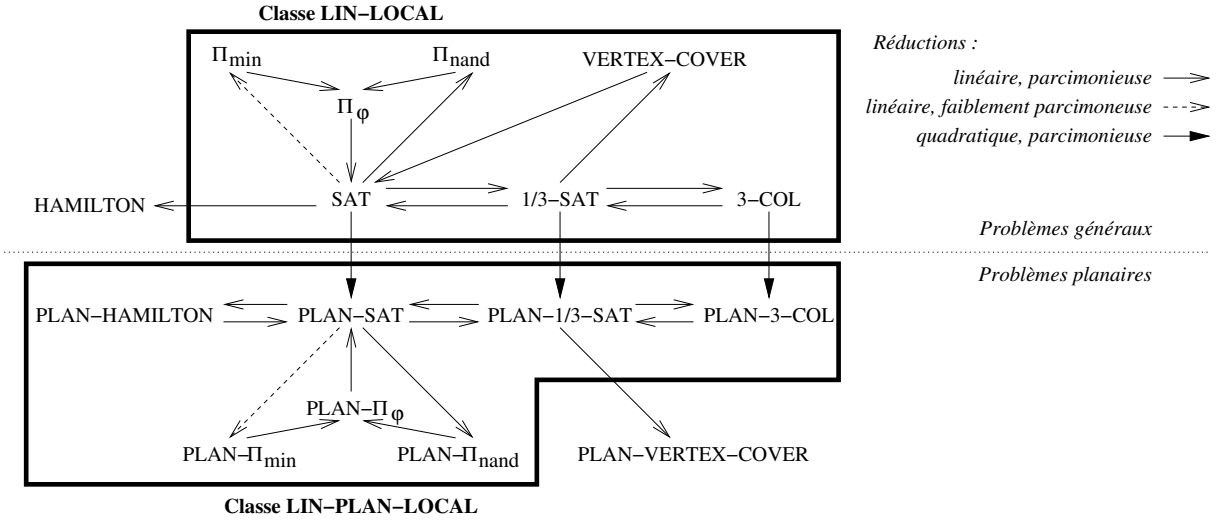


Fig. 1 – Synthèse des réductions présentées dans ce mémoire

Nous laissons un certain nombre de problèmes ouverts :

- En ce qui concerne nos réductions parcimonieuses à 3-Col et Plan-3-Col, nous n'avons pas essayé d'optimiser le degré des graphes de sortie. On peut se demander quel est le degré minimal pour lequel 3-Col et Plan-3-Col restent parcimonieusement équivalents à Sat et Plan-Sat.
- Des réductions linéaires de Plan-Vertex-Cover à Plan-Sat et de Hamilton à Sat existent-elles ? Bien que la réponse soit problablement négative dans les deux cas, le fait que Plan-Vertex-Cover et Plan-Sat ont le même schéma "diviser-pour-régner" pour leurs algorithmes sous-exponentiels nous pousse à penser que ces deux problèmes ont la même compléxité. Il est donc dommage de ne pas pouvoir établir leur équivalence linéaire.
- Un problème linéairement équivalent à Sat qui n'a pas été étudié dans ce mémoire est 2-Disjoint-Paths, le problème de l'existence de deux chemins disjoints à extrémités fixées dans un graphe orienté. La réduction originale par Fortune et Wyllie [32] de Sat à ce problème est non parcimonieuse, bien qu'elle puisse être rendue faiblement parcimonieuse.

Il semble qu'aucune réduction parcimonieuse entre ces deux problèmes ne soit possible, dans un sens comme dans l'autre. Il serait intéressant d'en avoir la preuve.

