

Étude de Systèmes de Contraintes pour le Raisonnement Qualitatif Temporel et Spatial

THÈSE

vendredi 3 décembre 2010

en vue de l'obtention du

Doctorat de l'Université d'Artois
(Spécialité Informatique)

par

Dominique D'Almeida

Composition du jury

<i>Président :</i>	Pierre Marquis	Université d'Artois
<i>Rapporteurs :</i>	Belaïd Benhamou Maroua Bouzid	Université de Provence Université de Caen - Basse Normandie
<i>Examineurs :</i>	Frédéric Saubion Christine Solnon	Université d'Angers Université de Lyon I
<i>Directeur de Thèse :</i>	Lakhdar Saïs	Université d'Artois
<i>Co-Directeurs :</i>	Jean-François Condotta Christophe Lecoutre	Université d'Artois Université d'Artois

Remerciements

J'adresse de grands remerciements à Madame Maroua Bouzid et Monsieur Belaïd Benhamou, pour avoir accepté d'être rapporteurs de cette thèse, ainsi que pour leurs remarques et conseils afin d'améliorer ce rapport. Je remercie également Madame Christine Solnon, Monsieur Frédéric Saubion et Monsieur Pierre Marquis pour m'avoir fait l'honneur de faire partie du jury et d'examiner ce travail.

Je remercie mes co-directeurs de thèse Jean-François Condotta et Christophe Lecoutre, ainsi que mon directeur de thèse Lakhdar Saïs pour m'avoir fait découvrir durant mes années de thèses les domaines dont ils sont spécialistes. Par leur encadrement, ils m'ont donné leur vision de la recherche et de leurs thématiques afin de forger ma propre vision. Ils m'ont transmis le goût de la recherche et je les en remercie fortement.

Merci à tous les membres du CRIL pour leur convivialité et la bonne humeur qu'ils véhiculent. Ils m'ont permis de passer des années très agréables et d'apprécier d'avantage ce travail passionnant. Un grand merci à mes collègues et amis (ex-)doctorants avec qui j'ai passé de très bons moments, en particulier au « MIT »-*crew* avec qui j'ai eu le plaisir de partager la majeure partie de ma thèse.

Je remercie infiniment (et ce n'est pas encore assez) mes parents à qui je dois énormément, en faisant de moi qui je suis et en m'ayant soutenu durant mes études jusqu'à l'écriture de cette thèse. Merci également à mon frère et à ma sœur pour leur présence et leur soutien.

Enfin, je remercie énormément mes chéries d'amour que j'aime de tout mon cœur très fort pour toujours, ma femme et ma fille, pour m'avoir soutenu durant cette thèse. Merci pour votre patience et merci d'avoir supporté mes sautes d'humeurs chroniques.

À ma femme, à ma fille et à mes parents.

Table des matières

Introduction générale	1
------------------------------	----------

État de l'art

Chapitre 1 Les réseaux de contraintes discrets	7
1.1 Introduction	7
1.2 Les Réseaux de Contraintes Discrets	8
1.2.1 Définition générale et terminologie	8
1.2.2 Instanciation et cohérence	11
1.3 Algorithmes pour l'inférence et la prise de décision	16
1.3.1 Algorithmes de recherche	16
1.3.2 Cohérence locale	19
1.3.3 Choix heuristiques	28
1.4 Conclusion	32
Chapitre 2 Formalismes qualitatifs et réseaux de contraintes qualitatives	35
2.1 Introduction	35
2.2 Les formalismes qualitatifs	37
2.2.1 Définition générale	37
2.2.2 Opérations sur les relations	38
2.2.3 Quelques formalismes qualitatifs	39
2.3 Les réseaux de contraintes qualitatives	45
2.3.1 Définition générale	45

2.3.2	Problème de satisfiabilité de contraintes qualitatives	47
2.3.3	Cohérence locale	48
2.3.4	Algorithme de recherche et techniques d'inférence	50
2.3.5	Les applications	54
2.4	Quelques fragments traitables de l'algèbre des intervalles	54
2.4.1	Relations pointisables	54
2.4.2	Relations ORD-Horn	56
2.4.3	Relations convexes	58
2.4.4	Relations préconvexes	60
2.5	Conclusion	63
Chapitre 3 Formules propositionnelles		64
3.1	Introduction	64
3.2	La logique propositionnelle	65
3.2.1	Syntaxe : le langage propositionnel	65
3.2.2	Sémantique : la satisfiabilité	67
3.3	Techniques algorithmiques	68
3.3.1	Algorithmes de recherche et propagation	68
3.3.2	Simplification de formules propositionnelles	69
3.3.3	Outils algorithmiques modernes	70
3.3.4	Classes traitables	71
3.4	Conclusion	72

Définir et exploiter les transformations

Chapitre 4 Du problème CSP au problème SAT		75
4.1	Introduction	75
4.2	De l'instanciation à l'interprétation	76
4.3	Modélisation des contraintes : les conflits	77
4.3.1	Décrire les tuples conflits	77
4.3.2	Vers de nouvelles approches exploitant les transformations	78
4.4	Modélisation des contraintes : les supports	82

4.4.1	Supports et Cohérence d'Arcs	82
4.4.2	Vers de nouvelles modélisations de cohérence locale	83
4.5	Conclusion	84
Chapitre 5 Du problème CSP qualitatif au problème SAT		85
5.1	Introduction	85
5.2	Identification des variables propositionnelles	86
5.2.1	Variables propositionnelles	86
5.2.2	Du scénario à l'interprétation	86
5.3	Description des contraintes en logique propositionnelle	87
5.3.1	Utilisation des conflits	88
5.3.2	Utilisation des supports	90
5.4	Conclusion	91
Chapitre 6 Du problème CSP qualitatif au problème SAT en exploitant les classes traitables		93
6.1	Introduction	93
6.2	Transformation	94
6.2.1	Exploiter le treillis conceptuel	94
6.2.2	Définition des variables propositionnelles	95
6.2.3	Des propriétés du treillis conceptuel aux formules CNF	96
6.2.4	Découpage et transformation d'un RCQ	97
6.2.5	Dimension et complexité	99
6.3	Propriétés et complétude	100
6.3.1	Transformation inverse	100
6.3.2	Transformation correcte et complète	101
6.4	Conclusion	103
Chapitre 7 Du problème CSP qualitatif au problème CSP		105
7.1	Introduction	105
7.2	Définition de la transformation	106
7.3	Propriétés liées à la transformation	107
7.3.1	Transformation inverse	107
7.3.2	Complétude	107
7.3.3	Cohérence locale	108
7.4	Conclusion	110

Chapitre 8 Relaxation de contraintes qualitatives	113
8.1 Introduction	113
8.2 Relaxation	114
8.2.1 Transformation incomplète	114
8.2.2 Transformation inverse et complétion de l'approche	116
8.2.3 Exploiter les relations traitables	118
8.3 Expérimentations	119
8.4 Conclusion	122
Chapitre 9 ϕ-Substituabilité	125
9.1 Introduction	125
9.2 Substituabilité basée sur la cohérence d'arcs de singletons	126
9.2.1 Définition et étude de la substituabilité au voisinage	126
9.2.2 Substituabilité au prétraitement	127
9.2.3 Recherche dynamique de la substituabilité	130
9.3 Expérimentations	133
9.3.1 Intégration statique	133
9.3.2 Intégration dynamique	136
9.4 Conclusion	138
Conclusion générale	141
Annexe A Algèbre des Intervalles	145
A.1 Table de composition	145
A.2 Algèbre des intervalles affaiblie	146
Annexe B Des RCD aux formules CNF	148
B.1 Transformation en exploitant la cohérence d'arcs	148
B.2 Vers de nouvelles modélisations de cohérence locale	149
Bibliographie	151

Introduction générale

La vérification de matériels et de logiciels, la bio-informatique, la planification temporelle et spatiale, l'allocation de ressources, l'ordonnancement de tâches, la gestion d'emploi du temps, les systèmes d'information géographique, la compréhension du langage naturel, sont des domaines d'applications qui soulèvent de nombreux problèmes pouvant être représentés par un ensemble de contraintes. De nombreuses études tant théoriques que pratiques portent sur la modélisation et la résolution de ce type de problèmes. En particulier, décider de la cohérence d'un problème, à savoir prouver l'existence d'une solution, est une question centrale en intelligence artificielle. Dans le cas général, la résolution nécessite un temps considérable, au-delà de l'échelle de temps humaine. La modélisation mathématique apporte un cadre théorique solide pour l'étude de ces problèmes, et l'utilisation de l'outil informatique facilite la résolution. De part la diversité des problèmes à traiter, de nombreux modèles de représentation ont été proposés. La simplicité de modélisation, l'expressivité, la compacité de la représentation, mais également la puissance des outils mathématiques et algorithmiques, sont des critères très importants dans le choix de la représentation.

Parmi les domaines d'applications précédemment cités, nombreux sont ceux qui requièrent la manipulation de connaissances de nature temporelle et spatiale. Certaines modélisations proposent une quantification du temps et de l'espace, d'autres s'attachent à l'aspect qualitatif en décrivant les relations entre les entités. La manipulation de connaissances de nature temporelle par un raisonnement qualitatif prend son essor avec la définition de l'algèbre des intervalles [ALLEN 1983]. Ce formalisme qualitatif permet la représentation d'intervalles de temps sur une droite dense et décrit les contraintes entre deux intervalles par leurs positions relatives. Allen propose une représentation sous la forme de *Réseaux de Contraintes Qualitatives* (RCQ) permettant une modélisation naturelle et compacte des connaissances temporelles. Durant ces dernières décennies, ce cadre a été fortement exploité et étendu à d'autres formalismes qualitatifs pour le temps et l'espace. Les techniques de résolution des RCQ sont inspirées d'autres modèles de représentation de problèmes sous contraintes : les réseaux de contraintes discrets (RCD) et les formules propositionnelles sous forme normale conjonctive (CNF). Cependant, l'étude des propriétés des formalismes qualitatifs ont conduit au développement d'outils algorithmiques spécifiques. Ainsi, les techniques d'inférences s'appuient sur un affaiblissement de la composition mathématique, et les algorithmes de recherche sont facilités par la caractérisation de fragments traitables de certains formalismes qualitatifs. Chaque modèle de représentation donne une vision différente dans la façon d'appréhender un problème. Cela se traduit par une grande diversité dans le développement d'outils algorithmiques et dans la direction des études théoriques. Cependant, le contraste entre les résultats pratiques pour la résolution de problèmes dans les différents cadres est fortement marqué. Pour répondre à un problème, il peut être plus efficace d'exploiter les outils algorithmiques d'un cadre alors qu'une autre modélisation semble plus naturelle. Ceci peut s'expliquer par la difficulté à manipuler les informations d'une structure compacte ou la facilité à adapter des informations en logique propositionnelle. Chaque formalisme devient une source d'inspiration pour les autres. *L'objectif de notre étude est d'approfondir les connexions entre les différents systèmes de contraintes afin d'apporter de nouveaux outils algorithmiques pour le raisonnement qualitatif temporel et spatial.*

Pour qu'un formalisme bénéficie des études menées dans d'autres domaines, une direction peut être les liens entre les représentations par le biais d'opérations de *transformation*. Une transformation est un processus algorithmique permettant de passer d'une représentation à une autre, voire de lier les problèmes de décisions. Cette approche est motivée par des intérêts tant pratiques que théoriques. Dans le cas général, tout raisonnement est fait à partir de la représentation d'un problème, dont la nature et le sens ne sont pas connus. Une analyse structurelle peut conduire à choisir un autre formalisme afin de bénéficier d'outils algorithmiques plus adaptés et simplifier la résolution par une transformation. D'autres approches consistent à s'inspirer ou à adapter les outils algorithmiques de chaque cadre. Les visions différentes d'un même problème peuvent amener à penser différemment et conduire à améliorer les propriétés théoriques ou les performances algorithmiques. Du point de vue théorique, les transformations peuvent aider à étudier la structure d'un problème et la complexité des algorithmes nécessaires à sa résolution. L'exploitation des classes traitables dans le but de faciliter les algorithmes de recherche est un aspect très étudié. Mettre en évidence ces classes particulières par les transformations et développer des approches pour les identifier peut contribuer à caractériser de nouvelles classes traitables dans les différents domaines. En clair, exploiter les transformations permet de voir un même problème sous différents angles afin d'y apporter une solution adaptée.

Notre première contribution est une étude d'une transformation des RCQ vers les formules CNF. Cette approche nous permet d'exploiter les outils algorithmiques d'un des formalismes les plus performants afin de répondre aux applications sur le temps et l'espace. Pour la définir, nous nous appuyons sur deux approches existantes, définies sur un schéma commun, pour transformer un RCD vers une formule CNF [WALSH 2000, GENT 2002]. La première étape passe par une description structurelle du RCD en associant chaque valeur du domaine de chaque variable du RCD à une variable propositionnelle de la formule CNF. Ceci a pour objectif de construire une interprétation complète de la formule CNF par un algorithme de recherche, pouvant être associée à une instantiation complète du RCD par la transformation. Ainsi, si l'interprétation complète est un modèle de la formule CNF alors il existe une solution du RCD. Pour vérifier cette condition, il est nécessaire que l'interprétation satisfasse les contraintes du problème. L'une des approches représente, dans la formule CNF, ce qui n'est pas autorisé par l'interprétation : les contraintes en conflits avec le résultat attendu. L'autre approche modélise une propriété de cohérence locale qu'est la cohérence d'arcs. Puisque la cohérence d'arcs permet de vérifier si une instantiation complète dans un RCD est une solution, transformer cette propriété permet de vérifier la cohérence de l'interprétation complète dans la formule CNF. En nous appuyant sur un raisonnement similaire, nous définissons une transformation permettant de faire la correspondance entre interprétation complète d'une formule CNF et scénario d'un RCQ. Nous proposons également deux façons de représenter les contraintes en décrivant ce que l'interprétation ne doit pas vérifier d'une part, et en décrivant la propriété de composition faible des RCQ permettant de vérifier la cohérence d'un scénario d'autre part. Cependant, il existe des formalismes qualitatifs pour lesquels la méthode de fermeture par composition faible ne permet pas de vérifier la cohérence d'un scénario. Par conséquent, la transformation n'est pas complète pour tout formalisme qualitatif, mais elle l'est pour une grande majorité d'entre eux.

Pour notre deuxième contribution, nous proposons d'exploiter une technique algorithmique efficace dans le domaine des RCQ, afin de définir une transformation vers les formules CNF [CONDOTTA & D'ALMEIDA 2007]. Certains formalismes qualitatifs bénéficient de propriétés mathématiques intéressantes, permettant de mettre en avant des classes de problèmes traitables. Ces problèmes sont mis en évidence par les contraintes particulières qui les décrivent : des relations traitables. Les algorithmes de recherche dans le cadre des RCQ mettent à profit ces relations en découpant les contraintes du RCQ en relation traitables. À chaque contrainte n'est plus associée une seule relation atomique, mais un ensemble de relations atomiques. Nous exploitons ce procédé afin de définir une transformation vers les formules CNF en utilisant ce découpage en relations traitables. Cette approche originale permet de conduire l'algo-

rythme de recherche des formules propositionnelles à se comporter de la même façon que les algorithmes de recherches des RCQ, tout en conservant les techniques d'inférence efficace du cadre propositionnel.

Notre troisième contribution est une transformation des RCQ vers les RCD [CONDOTTA *et al.* 2006a], une fois encore pour exploiter les techniques algorithmiques du cadre des RCD afin de répondre à des problèmes qualitatifs, mais également pour faire le lien entre les différentes techniques de cohérence locale. Les informations temporelles et spatiales sont décrites dans un formalisme qualitatif, permettant la description des entités temporelles et spatiales ainsi que les contraintes entre celles-ci. Dans notre transformation, chaque contrainte du RCQ sera liée à une variable du RCD et les valeurs du domaine de chaque variable du RCD coïncideront avec la relation décrivant la contrainte. Pour contraindre le RCD, nous garantissons la propriété de composition faible à l'aide d'une table de composition du formalisme qualitatif. Nous montrons que la méthode de fermeture par composition faible des RCQ est équivalente à la cohérence d'arcs pour les RCD. Puisque la cohérence d'arcs permet de vérifier la cohérence d'un scénario, nous montrons qu'il est possible de répondre à la cohérence d'un RCQ par la cohérence d'un RCD lorsque la méthode de fermeture par composition faible permet de vérifier la cohérence d'un scénario.

Notre quatrième contribution traite de la relaxation des contraintes qualitatives [CONDOTTA *et al.* 2007a]. « Relâcher une contrainte » signifie l'assouplir en autorisant plus qu'initialement. Elle peut être utilisée pour restaurer la cohérence d'un problème, mais également pour faciliter la preuve de l'incohérence. Cependant, il existe une contrainte des RCQ qui ne peut pas être relâchée localement : la propriété de composition faible. La transformation vers les RCD offre un outil de manipulation pour la relaxation des RCQ. En particulier, cette approche nous permet de relâcher les contraintes des RCQ les plus permissives : les contraintes universelles. Dans un premier temps, nous proposons une méthode incomplète pour le problème de la cohérence des RCQ, en relaxant la propriété de composition faible sur les contraintes universelles d'un RCQ en exploitant la transformation vers les RCD. Nous étendons ensuite ce cadre à une méthode complète en utilisant les classes traitables de certains formalismes qualitatifs. À partir de chaque solution du RCQ transformé et relaxé, nous pouvons construire un RCQ en réinstaurant la propriété de composition faible afin de vérifier sa cohérence. En la comparant avec une approche classique, notre approche montre des résultats intéressants sur des problèmes difficiles pour la preuve de l'incohérence. Cependant, l'approche complète montre quelques lacunes et celle-ci nécessite d'adapter les algorithmes existants.

Enfin, la dernière contribution présentée sort du cadre des transformations, mais a été inspirée par l'étude de l'adaptation des classes traitables des formalismes qualitatifs aux RCD. Dans le cadre discret, une analyse sémantique des contraintes peut mettre en avant la propriété particulière de certaines valeurs : la substituabilité [D'ALMEIDA & SAÏS 2009]. Une valeur est substituable par une seconde si et seulement si, pour toute solution dans laquelle cette valeur apparaît, il est possible de construire une autre solution en la remplaçant par la seconde. Montrer la substituabilité de deux valeurs peut faciliter la résolution du problème, mais les mettre en évidence est très difficile. En nous appuyant sur un cadre classique utilisé dans les algorithmes de propagations de contraintes et dans les algorithmes de recherche, nous définissons alors une forme de substituabilité locale. Nous proposons tout d'abord une détection statique, intégrée à un prétraitement de cohérence d'arcs de singletons, afin de faciliter le problème. Bien que l'approche puisse être coûteuse en temps, elle n'en reste pas moins efficace puisqu'elle permet de prouver la cohérence de certains problèmes avant la phase recherche, contrairement à un prétraitement classique. Une détection dynamique est ensuite intégrée dans les algorithmes de recherche et malgré une vérification systématique durant la recherche, les résultats se montrent plus intéressants qu'un algorithme de recherche classique.