- Nos réductions linéaires entre la satisfaisabilité planaire et l'Hamiltonicité planaire sont tributaires des propriétés de la géométrie euclidienne dans le plan. En particulier, nos preuves ne tiennent plus pour des surfaces de genre supérieur à zéro comme le tore. Il serait intéressant de savoir si un lien linéaire existe entre l'Hamiltonicité torique et la satisfaisabilité torique ou même générale.
- Existe-t-il une réduction parcimonieuse de SAT au problème $\Pi_{\min}$? Nous conjecturons que non.
- La classe des problèmes linéairement réductibles à HAMILTON admet-t-elle une caractérisation logique comme la classe de SAT ? Une piste consiste à étendre les formules locales par l'insatisfaisabilité d'une certaine formule EMSO-Horn à 1 variable. Nous savons qu'une telle formule définit une sous-classe de NLIN et que cette classe contient HAMILTON. Nous ne savons cependant pas prouver la complétude de HAMILTON pour cette classe.

Bibliographie

[1] A. V. Aho, J. Hopcroft, and J. D. Ullmann. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[2] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I and II*. Springer, 1990.

[3] R. Barbanchon. Planar Hamiltonian problems and linear parsimonious reductions. Tech. report, Les Cahiers du GREYC 1, 2001. (postscript available at http ://www.info.unicaen.fr/~regis).

[4] R. Barbanchon. The problems Sat and Hamilton are equivalent under linear parsimonious reductions in the plane. Tech. report, Les Cahiers du GREYC 4, 2001. (postscript available at http ://www.info.unicaen.fr/~regis).

[5] R. Barbanchon. On unique 3-colorability and parsimonious reductions in the plane. *Theoretical Computer Science*, à paraître en 2004.

[6] R. Barbanchon and E. Grandjean. Local problems and linear time. Tech. report, Les Cahiers du GREYC 8, 2001. (postscript available at http ://www.info.unicaen.fr/~regis).

[7] R. Barbanchon and E. Grandjean. Local problems, planar local problems and linear time. In *Computer Science Logic*, volume 2471 of *Lecture Notes in Computer Science*, pages 397–411. Springer, 2002.

[8] R. Barbanchon and E. Grandjean. The minimal logically-defined NP-complete problem. *Symposium on Theoretical Aspect of Computer Science*, à paraître en 2004.

[9] P. Beame. A general sequential time-space tradeoff for finding unique elements. *SIAM Journal on Computing*, 20(2) :270–277, 1991.

[10] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of Computer and System Sciences*, 13(3) :335–379, 1976.

[11] A. Borodin and S. A. Cook. A time-space tradeoff for sorting on a general sequential model of computation. *Society for Industrial and Applied Mathematic Journal on Computing*, 11(2) :287–297, 1982.

[12] G. Chartrand and L. Lesniak. *Graphs and Digraphs*. Wadsworth and Brooks, California, 1986.

[13] S. A. Cook. An overview of computational complexity. *Communications of the ACM*, 26(6) :400–408, June 1983.

[14] S. A. Cook. Short propositional formulas represent nondeterministic computations. *Information Processing Letters*, 26(5) :269–270, 1988.

[15] Stephen A. Cook. The complexity of theorem-proving procedures. In *ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[16] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. Mc Graw Hill, 1991.

[17] B. Courcelle. The monadic second-order logic of graphs vi : On several representations of graphs as relational structures. *Discrete Applied Mathematics*, 54 :117–149, 1994.

[18] B. Courcelle. On the expression of graph properties in some fragments of monadic second-order logic. In N. Immerman and P. Kolaitis, editors, *Descriptive Complexity and Finite Models*, pages 33–62. American Mathematical Society, 1997.

[19] N. Creignou. *Temps linéaire et problèmes NP-complets*. 1993. (Thèse de doctorat, Université de Caen, France).

[20] N. Creignou. The class of problems that are linearly equivalent to Satisfiability or a uniform method for proving NP-completeness. *Theoretical Computer Science*, 145(1–2) :111–145, 1995.

[21] N. Creignou and J-J. Hébrard. On generating all solutions of generalized satisfiability problems. *Informatique Théorique et Applications*, 31(6) :499–511, 1997.

[22] N. Creignou and M. Hermann. On #P-completeness of some counting problems. Technical report, INRIA, 1993.

[23] N. Creignou and M. Hermann. Complexity of generalized satisfiability counting problems. *Information and Computation*, 125(2) :1–12, 1996.