Nos travaux sont présentés comme suit. Dans une première partie, nous décrivons les différents domaines étudiés. Nous définissons les RCD et la terminologie associée, puis nous présentons les techniques de recherche et d'inférence permettant de répondre au problème de la cohérence des RCD. Nous

définissons ensuite les formalismes qualitatifs et les RCQ, avant d'entrevoir les techniques de résolution, très proches du cadre des RCD. Nous proposons également un aperçu des différentes classes traitables de l'algèbre des intervalles, que nous exploiterons durant nos travaux. Nous présentons succinctement le langage de la logique propositionnelle et le problème SAT, en évoquant également les approches utiles à la résolution de celui-ci. Dans une seconde partie, nous développons nos contributions. Nous proposons dans un premier temps une étude des transformations existantes des RCD vers les formules CNF, puis nous définissons notre transformation naturelle des RCQ vers les formules propositionnelles. Nous exploitons ensuite les classes traitables pour proposer une nouvelle transformation vers la logique propositionnelle, en nous appuyant sur une décomposition des contraintes du RCQ. Nous définissons également une transformation des RCQ vers les RCD, afin d'étudier les mécanismes de propagations des contraintes. Puis nous exploitons ce cadre, en caractérisant une nouvelle forme de relaxation des contraintes dans le cadre des RCQ. Enfin, nous définissons une nouvelle forme de substituabilité locale, inspirée d'une intuition en manipulant les transformations.

État de l'art

Chapitre 1

Les réseaux de contraintes discrets

Sommaire

1.1	Introduction	7
1.2	Les Réseaux de Contraintes Discrets	8
1.2.1	Définition générale et terminologie	8
1.2.2	Instanciation et cohérence	11
1.3	Algorithmes pour l'inférence et la prise de décision	16
1.3.1	Algorithmes de recherche	16
1.3.2	Cohérence locale	19
1.3.3	Choix heuristiques	28
1.4	Conclusion	32

1.1 Introduction

L'une des principales motivations d'une modélisation mathématiques et informatique de problèmes ou de connaissances est de pouvoir vérifier la cohérence de ceux-ci. Parmi la grande diversité de problèmes ou de connaissances à modéliser, la modélisation par un *formalisme discret* prend une place importante en intelligence artificielle. Par le terme « formalisme discret », nous entendons un cadre de représentation des connaissances exprimées sur des domaines finis et discrets. Les connaissances sont représentées par un ensemble fini d'*entités*, chacune devant être associée à une et une seule valeur parmi un ensemble fini de valeurs, appelé son *domaine*. Par exemple, les entités peuvent être les étudiants d'une classe, chacun ayant pour domaine l'intervalle d'entiers $[1 ; d]$ représentant un numéro de chaise pour un examen. Les entités sont mutuellement soumises à des *contraintes discrètes*, chaque contrainte étant associée à une relation mathématiques sur les domaines des entités impliquées. Par exemple, une contrainte entre deux étudiants décrivant la relation de différence permet de leur imposer des numéros de chaise différents.

Le *Problème de Satisfiabilité des Contraintes*, CSP en abrégé, consiste à déterminer s'il existe une *instanciation*, c'est-à-dire l'association d'une valeur à chaque entité, satisfaisant les relations décrites par les contraintes. C'est un problème de décision, c'est-à-dire un problème pour lequel la réponse attendue est *Oui* ou *Non*. Dans la hiérarchie des problèmes de la théorie de la complexité, le problème CSP est *NP-complet*, pour *Non-Déterministe Polynomial Complet*. En d'autres termes, pour répondre à ce problème en temps polynomial ¹, il faut une machine virtuelle capable d'énumérer chaque instanciation en parallèle

1. Un processus est *polynomial en temps* signifie qu'il nécessite un temps qui est de l'ordre d'une fonction polynomiale par rapport à la taille des données

et, pour chaque instanciation, vérifier en temps polynomial si elle satisfait les contraintes du problème. En pratique et dans le cas général, répondre au problème CSP est une tâche difficile. À ce problème de décision, on associe le problème visant à exhiber une *solution* au problème. Celui-ci est *NP-difficile*, il est au moins aussi difficile d'y répondre que de répondre au problème de décision associé.

Au début des années 1970, Montanari exploite le formalisme discret dans le domaine du traitement d'image et de la reconnaissance de formes. Il représente alors les connaissances sous la forme de *réseaux de contraintes discrets*, noté RCD. Un réseau de contraintes est un hypergraphe, dont les nœuds et les hyperarêtes représentent respectivement des variables décrivant les entités, et les contraintes entre ces variables. Cette représentation permet de s'appuyer sur le cadre théorique de la théorie des graphes afin de développer les outils algorithmiques nécessaires pour décider de la cohérence des connaissances. Dans le cas général, répondre au problème CSP est une tâche difficile en pratique. L'approche consiste à associer une valeur à chaque variable afin de construire une *instanciation* des variables du RCD, puis à vérifier si celle-ci satisfait les contraintes. Si les contraintes sont satisfaites, alors l'instanciation décrit une *solution* au problème et le RCD est *cohérent*. Sinon, il est *incohérent*. Cependant, énumérer l'ensemble des instanciations nécessite un temps considérable dans le cas général. Afin de simplifier cette tâche, de nombreuses approches, appelées *techniques d'inférences*, ont été développées. Elles s'appuient sur la déduction logique d'informations en exploitant les informations structurelles et sémantiques du RCD. D'autres outils algorithmiques visent à analyser les choix, à analyser les résultats obtenus et à apprendre des erreurs effectuées. L'objectif commun est de limiter le nombre d'instanciations à explorer afin de décider de la cohérence du RCD. L'utilisation d'outils informatiques est donc indispensable pour faciliter les calculs, mais n'est pas suffisante malgré les progrès et innovations technologiques.

Dans ce chapitre, nous définissons les réseaux de contraintes discrets et les outils algorithmiques pour répondre au problème CSP. Dans un premier temps, nous donnons une définition générale des RCD et nous présentons les notions de bases afin de les exploiter. Au travers d'un exemple académique reposant sur le problème de Dirichlet, nous proposons un aperçu des techniques algorithmiques les plus utilisées. Nous présentons, dans un premier temps, les algorithmes de recherche, puis nous décrivons les outils mis en œuvre pour les améliorer au cours du temps, tel que la cohérence locale et les choix heuristiques.

1.2 Les Réseaux de Contraintes Discrets

1.2.1 Définition générale et terminologie

Modéliser un problème par un formalisme discret impose la définition des entités, des domaines, ainsi que des contraintes entre celles-ci. Un réseau de contraintes discret, noté RCD, est défini par un hypergraphe, composé d'un ensemble de nœuds et d'un ensemble d'hyperarêtes entre ceux-ci. Cela nous permet de faire l'analogie avec les entités, représentées par les nœuds, et les contraintes, représentées par les hyperarêtes de l'hypergraphe. Néanmoins, le problème à représenter intègre la notion de domaine associé à une entité et de relation associée à une contrainte. Un RCD est la représentation combinée de deux graphes, permettant de donner à la fois une vue synthétique du problème mais également de détailler sa microstructure par une représentation fine des informations.

Définition 1 (Réseau de Contraintes Discret) Un Réseau de Contraintes Discret, noté RCD, est décrit par un triplet $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ tel que :

- $\mathcal{X} = \{x_1, \dots, x_n\}$ est un ensemble fini de variables ;
- $\mathcal{D} = \{D_{x_1}, \dots, D_{x_n}\}$ est un ensemble fini de domaines tel que, à toute variable $x_i \in \mathcal{X}$, est associé un domaine fini $D_{x_i} \in \mathcal{D}$;
- \mathcal{C} décrit un ensemble de contraintes entre les variables de \mathcal{X} , tel que $\mathcal{C} = \{c_1, \dots, c_e\}$.

En suivant la définition 1, nous noterons dans le cas général par n et e , respectivement le nombre de variables et le nombre de contraintes d'un RCD. Les variables représentent les entités du problème. Le domaine associé à la variable $x_i \in \mathcal{X}$, noté $D_{x_i} \in \mathcal{D}$, est décrit par l'ensemble de valeurs $D_{x_i} = \{v_{i_1}, \dots, v_{i_d}\}$. Nous noterons d le nombre de valeurs d'un domaine dans le cas général. Les valeurs du domaine d'une variable sont complètes et mutuellement exclusives. Par conséquent, une variable $x_i \in \mathcal{X}$ est associée à une et une seule valeur de son domaine D_{x_i} . Cela représente une contrainte implicite du problème CSP.

Historiquement, un RCD était parfois appelé *réseau de relations* [MACKWORTH 1977]. Une contrainte décrit une relation, définie sur les domaines de certaines variables du réseau de contraintes. Une contrainte $c \in \mathcal{C}$ implique un ensemble de variables $X \subseteq \mathcal{X}$, appelé la *portée* de la contrainte. Le nombre de variables impliquées dans une contrainte est appelé *arité*, et il est communément noté r . Une contrainte d'arité 2 est une *contrainte binaire*, tandis qu'une contrainte d'arité r est une *contrainte r-aire*. L'application scp permet d'associer cette portée à une contrainte, telle que $scp(c) = X$. Ainsi, $|scp(c_i)| = r$. Afin de définir la relation décrite par une contrainte, nous définissons une application de produit cartésien π des domaines d'un ensemble de variables.

Définition 2 (Produit cartésien des domaines) π est une application qui, à tout ensemble $X \subseteq \mathcal{X}$, associe $\pi(X) = \prod_{x \in scp(X)} D_x$.²

Cette définition nous permet, pour tout $c \in \mathcal{C}$ tel que $scp(c) = X$, de décrire sa relation par $rel(c) \subseteq \pi(X)$, où rel est une application qui, à toute contrainte, associe sa relation. Toute contrainte $c \in \mathcal{C}$ telle que $rel(c) = \pi(X)$ est appelée contrainte universelle. Dans la suite, afin de simplifier la lecture, nous noterons par $c_{(x_{i_1}, \dots, x_{i_r})}$ la relation $rel(c)$ de la contrainte $c \in \mathcal{C}$ de portées $\{x_{i_1}, \dots, x_{i_r}\}$.

La définition de la portée des contraintes permet de définir une notion de voisinage qui s'apparente à la notion de voisinage dans un hypergraphe.

Définition 3 (Voisinage) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD et deux variables $x_i, x_j \in \mathcal{X}$ avec $i \neq j$. x_i et x_j sont voisines si et seulement si il existe une contrainte $c \in \mathcal{C}$ telle que $x_i, x_j \in scp(c)$. L'ensemble des variables voisines d'une variable x_i est appelé *voisinage* de x_i .

Définition 4 (Réseau de contraintes complet) Un réseau de contraintes $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ est dit *complet* si et seulement si pour tout $x_i, x_j \in \mathcal{X}$, avec $i \neq j$, x_i est voisine de x_j .

Il nous est possible d'utiliser les opérations ensemblistes usuelles sur les relations des contraintes, telles que l'intersection ou l'union, mais aussi d'utiliser les relations d'appartenance, d'inclusion, etc. Par ailleurs, tout élément $\tau \in rel(c)$ est appelé *tuple* ou *r-uplet*, que l'on peut assimiler à une liste ordonnée de valeurs. Pour un tuple $\tau \in c_{(x_{j_1}, \dots, x_{j_r})}$ tel que $\tau = (v_{i_1}, \dots, v_{i_r})$, nous utiliserons la notation $\tau[x_{j_k}]$ afin d'indiquer la valeur v_{i_k} de x_{j_k} dans le tuple τ , telle que $\tau[x_{j_k}] = v_{i_k}$. Nous utiliserons également la notation $\tau[\hat{x}_{j_k}]$ pour décrire le tuple $(v_{i_1}, \dots, v_{i_{k-1}}, v_{i_{k+1}}, \dots, v_{i_r})$.

Les notations relatives à l'arité d'une contrainte ont été étendues au réseau de contraintes lui-même. Ainsi, un RCD dont l'arité la plus grande est r est un RCD *r-aires*. De nombreux travaux ont permis de définir la transformation d'une contrainte *r-aire* en contraintes binaires [ROSSI *et al.* 1990, DECHTER 1990, BACCHUS & VAN BEEK 1998, STERGIOU & WALSH 1999], et donc de transformer un réseau de contraintes *r-aires* en réseau de contraintes binaires.

Trois formes de contraintes peuvent être définies. Les contraintes en *extension* énumèrent l'ensemble des tuples d'une relation sous une forme détaillée. Les contraintes en *intention* permettent la description d'une relation sous une forme « compilée », notamment à l'aide de relations mathématiques entre

2. $\prod_{e \in E}$ dénotera le produit cartésien sur les éléments $e \in E$

les variables ou de descriptions ensemblistes. Enfin, les *contraintes globales* [LAURIÈRE 1978] sont des contraintes permettant d’englober plusieurs contraintes et de décrire de manière concise la relation. Elles permettent, dans la plupart des cas, de conserver le sens de la contrainte du problème. Dans notre exemple, la contrainte globale *allDiff* ou « toutes différentes » permet de préciser que les valeurs choisies pour les variables impliquées doivent toutes être différentes [REGIN 1999]. Les contraintes globales sont donc des contraintes en intention permettant de garder le sens de la contrainte, ce qui permet d’adapter la recherche à chaque contrainte globale. Il existe un catalogue de plus de 350 contraintes globales [BELDICEANU *et al.* 2009]. Nous concentrerons notre étude sur les représentations des contraintes en extension ou en intention.

Enfin, à chaque forme de contrainte peut être associée une « orientation ». Une contrainte $c \in \mathcal{C}$ de portée (x_1, \dots, x_r) représente les *supports* si les tuples qui la décrivent représentent les combinaisons de valeurs permises afin de satisfaire c . À l’inverse, nous noterons par $\bar{c}_{(x_1, \dots, x_r)} = \pi(X) \setminus c_{(x_1, \dots, x_r)}$ la relation représentant les *conflits*. On parle de contraintes supports (resp. conflits) décrites par des tuples supports (resp. conflits). Cette définition montre que les contraintes conflits peuvent être déduites des contraintes supports et *vice versa*. Une contrainte $c \in \mathcal{C}$ et $\bar{c} \in \mathcal{C}$ représente la même contrainte mais permet de préciser son orientation. Sans précision de notre part, nous mentionnerons implicitement des contraintes supports. Outre les contraintes globales, qui sont un support d’étude particulier, la distinction entre les différentes formes de contraintes a essentiellement un intérêt pratique.

Dans la suite, pour imposer un cadre d’étude, nous définissons les RCD *normalisés*. Cela permet notamment de ne pas traiter certains cas particuliers afin d’alléger les définitions et propriétés en étudiant une seule contrainte sur un même ensemble de variables.

Définition 5 (RCD normalisé) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un réseau de contraintes. \mathcal{P} est normalisé si et seulement si $\text{scp}(c) \neq \text{scp}(c')$ pour toutes contraintes $c, c' \in \mathcal{C}$.

Dans le cas général, tout RCD n’est pas normalisé, cependant unifier les contraintes de même portée peut être fait facilement par intersection des relations. Dans la suite, nous ferons uniquement référence aux réseaux de contraintes normalisés.

Comme défini précédemment, un réseau de contraintes est caractérisé par une représentation synthétique et la microstructure d’un problème modélisé. Nous décrivons donc une un ordre partiel sur la microstructure afin définir une relation d’inclusion entre les RCD.

Définition 6 (Inclusion de RCD) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ et $\mathcal{P}' = (\mathcal{X}, \mathcal{D}', \mathcal{C}')$ deux réseaux de contraintes. \mathcal{P} est un sous-réseau de \mathcal{P}' , noté $\mathcal{P} \sqsubseteq \mathcal{P}'$, si et seulement si $\text{rel}(c) \subseteq \text{rel}(c')$ pour toutes contraintes $c \in \mathcal{C}$ et $c' \in \mathcal{C}'$ telles que $\text{scp}(c) = \text{scp}(c')$.

Définition 7 (RCD atomique) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD. \mathcal{P} est dit atomique si et seulement si $|D_x| = 1$, pour tout $x \in \mathcal{X}$.

Remarquons, quel que soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, qu’il existe un nombre fini de RCD atomiques $\mathcal{P}' = (\mathcal{X}, \mathcal{D}', \mathcal{C}')$ tel que $\mathcal{P}' \sqsubseteq \mathcal{P}$.

L’exemple 1 illustre la modélisation de connaissances par un formalisme discret en exploitant la représentation des RCD.

Exemple 1 Pour représenter un problème nécessitant la modélisation par le formalisme discret, utilisons un exemple académique. Un examen va débiter : n étudiants, représentant les entités, doivent prendre place dans la salle d’examen, dans laquelle d chaises sont disponibles. Chacun des étudiants a la possibilité de s’asseoir sur l’une des d chaises. Chaque étudiant a donc pour domaine l’ensemble des chaises de la salle. Pour répondre aux us et coutumes, les contraintes entre les étudiants spécifient que

chaque étudiant doit s'asseoir sur une chaise et une seule, et qu'il est interdit de s'asseoir sur la chaise occupée par l'un(e) de ses camarades.

Les étudiants sont représentés par les variables, notées $\mathcal{X} = \{E_1, \dots, E_n\}$. Le domaine de chaque variable $x_i \in \mathcal{X}$ est décrit par les chaises tel que $D_{E_i} = [1 ; d]$. Les contraintes spécifient que chaque étudiant doit être assis sur une chaise et une seule. Cette contrainte est implicite, puisqu'on ne peut associer qu'une seule valeur à une variable. De plus, deux étudiants ne peuvent prendre place sur une même chaise. Par conséquent, pour tout $E_i, E_j \in \mathcal{X}$ avec $i \neq j$, nous définissons la relation $c_{(E_i, E_j)} = \{(v_{i_k}, v_{j_{k'}}) \in D_{E_i} \times D_{E_j} : v_{i_k} \neq v_{j_{k'}}\}$. La figure 1.1 illustre la représentation de ces connaissances, pour quatre étudiants et quatre chaises, sous la forme de deux RCD.