[24] N. Creignou, S. Khanna, and M. Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems*. SIAM Press, Philadelphia, USA, 2001.

[25] A. K. Dewdney. Linear time transformations between combinatorial problems. *International Journal of Computer Mathematics*, 11 :91–110, 1982.

[26] H. D. Ebbinghaus and J. Flum. *Finite Model Theory*. Perspectives in Mathematical Logic. Springer, 1995.

[27] T. Eiter, G. Gottlob, and Y. Gurevich. Existential second order logic over strings. *Journal of the ACM*, 41(1) :77–131, 2000.

[28] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. *Complexity and Computation*, 7 :43–73, 1974.

[29] R. Fagin. Monadic generalized spectra. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 21 :89–96, 1975.

[30] L. Fortnow. Time-space tradeoffs for satisfiability. *Journal of Computer and System Sciences*, 60(2) :337–353, 2000.

[31] L. Fortnow and D. van Melkebeek. Time-space tradeoffs for nondeterministic computation. In *Proceedings of the IEEE Conference on Computational Complexity*, pages 2–13, 2000.

[32] S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2) :111–121, 1980.

[33] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman and Co., 1979.

[34] M. R. Garey, D. S. Johnson, and R. E. Tarjan. The planar hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing*, 5(4) :704–714, 1976.

[35] G. Gottlob, P. G. Kolaitis, and T. Schwentick. Existential second-order logic over graphs : Charting the tractability frontier. *Foundations of Computer Science*, pages 664–674, 2000.

[36] E. Grädel. On the notion of linear time computability. *International Journal of Foundations of Computer Science*, 1, 1990.

[37] E. Grädel. Capturing complexity classes by fragments of second order logic. In *Structure in Complexity Theory Conference*, pages 341–352. IEEE Computer Society Press, 1991.

[38] E. Grädel. The expressive power of second order horn logic. In *Symposium on Theoretical Aspects of Computer Science*, volume 480 of *Lecture Notes in Computer Science*, pages 466–477. Springer, 1991.

[39] E. Grandjean. The spectra of first-order sentences and computational complexity. *SIAM Journal on Computing*, 13(2) :356–373, 1984.

[40] E. Grandjean. Universal quantifiers and time complexity of random access machines. *Mathematical Systems Theory*, 18 :171–187, 1985.

[41] E. Grandjean. First-order spectra with one variable. *Journal of Computer and System Sciences*, 40(2) :136–153, 1990.

[42] E. Grandjean. A nontrivial lower bound for an NP problem on automata. *SIAM Journal on Computing*, 19 :438–451, 1990.

[43] E. Grandjean. Invariance properties of RAMs and linear time. *Computational Complexity*, 4 :62–106, 1994.

[44] E. Grandjean. Linear time algorithms and NP-complete problems. *SIAM Journal on Computing*, 23(3) :573–597, 1994.

[45] E. Grandjean. Sorting, linear time and the satisfiability problem. *Annals of Mathematics and Artificial Intelligence*, 16 :183–236, 1996.

[46] E. Grandjean and F. Olive. Monadic logical definability of nondeterministic linear time. *Computational Complexity*, 7(1) :54–97, 1998.

[47] E. Grandjean and F. Olive. Graph properties checkable in linear time in the number of vertices. *Journal of Computer and System Sciences*, 2003.

[48] E. Grandjean and T. Schwentick. Machine-independent characterizations and complete problems for deterministic linear time. *SIAM Journal on Computing*, 32(1) :196–230, 2002.

[49] Y. Gurevich and S. Shelah. Nearly linear time. In *Proceedings of the Symposium on Logical Foundations of Computer Science*, volume 363 of *Lecture Notes in Computer Science*, pages 108–118. Springer, 1989.

[50] Y. Gurevich and S. Shelah. Nondeterministic linear-time tasks may require substantially nonlinear deterministic time in the case of sublinear work space. *Journal of the ACM*, 37(3) :674–687, 1990.