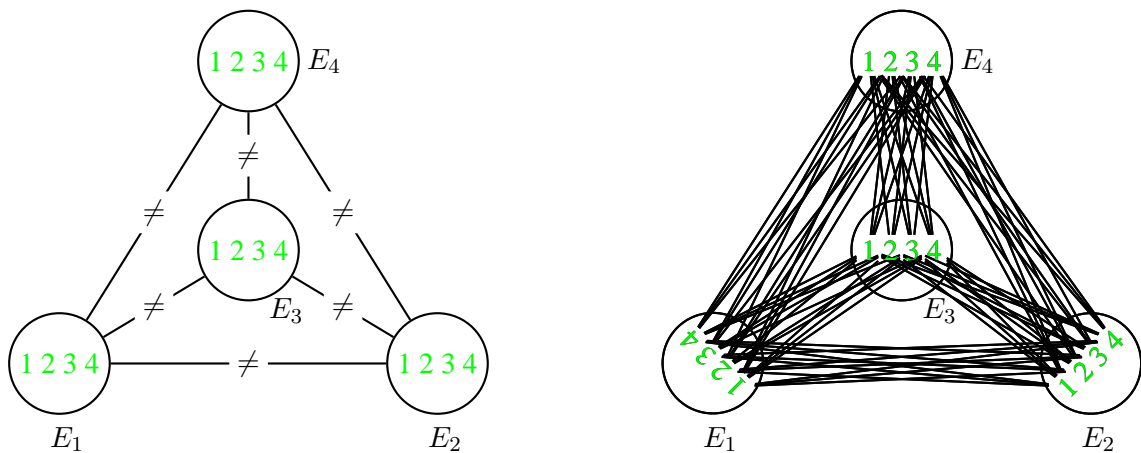


FIGURE 1.1 – Problème des étudiants pour $n = 4$ étudiants et $d = 4$ chaises.

L'exemple 1 illustre plusieurs aspects de la modélisation sous la forme d'un réseau de contraintes. En effet, celle-ci est influencée par la compréhension et l'analyse du problème qui n'est pas unique. Par exemple, une autre modélisation du problème aurait pu être de décrire les chaises en tant que variables, et les étudiants sous la forme du domaine associé à chacune des chaises. Un même problème peut donc être représenté sous la forme de réseaux de contraintes différents.

La figure 1.1 présente deux formes de contraintes différentes, utilisées pour décrire un même RCD. Les contraintes du RCD de gauche sont exprimées en intention, tandis que les contraintes du RCD de droite représente les tuples supports en extension. L'aspect graphique de ces deux formes de contraintes illustre parfaitement les avantages et inconvénients des deux approches. Du côté intention, déterminer si les tuples précisément demande un calcul sur la relation décrite afin de faire le lien entre les valeurs. Cependant, cette représentation est claire et concise. Du côté extension, les informations sont toutes représentées par les tuples mais le RCD est surchargé. Cela permet d'illustrer l'intérêt pratique de chaque représentation des contraintes, avec un compromis entre la facilité de manipulation et la quantité d'informations à sauvegarder.

1.2.2 Instanciation et cohérence

À partir de la modélisation d'un problème sous la forme d'un RCD, nous souhaitons vérifier si celui-ci possède ou non une solution. Il est donc nécessaire de définir certaines structures et opérations à effectuer. À partir de $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, le problème CSP est décrit par la question : est-il possible d'associer à toute variable $x \in \mathcal{X}$, une et une seule valeur $v \in D_x$, telle que toutes les contraintes soient satisfaites. Si oui, alors \mathcal{P} est *cohérent* ou *satisfiable* et le problème modélisé admet une solution. Sinon, \mathcal{P} est *incohérent* ou *insatisfiable* et le problème modélisé n'admet pas de solutions.

Pour donner une réponse à cette question, il est nécessaire de définir quelques notions, telles que l'association d'une valeur à une variable ou la *satisfaction* des contraintes. Tout d'abord, caractérisons l'association d'une valeur à une variable.

Définition 8 (Instanciation) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD. Une instanciation d'un ensemble de variables $X \subseteq \mathcal{X}$ de \mathcal{P} , notée \mathcal{I}_X , est une application qui, à tout $x \in X$, associe une valeur $v \in D_x$.

Une instanciation \mathcal{I}_X est appelée *instanciation complète* si et seulement si $X = \mathcal{X}$, et *instanciation partielle* sinon. Dans le cas général, nous mentionnerons une instanciation sans distinction, et nous préciserons le cas échéant. La construction d'une instanciation peut être rapprochée de deux opérations.

Définition 9 (Affectation) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD. L'affectation d'une valeur $v \in D_x$ à une variable $x \in \mathcal{X}$ de \mathcal{P} , notée $x = v$, est une instanciation $\mathcal{I}_{\{x\}}$ telle que $\mathcal{I}_{\{x\}}(x) = v$.

En utilisant cette définition, nous pouvons définir un opérateur $|$ permettant d'appliquer une instanciation à un RCD.

Définition 10 (Application d'un instanciation) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD et \mathcal{I}_X une instanciation sur $X \subseteq \mathcal{X}$. $\mathcal{P}_{|\mathcal{I}_X}$ est un RCD $\mathcal{P}' = (\mathcal{X}, \mathcal{D}', \mathcal{C})$ tel que pour tout $x \in \mathcal{X}$, si $x \in X$ alors $D'_x = \{\mathcal{I}_X(x)\}$, $D'_x = D_x$ sinon.

Notons que, quelle que soit une instanciation \mathcal{I}_X sur $X \subseteq \mathcal{X}$, nous avons $\mathcal{P}_{|\mathcal{I}_X} \sqsubseteq \mathcal{P}$. La notation $\mathcal{P}_{|\mathcal{I}_X}$ peut être interprétée comme « \mathcal{P} sachant \mathcal{I}_X ». Appliquer une instanciation \mathcal{I}_X , avec $X \neq \emptyset$, à un réseau de contraintes \mathcal{P} est donc équivalent à appliquer plusieurs affectations à \mathcal{P} . Appliquer une affectation $x = v$ se traduit par la suppression de toute autre valeur que v dans le domaine de x . Ce qui nous permet de définir la suppression d'une valeur dans le domaine d'une variable.

Définition 11 (Réfutation) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un réseau de contraintes. La réfutation de la valeur $v \in D_x$ de la variable $x \in \mathcal{X}$, notée $x \neq v$, décrit la suppression de la valeur au domaine tel que $v \notin D_x$.

Ainsi, nous admettons qu'il est également possible d'appliquer une réfutation à un RCD. $\mathcal{P}_{|x_i \neq v}$ est un réseau de contraintes $\mathcal{P}' = (\mathcal{X}, \mathcal{D}', \mathcal{C})$ tel que $D'_{x_i} = D_{x_i} \setminus \{v\}$ et pour tout $x_j \in \mathcal{X}$, avec $i \neq j$, $D'_{x_j} = D_{x_j}$. Il est facile de voir que l'application d'une instanciation est équivalente à l'application d'un ensemble de réfutations. Appliquer une affectation à un RCD est également appelé *décision positive* et appliquer une réfutation est appelée *décision négative*.

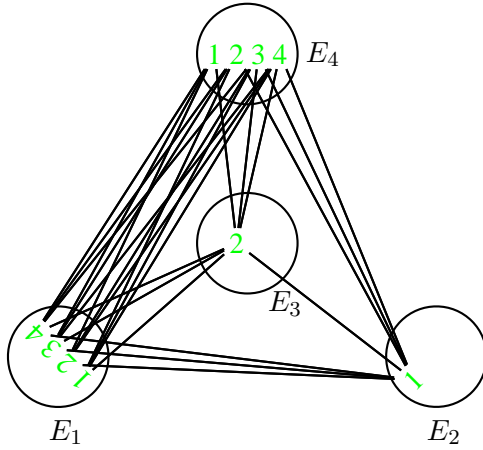
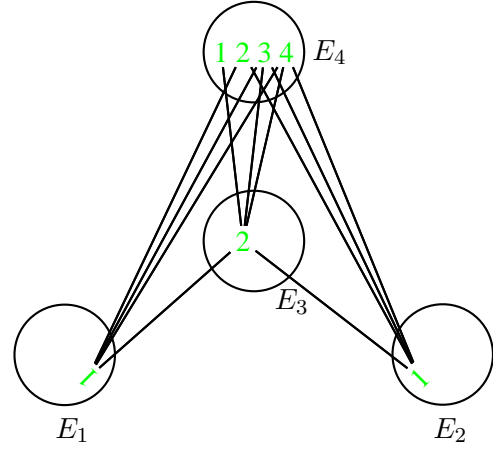
Une instanciation est une application et donc, par notre définition, un ensemble de couples variable/valeur. En cela, il est possible d'appliquer aux instanciations les opérations ensemblistes usuelles.

Définition 12 (Restriction et prolongement) Soient \mathcal{I}_X et $\mathcal{I}_{X'}$ deux instanciations de $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ sur $X \subseteq \mathcal{X}$ et $X' \subseteq \mathcal{X}$ respectivement.

- $\mathcal{I}_{X'}$ est la restriction de \mathcal{I}_X si et seulement si $\mathcal{I}_{X'} \subseteq \mathcal{I}_X$;
- $\mathcal{I}_{X'}$ est le prolongement de \mathcal{I}_X si et seulement si $\mathcal{I}_X \subseteq \mathcal{I}_{X'}$.

En utilisant la définition 12, nous pouvons également définir l'union et l'intersection d'instanciations. L'union $\mathcal{I}_{X \cup X'} = \mathcal{I}_X \cup \mathcal{I}_{X'}$, avec $X \cap X' = \emptyset$, est le plus petit prolongement commun à \mathcal{I}_X et $\mathcal{I}_{X'}$. L'intersection $\mathcal{I}_{X \cap X'} = \mathcal{I}_X \cap \mathcal{I}_{X'}$ est la plus grande restriction commune à \mathcal{I}_X et $\mathcal{I}_{X'}$.

Les figures 1.2 et 1.3 illustrent l'application de deux instanciations au RCD de l'exemple 1, noté \mathcal{P} . La figure 1.2 représente ainsi $\mathcal{P}_{|\mathcal{I}_{\{E_2, E_3\}}}$ avec $\mathcal{I}_{\{E_2, E_3\}} = \{(E_2, 1), (E_3, 2)\}$, c'est-à-dire $E_2 = 1$ et $E_3 = 2$. La figure 1.3 représente $\mathcal{P}_{|\mathcal{I}'_{\{E_1, E_2, E_3\}}}$ tel que $\mathcal{I}'_{\{E_1, E_2, E_3\}} = \mathcal{I}_{\{E_2, E_3\}} \cup \{(E_1, 1)\}$. Notons que la réfutation de valeurs dans les domaines occasionne une suppression des tuples des contraintes, afin de faciliter la lecture. Les tuples supprimés pour une contrainte c sont ceux n'appartenant plus à la projection des domaines $\pi(\text{scp}(c))$. Par ailleurs, la Figure 1.3 présente une absence de tuples entre les variables E_1 et E_2 , ce qui dénote une incohérence de l'instanciation dans le réseau de contraintes puisque la contrainte de différence n'est pas satisfaite.


 FIGURE 1.2 – $\mathcal{P}_{|\mathcal{I}_{\{E_2, E_3\}}}$

 FIGURE 1.3 – $\mathcal{P}_{|\mathcal{I}'_{\{E_1, E_2, E_3\}}}$

La définition d'instanciation nous rapproche des pré-requis permettant de répondre au problème CSP. Néanmoins, il nous reste à définir la notion de satisfaction d'une contrainte.

Définition 13 (Satisfaction d'une contrainte par une instanciation) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, $c \in \mathcal{C}$ une contrainte et \mathcal{I}_X une instanciation sur $X \subseteq \mathcal{X}$ telle que $\text{scp}(c) \subseteq X$. c est satisfaite par \mathcal{I}_X si et seulement si $(\mathcal{I}_X(x_i), \dots, \mathcal{I}_X(x_j)) \in c_{(x_i, \dots, x_j)}$.

Définition 14 (Cohérence d'une instanciation) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD et \mathcal{I}_X une instanciation sur $X \subseteq \mathcal{X}$. \mathcal{I}_X est cohérente dans \mathcal{P} si et seulement si toute contrainte $c \in \mathcal{C}$, telles que $\text{scp}(c) \subseteq X$, est satisfaite par \mathcal{I}_X . Elle est incohérente sinon.

Définition 15 (Solution) Soit \mathcal{P} un RCD. Une solution de \mathcal{P} est une instanciation complète cohérente de \mathcal{P} .

L'ensemble des instanciations d'un réseau de contraintes \mathcal{P} est appelé son *espace de recherche*. L'ensemble des instanciations qui ont pour restriction une instanciation \mathcal{I}_X de \mathcal{P} , avec $X \subseteq \mathcal{X}$, est appelé *sous-espace de recherche* de \mathcal{P} associé à \mathcal{I}_X . Clairement, le sous-espace de recherche de \mathcal{P} associé à \mathcal{I}_\emptyset est l'espace de recherche complet de \mathcal{P} , puisque $\mathcal{P} = \mathcal{P}_{|\mathcal{I}_\emptyset}$. Plus généralement, l'espace de recherche associé au réseau de contraintes $\mathcal{P}_{|\mathcal{I}_X}$ coïncide avec le sous-espace de recherche de \mathcal{P} associé à \mathcal{I}_X . Dans le cas général, quels que soient $\mathcal{P}, \mathcal{P}'$ deux RCD, si $\mathcal{P} \sqsubseteq \mathcal{P}'$, alors l'espace de recherche de \mathcal{P} est inclus dans l'espace de recherche de \mathcal{P}' .

Définition 16 (Cohérence d'un RCD) Un RCD est cohérent si et seulement si l'espace de recherche de celui-ci contient au moins une solution. Dans le cas contraire, il est incohérent.

La définition de la cohérence des RCD nous permet de déduire qu'un RCD \mathcal{P} est cohérent si et seulement si il possède au moins un sous-réseau atomique $\mathcal{P}' \sqsubseteq \mathcal{P}$ tel que \mathcal{P}' est cohérent. De plus, remarquons que s'il existe une variable $x \in \mathcal{X}$ telle que $D_x = \emptyset$ alors le RCD est trivialement incohérent, puisque l'espace de recherche ne contient pas d'instanciations complètes, et par conséquent pas de solutions.

Les figures 1.5 et 1.4 illustrent deux RCD atomiques issus du RCD décrivant le problème de l'exemple 1, noté \mathcal{P} . La figure 1.4 présente le RCD $\mathcal{P}_{|\mathcal{I}'}$, avec $\mathcal{I}' = \{(E_1, 1), (E_2, 1), (E_3, 2), (E_4, 3)\}$. \mathcal{I}' est incohérente car la contrainte entre E_1 et E_2 n'est pas respectée. La figure 1.5 présente le RCD $\mathcal{P}_{|\mathcal{I}}$, avec $\mathcal{I} = \{(E_1, 4), (E_2, 1), (E_3, 2), (E_4, 3)\}$. \mathcal{I} est une solution puisque c'est une instanciation complète satisfaisant chaque contrainte. L'espace de recherche de \mathcal{P} contient donc au moins une solution, \mathcal{P} est cohérent.

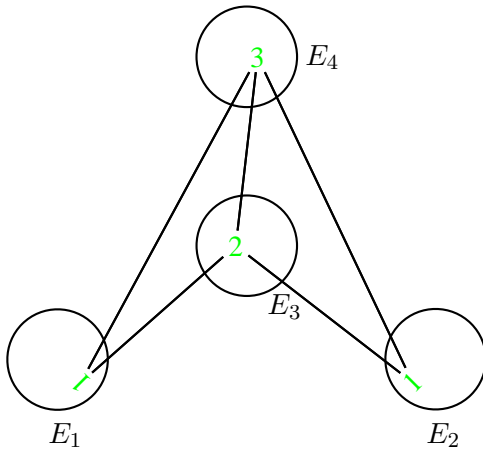


FIGURE 1.4 – $\mathcal{P}_{|\mathcal{I}'}$

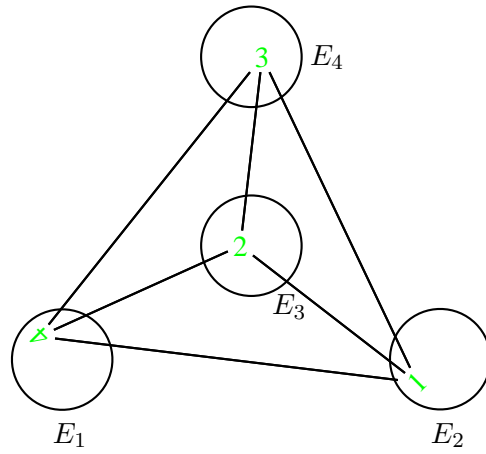


FIGURE 1.5 – $\mathcal{P}_{|\mathcal{I}}$

Le problème CSP s'attache donc à déterminer l'existence d'une solution dans un RCD \mathcal{P} , ou de manière équivalente l'existence d'un RCD atomique $\mathcal{P}' \sqsubseteq \mathcal{P}$ cohérent. Notons qu'il n'est en aucun cas nécessaire de déterminer une solution pour répondre au problème CSP, mais qu'il faut en prouver l'existence. Dans la suite, nous noterons un réseau de contraintes incohérent, quel qu'il soit, par un symbole unique \perp .

Propriété 1 Soient \mathcal{P} et \mathcal{P}' deux RCD tels que $\mathcal{P} \sqsubseteq \mathcal{P}'$. Si \mathcal{P} est cohérent, \mathcal{P}' l'est aussi.

Preuve Cette propriété est trivialement vérifiée en considérant que si \mathcal{P} est cohérent alors il admet un sous-réseau atomique cohérent $\mathcal{P}'' \sqsubseteq \mathcal{P}$. Par conséquent, $\mathcal{P}'' \sqsubseteq \mathcal{P} \sqsubseteq \mathcal{P}'$ et donc \mathcal{P}'' est également un sous-réseau atomique cohérent de \mathcal{P}' . \mathcal{P}' est donc cohérent, et il admet une solution \mathcal{I} telle que $\mathcal{P}'' = \mathcal{P}'_{|\mathcal{I}}$. \square

Notons, par contraposée de la propriété 1, que si $\mathcal{P}' = \perp$, alors $\mathcal{P} = \perp$. Puisque si l'espace de recherche de \mathcal{P}' ne contient pas de solutions, l'espace de recherche \mathcal{P} n'en contient pas non plus.

Le terme « cohérence » a deux aspects différents. On parle de cohérence pour un RCD et de cohérence pour une instanciation partielle. Cependant, il est clair que toute instanciation partielle cohérente n'implique pas d'avoir un RCD cohérent. Nous distinguons donc deux formes de cohérence d'une instanciation. La définition 14 décrit la cohérence de l'instanciation elle-même, s'appuyant sur les contraintes impliquant les variables de celles-ci. Plus généralement, nous pouvons étendre la notion de cohérence aux

voisins des variables d'une instanciation. Lorsqu'il n'existe pas de contrainte du RCD non satisfaite par une instanciation partielle, on dit qu'elle *cohérente localement*.

Définition 17 (Cohérence locale) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, \mathcal{I}_X une instanciation sur $X \subseteq \mathcal{X}$ et $\mathcal{P}' = (\mathcal{X}, \mathcal{D}', \mathcal{C})$ le RCD tel que $\mathcal{P}' = \mathcal{P}_{|\mathcal{I}_X}$. \mathcal{I}_X est localement cohérente si et seulement si, pour toute contrainte $c \in \mathcal{C}$, $\pi(\text{scp}(c)) \cap \text{rel}(c) \neq \emptyset$. \mathcal{P}' est dit alors localement cohérent.