[51] M. R. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *Journal of Computer and System Sciences*, 55(1) :3–23, 1997.

[52] J. E. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM Journal on Computing*, 2(3) :135–158, 1973.

[53] John E. Hopcroft and Robert Endre Tarjan. Efficient planarity testing. *Journal of the ACM*, 21(4) :549–568, 1974.

[54] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, and R. E. Stearns. The complexity of planar counting problems. *SIAM Journal on Computing*, 27(4) :1142–1167, 1998.

[55] H. B. Hunt III, R. E. Stearns, and M. V. Marathe. On the complexity of planar graph and hypergraph coloring. (communication de Marathe), 2001.

[56] N. Immerman. *Descriptive Complexity*. Graduate Texts in Computer Science. Springer, 1998.

[57] M. Jerrum. Counting trees in a graph is #P-complete. *Information Processing Letters*, 51(3) :111–116, 1994.

[58] D. S. Johnson and C. H. Papadimitriou. *Computional Complexity, in The Traveling Salesman Problem.* Wiley and sons ltd, 1985.

[59] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[60] D. K. Kozen. *The Design and Analysis of Algorithms.* Springer-Verlag, 1991.

[61] C. L. Lautemann and B. Weinzinger. Monadic-NLIN and quantifier-free reductions. In *Computer Science Logic*, volume 1683 of *Lecture Notes in Computer Science*, pages 322–337, 1999.

[62] D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2) :329–343, 1982.

[63] N. Linial. Hard enumeration problems in geometry and combinatorics. *SIAM Journal on Algebraic Discrete Methods*, 7 :331–335, 1986.

[64] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36 :177–189, 1979.

[65] R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3) :615–627, 1980.

[66] R. J. Lipton and A. Viglas. On the complexity of SAT. In *IEEE Symposium on Foundations of Computer Science*, pages 459–464, 1999.

[67] K. Mehlhorn and P. Mutzel. On the embedding phase of the Hopcroft and Tarjan planarity testing algorithm. *Algorithmica*, 16(2) :233–242, 1996.

[68] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, Reading, MA, 1994.

[69] W. J. Paul, N. Pippenger, E. Szeméredi, and W. T. Trotter. On determinism versus nondeterminism and related problems. In *24th Annual Symposium on Foundations of Computer Science*, pages 429–438. IEEE Computer Society Press, 1982.

[70] J. Plesník. The NP-completeness of the hamiltonian cycle problem in planar digraphs with degree bound two. *Information Processing Letters*, 8(4) :199–201, 1979.

[71] J. S. Provan. The complexity of reliability computations in planar and acyclic graphs. *SIAM Journal on Computing*, 15(3) :694–702, 1986.

[72] J. S. Provan and M. O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4) :777–788, 1983.

[73] S. Ranaivoson. Nontrivial lower bounds for some NP-complete problems on directed graphs. In *Computer Science Logic*, volume 533 of *Lecture Notes in Computer Science*, pages 318–339, 1991.

[74] S. S. Ravi and H. B. Hunt, III. An application of the planar separator theorem to counting problems. *Information Processing Letters*, 25(5) :317–321, 1987.

[75] J. M. Robson. Subexponential algorithms for some NP-complete problems. Manuscript, 1985.

[76] T. J. Schaefer. The complexity of satisfiability problems. In *ACM Symposium on Theory of Computing*, pages 216–226, 1978.

[77] C. P. Schnorr. Satisfiability is quasilinear complete in NQL. *Journal of the ACM*, 25(1) :136–145, 1978.

[78] T. Schwentick. On winning Ehrenfeucht games and Monadic NP. *Annals of Pure and Applied Logic*, 79(1) :61–92, 1996.

[79] T. Schwentick. Algebraic and logical characterizations of deterministic linear time classes. In *14th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1200 of *Lecture Notes in Computer Science*, pages 463–474. Springer, 1997.

[80] T. Schwentick. Descriptive complexity, lower bounds and linear time. In *Computer Science Logic*, volume 1584 of *Lecture Notes in Computer Science*, pages 9–28. Springer, 1998.