Définition 18 (Cohérence globale) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un réseau de contraintes. Une instanciation de \mathcal{P} sur $X \subseteq \mathcal{X}$ est globalement cohérente si et seulement si elle est la restriction d'une solution de \mathcal{P} .

Propriété 2 Une instanciation \mathcal{I}_X de \mathcal{P} sur $X \subseteq \mathcal{X}$ est globalement cohérente si et seulement si toutes ses restrictions $\mathcal{I}_{X'}$ de \mathcal{P} sur $X' \subseteq X$ sont globalement cohérentes.

Preuve La preuve est immédiate. Pour le sens *si* : \mathcal{I}_X est globalement cohérente permet d'affirmer que \mathcal{I}_X est la restriction d'une solution. Or, quel que soit l'ensemble $X' \subseteq X$, $\mathcal{I}_{X'}$ est une restriction de \mathcal{I}_X . Donc pour tout $X' \subseteq X$, $\mathcal{I}_{X'}$ est la restriction d'une solution. Par conséquent, pour tout $X' \subseteq X$, $\mathcal{I}_{X'}$ est globalement cohérente. Le sens *seulement si* est trivialement vrai pour $X = X'$. Faisons maintenant l'hypothèse qu'il existe une restriction $\mathcal{I}_{X'}$ pour $X' \subset X$ qui ne soit pas globalement cohérente alors que \mathcal{I}_X l'est. \mathcal{I}_X est la restriction d'une solution, mais $\mathcal{I}_{X'}$ ne l'est pas. Or $X' \subset X$, donc $\mathcal{I}_{X'}$ n'est pas une restriction de \mathcal{I}_X , ce qui contredit l'hypothèse. Donc il n'existe pas de restriction d'une instanciation globalement cohérente qui ne soit pas, elle-même, globalement cohérente. L'équivalence est donc démontrée, et par conséquent la propriété est prouvée. \square

La cohérence globale est une définition *plus forte* que la cohérence locale. Si une instanciation est globalement cohérente, alors elle est localement cohérente. En d'autres termes, si une instanciation \mathcal{I}_X de \mathcal{P} sur $X \subseteq \mathcal{X}$ est localement incohérente, son sous-espace de recherche associé ne contient pas de solutions, c'est-à-dire $\mathcal{P}_{|\mathcal{I}_X} = \perp$. Par la même occasion, nous pouvons affirmer qu'un RCD est localement, resp. globalement, cohérent si et seulement si il est issu d'une instanciation localement, resp. globalement, cohérente. Prouver qu'une instanciation est globalement cohérente permet de répondre au problème de satisfiabilité des contraintes, il est donc facile de voir que ce problème est au moins aussi difficile que le problème de la satisfiabilité des contraintes.

Nous avons défini les notions de base permettant de décider de la cohérence d'un RCD. Pour prouver qu'un RCD est cohérent, il est nécessaire de prouver l'existence d'une solution. Dans le cas général, n variables ont chacune un domaine fini de taille d . L'espace de recherche contient donc d^n instanciations complètes par définition. Par conséquent, énumérer les instanciations complètes de l'espace de recherche peut toujours être effectué en temps fini. ce qui est illustré dans l'exemple 2.

Exemple 2 Considérons l'exemple 1 pour $n = 20$ étudiants et $d = 20$ chaises, l'espace de recherche contient 20^{20} , soit $1,05 \times 10^{26}$, instanciations complètes. En schématisant la modélisation et les traitements informatique, si construire une instanciation complète nécessite 1 opération pour un microprocesseur, et si celui-ci permet de traiter 2×10^{10} opérations par seconde, il faudrait plus de 165 millions d'années pour énumérer les instanciations complètes de l'espace de recherche. À cela, il faut ajouter l'opération consistant à vérifier si une instanciation complète est une solution ou non. Pour se donner une idée, on estime que l'univers est « seulement » composé de 10^{84} atomes, ce qui est 8 fois moins important que le nombre d'instanciations complètes pour notre problème s'il y a $n = 50$ étudiants et $d = 50$ chaises. Les dimensions de l'espace de recherche pour les problèmes

Il est difficile de compter uniquement sur la puissance de calcul d'un ordinateur pour résoudre le problème de la satisfiabilité en énumérant uniquement l'espace de recherche et ces instanciations complètes.

Dans le cas général, un problème simple par son énoncé peut se transformer en problème complexe une fois modélisé. Cependant, pour décider de la cohérence d'un RCD, il est fait l'hypothèse qu'il n'y a aucune connaissance du problème modélisé. Le développement d'outils algorithmiques s'appuie donc sur les propriétés des formalismes discrets. Ceci constitue le cadre de l'algorithmique pour l'*inférence*, déduire par la logique des informations pour résoudre un problème, et la *prise de décision*, choisir et adapter les techniques de résolution face aux problèmes rencontrés.

1.3 Algorithmes pour l'inférence et la prise de décision

1.3.1 Algorithmes de recherche

Initialement, les méthodes permettant de répondre au problème de la satisfiabilité des contraintes sont naïves, au sens où aucune information n'est déduite et aucune amélioration n'est apportée. Bien que le problème de décision CSP s'attache à prouver l'existence d'une solution, les algorithmes dans le cadre discret se concentrent très souvent sur la recherche d'une d'entre elle. En pratique, il existe deux grandes familles d'algorithmes de recherche : les algorithmes de recherche *complets* et *incomplets*. Les algorithmes de recherche complets permettent de répondre au problème de la satisfiabilité, en prouvant qu'il existe une solution ou que l'espace de recherche n'en contient pas. Les algorithmes de recherche incomplets permettent, quant à eux, de prouver soit la satisfiabilité, soit l'insatisfiabilité d'un problème, et non les deux. Pour ces deux approches, nous présentons deux algorithmes de recherche les plus connus pour répondre au problème CSP : la *recherche locale* [SELMAN *et al.* 1992] pour la méthode de résolution incomplète et la *recherche exhaustive* [DAVIS *et al.* 1962] pour la méthode de résolution complète. Notons qu'il existe une approche permettant de combiner la recherche locale et la recherche exhaustive, afin de tirer partie des avantages des deux approches. Cependant, nous ne développerons pas ce point, malgré tous les intérêts que cette approche puissent susciter.

La recherche locale

Un algorithme de recherche locale est une approche de résolution stochastique basée sur la construction aléatoire d'instanciations complètes [SELMAN *et al.* 1992, SELMAN *et al.* 1995]. À partir de l'une d'elle, les contraintes du RCD sont vérifiées. Si l'instanciation est cohérente, le problème est satisfiable et celle-ci est une solution. Dans le cas contraire, il convient de déterminer la source de l'incohérence, c'est-à-dire les contraintes $C \subseteq \mathcal{C}$ n'ont pas été satisfaites. Ceci permet de corriger localement l'instanciation afin de réparer les incohérences. Parmi toutes les variables impliquées dans les contraintes, une variable $x \in \bigcup_{c \in C} scp(c)$ est choisie. La restriction $\mathcal{I}_{|\mathcal{X} \setminus \{x\}}$ est conservée et une opération, appelée *flip* ou *switch*,³ détermine une nouvelle affectation pour x . Ceci permet de contruire une nouvelle instanciation \mathcal{I}' par prolongement de $\mathcal{I}_{|\mathcal{X} \setminus \{x\}}$ sur x . Les contraintes sont à nouveau vérifiées pour \mathcal{I}' et le procédé se poursuit jusqu'à ce qu'une solution soit trouvée. L'algorithme 1 est un algorithme de recherche locale « élémentaire », ayant pour but de rechercher une solution dans un RCD. De nombreuses améliorations peuvent y être apportées. Une distance peut être définie permettant de s'approcher vers une solution potentielle. Par exemple, le couple variable/valeur le plus intéressant pour le *switch* peut être celui qui permettra la satisfaction du plus grand nombre de contraintes. Si, pour une variable choisie, il n'est pas possible d'améliorer l'instanciation courante à l'aide de la distance définie, on parle de *minimum local*. Si certains critères montrent qu'il est nécessaire de sortir de cette situation, comme l'impossibilité de sa-

3. Le terme *Flip* est utilisé dans le cadre booléen ou une variable est soit vraie, soit fausse. On peut le traduire par *inverser*. Pour les réseaux de contraintes, le choix se porte sur un ensemble de valeurs, une valeur est donc remplacée par une autre. Le terme *switch* peut donc désigner *intervenir*.

tisfaire plus de contraintes, une nouvelle instanciation est choisie, et la recherche se poursuit. L'exemple 3 décrit l'algorithme de recherche locale sans amélioration sur un problème concret.

Algorithme 1: Recherche Locale

Entrées : $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$: un RCD
Sortie : \mathcal{P} modélise-t-il un problème satisfiable ?

```

1 début
2    $\mathcal{I} \leftarrow$  Générer une instanciation complète pour  $\mathcal{P}$ 
3   tant que ( $\mathcal{I}$  est incohérente) faire
4      $X \leftarrow \bigcup_{c \in \mathcal{C}} scp(c) : c$  non satisfaite
5      $x_i \leftarrow$  Choisir une variable dans  $X$ 
6      $v \leftarrow$  Choisir  $v \in D_{x_j}$ 
7      $\mathcal{I}(x_i) \leftarrow v$ 
8   retourner Oui

```

Exemple 3 Appliquons l'algorithme 1 sur l'exemple 1. Avant de rentrer dans la salle, l'enseignant tire au sort aléatoirement une chaise par étudiant, afin de construire une instanciation complète. Tout le monde s'assied et les contraintes sont vérifiées. Si des étudiants sont assis sur les genoux de leurs camarades, l'un des étudiants se lève, puis une nouvelle chaise lui est affectée. Ce procédé est réitéré jusqu'à ce qu'une solution soit trouvée.