[81] Y. Shiloach. A polynomial solution to the undirected two paths problem. *Journal of the ACM*, 27(3) :445–456, 1980.

[82] R. E. Stearns and H. B. Hunt III. Power indices and easier hard problems. *Mathematical Systems Theory*, 23(4) :209–225, 1990.

[83] W. T. Tutte. On Hamilton circuits. *Journal of London Mathematical Society*, 21 :98–101, 1946.

[84] S. P. Vadhan. the complexity of counting in sparse, regular, and planar graphs. *SIAM Journal on Computing*, 31(2) :398–427, 2001.

[85] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2) :189–201, 1979.

[86] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3) :410–421, 1979.

[87] L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47(1) :85–93, 1986.

[88] H. Venkateswaran. Properties that characterize LOGCFL. In *Proceedings of the ACM 19th annual Symposium on Theory of Computing*, pages 141–150. ACM Press, 1987.

Résumé

Réductions fines entre problèmes NP-complets
Linéarité, planarité, parcimonie et minimalité logique

Nous étudions les problèmes combinatoires NP-complets autour de la Satisfaisabilité (Sat) : 3-Colorabilité (3-Col), Hamiltonicité (Hamilton), etc., et les réductions fines entre ces problèmes et leurs versions planaires. Nous cherchons à préserver la structure des solutions (parcimonie), la complexité des problèmes (linéarité), et la géométrie des instances (planarité). Nous exhibons une réduction quadratique et parcimonieuse de 3-Col à Plan-3-Col et une réduction linéaire, planaire, et parcimonieuse de Sat à 3-Col, unifiant et raffinant les preuves de NP-complétude et de #P-complétude pour 3-Col, Plan-3-Col, #3-Col et #Plan-3-Col, et montrant la DP-complétude de Unique-3-Col et Unique-Plan-3-Col. Nous établissons l'équivalence linéaire et parcimonieuse de Plan-Sat et de Plan-Hamilton, un lien peu probable entre Sat et Hamilton. Nous exhibons enfin une formule Existentielle Monadique du Second Ordre prouvée minimale et unique, définissant un problème NP-complet sur structures bijectives.

Mots-clés : *Problèmes combinatoires, graphes, satisfaisabilité, 3-colorabilité, Hamiltonicité ; Complexité algorithmique, non-déterminisme, temps polynomial, comptage, solutions uniques ; Complexité descriptive, logique existentielle monadique du second ordre ; Réductions, linéarité, parcimonie, planarité.*

Abstract

Tight reductions between NP-complete problems
Linearity, planarity, parsimony, and logical minimality

We study the combinatorial NP-complete problems near to Satisfiability (Sat) : 3-Colorability (3-Col), Hamiltonicity (Hamilton), etc., and the tight reductions between these problems and their planar versions. We aim at preserving the structure of the solutions (parcimony), the complexity of the problems (linearity), and the geometry of the instances (planarity). We present a reduction from 3-Col to Plan-3-Col that is quadratic and parsimonious, and a reduction from 3-Col to Sat that is linear, parsimonious and planar, thus unifying and refining the proofs of NP-completeness and #P-completeness for 3-Col, Plan-3-Col, #3-Col et #Plan-3-Col, and showing the DP-completeness of Unique-3-Col and Unique-Plan-3-Col. We establish the linear and parsimonious equivalence of Plan-Sat and Plan-Hamilton, an unlikely link between Sat and Hamilton. Finally, we present an Existential Monadic Second Order formula, proved minimal and unique, that defines an NP-complete problem over bijective structures.

Keywords : *Combinatorial problems, graphs, satisfiability, 3-colorability, Hamiltonicity ; Computational complexity, nondeterminism, polynomial time, counting, unique solutions ; Descriptive complexity, existential monadic second order logic ; Reductions, linearity, parsimony, planarity.*

Spécialité : Informatique

GREYC – CNRS UMR 6072
Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen
Université de Caen campus II – Boulevard du Maréchal Juin 14032 Caen Cedex