L'« inconvéient » de cette approche est exprimé dans la description de celle-ci : le procédé se poursuit jusqu'à la preuve de la cohérence du RCD. Une instanciation complète peut être vérifiée plusieurs fois, sans que l'algorithme n'y prête attention. Comme énoncé précédemment, le nombre d'instanciations complètes est astronomique et préserver en mémoire celles qui ont été vérifiées est inconcevable dans le cas général. Cette approche ne permet donc pas de prouver qu'il n'existe pas de solutions. En pratique, le temps laissé à l'algorithme est limité. L'algorithme répond donc par « oui », si le RCD est cohérent, ou « résultat inconnu », si le temps imparti est écoulé. Cet algorithme est donc correct mais incomplet.

Cette approche est très utilisée pour de nombreux problèmes d'optimisations et, paradoxalement, lorsque les problèmes sont insatisfiables, comme le problème MAXCSP. Elle s'attache alors à approcher le nombre maximal de contraintes qui peuvent être satisfaites, ou encore à détecter une partie du problème qui empêche la satisfiabilité des contraintes, tels que les problèmes d'extraction de sous-problèmes incohérents [GRÉGOIRE *et al.* 2006]. Cela permet de déterminer les sources d'incohérences pour ensuite réparer le problème ou les connaissances modélisés. De plus, cette approche produit de très bons résultats pour le problème CSP sur des instances difficiles à résoudre par d'autres approches, notamment l'approche complète.

La recherche exhaustive

L'approche par recherche exhaustive est celle sur laquelle se concentreront nos travaux. Comme son nom l'indique, elle va permettre de faire un parcours exhaustif de l'espace de recherche en temps fini afin de trouver une solution. Cet algorithme est donc correct et complet. Cependant, comme mentionné précédemment, il n'est pas possible de sauvegarder chaque instanciation étudiée. L'espace de recherche est donc parcouru sous la forme d'un *arbre de recherche*.

L'algorithme de recherche exhaustive suit un raisonnement par récurrence, basé sur la définition de cohérence globale d'une instanciation. Considérons $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD :

1. \mathcal{P} est cohérent si et seulement si \mathcal{I}_\emptyset est globalement cohérente ;

2. soient un ensemble de variables $X \subset \mathcal{X}$ et une variable $x \in \mathcal{X} \setminus X$. Une instantiation \mathcal{I}_X est globalement cohérente si et seulement si il existe une valeur $v \in D_x$ telle que l'instanciation $\mathcal{I}_X \cup (x = v)$ soit globalement cohérente ;
3. une instantiation complète \mathcal{I}_X de \mathcal{P} est globalement cohérente si et seulement si elle est une solution.

L'algorithme de recherche exhaustif le plus élémentaire, présenté dans Algorithme 2, est l'algorithme de *génération et vérification*, ou *generate and test* en anglais. Il permet de parcourir l'arbre de recherche *en profondeur d'abord* (DFS pour *Depth First Search*). Le parcours est donc effectué branche par branche. Il consiste à choisir une variable à chaque niveau de l'arbre et à lui affecter une valeur de son domaine. Celle-ci est choisie dans un ensemble de variables non affectées, c'est-à-dire dont la taille du domaine n'est pas 1, décrit par l'application $futures(\mathcal{P})$.

Définition 19 (Variables futures) *futures* est une application qui, à tout RCD, associe un ensemble $X \subseteq \mathcal{X}$, tel que $|D_x| > 1$ pour tout $x \in X$.

Cela permet de construire, affectation par affectation, une instantiation complète \mathcal{I} , afin de vérifier si celle-ci est une solution. Si c'est le cas, toute restriction de \mathcal{I} est globalement cohérente et donc le RCD est cohérent. Sinon, une instantiation $\mathcal{I}_X \cup (x = v)$ n'est pas globalement cohérente. Il convient de faire un choix différent pour la variable x , et ainsi construire une instantiation $\mathcal{I}_X \cup (x = v')$. Si, pour tout $v \in D_x$, $\mathcal{I}_X \cup (x = v)$ n'est pas globalement cohérente, alors \mathcal{I}_X n'est pas globalement cohérente. On effectue une opération de *retour-arrière*, ou *backtrack*, qui consiste à annuler l'affectation d'une variable de X afin de construire une nouvelle instantiation partielle. Si l'espace de recherche associé à \mathcal{I}_\emptyset , c'est-à-dire l'espace de recherche complet, ne contient pas de solutions, le RCD est incohérent. L'exemple 4 illustre l'application de l'algorithme 2 sur l'exemple 1.

Algorithme 2: Generate and Test

Entrées : $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$: un RCD
Sortie : Oui si \mathcal{P} est cohérent, Non sinon

```

1  début
2  └─ retourner Recherche( $\mathcal{P}$ )

```

```

3  fonction Recherche( $\mathcal{P}$ ) : Booléen
4  début
5  └─ si ( $\mathcal{P}$  est atomique) alors
6  │   └─ retourner Les contraintes de  $\mathcal{P}$  sont-elles satisfaites ?
7  └─ sinon
8  │   └─  $x \leftarrow$  Choisir une variable dans  $futures(\mathcal{X})$ 
9  │   └─ pour chaque ( $v \in D_x$ ) faire
10 │       └─ si (Recherche( $\mathcal{P}_{|x=v}$ ) répond Oui) alors
11 │           └─ retourner Oui
12 └─ retourner Non

```

Exemple 4 Appliquons l'algorithme *Generate and Test* à l'exemple 1. L'enseignant choisit un étudiant, puis lui assigne une chaise. Il réitère séquentiellement le procédé jusqu'à ce que chacun des étudiants soit assis. Les contraintes sont alors vérifiées et, si celles-ci ne sont pas satisfaites, le dernier étudiant assis se lève afin d'essayer la chaise suivante. Si toutes les chaises sont passées en revue sans satisfaire les contraintes, le dernier étudiant ne peut s'asseoir. Puisque le sous-espace de recherche associé aux

choix effectués pour ses camarades ne conduisent pas à une solution, celui-ci a prouvé qu'il ne pouvait y avoir de solutions si ses camarades ne changent pas de places. Lorsqu'un étudiant a montré que pour toutes chaises essayées, aucune n'a conduit à une solution, c'est son prédécesseur qui doit changer de place, et le procédé est réitéré. Si l'étudiant n'a pas de prédécesseurs, l'espace de recherche a été évalué entièrement et ne contient pas de solutions.

Cet algorithme constitue l'approche la plus élémentaire afin de garantir l'évaluation de l'intégralité de l'espace de recherche. Cependant, celui-ci présente quelques lacunes clairement visibles. En particulier, l'algorithme évalue toute instanciation localement incohérente, bien que celles-ci ne conduiront trivialement pas à une solution.

Améliorer cet algorithme de recherche signifie réduire l'espace de recherche à évaluer. La détection des instanciations partielles incohérentes [BITNER & REINGOLD 1975] fait partie de ces améliorations. Le retour-arrière est également effectué lorsque l'instanciation est localement incohérente. Cependant, déduire qu'une instanciation partielle en cours d'évaluation est localement incohérente n'est pas suffisant, car le fait de choisir une valeur qui rendra une instanciation partielle localement incohérente semble tout à fait contre-intuitif. Une amélioration peut donc être d'éliminer les valeurs des domaines qui produiront trivialement l'incohérence de l'instanciation après affectation. Ce type d'amélioration s'appelle l'*anticipation* ou *look-ahead* en anglais. Plusieurs formes de stratégies par anticipation ont alors été proposées. Parmi celles-ci, la *vérification en avançant*, plus communément dénommée *forward checking* [HARALICK & ELLIOTT 1979], s'attache à choisir un couple variable/valeur qui, avec l'instanciation courante, ne formera pas une instanciation localement incohérente. D'autres formes d'algorithmes s'appuient sur une anticipation forte en étendant la définition de cohérence locale d'une instanciation. Outre l'exploitation fine de la cohérence locale, d'autres approches peuvent être définies. En analysant l'algorithme, nous remarquons un choix arbitraire de la variable, et un parcours arbitraire de l'ensemble des valeurs. Or, une incohérence provient peut-être d'un mauvais choix du couple variable/valeur utilisé pour l'affectation. Un meilleur choix permettrait d'évaluer un sous-espace de recherche contenant une solution.

Dans la suite, nous présentons un aperçu, non-exhaustif, des améliorations qui peuvent se greffer à l'algorithme *Generate and Test*, afin de faciliter la résolution d'un problème. Ces approches peuvent être classées en trois groupes de stratégies pour faciliter la résolution. Les *approches prospectives* cherchent à anticiper l'état de la recherche, en essayant de minimiser l'évaluation de sous-espaces de recherche ne contenant pas de solutions. La cohérence locale et l'utilisation de choix heuristiques pour les variables et les valeurs sont des approches que nous présenterons. Les *approches rétrospectives* proposent un raisonnement sur le passé et cherchent à apprendre des choix qui ont été effectués durant la recherche. L'analyse des conflits et la génération de *nogoods*, un état dans lequel le RCD est incohérent, seront des aspects abordés dans cette section. Il existe également des traitements qui se prévalent des deux stratégies. Tirant des leçons des conflits afin de diriger la recherche.

1.3.2 Cohérence locale

Pour vérifier si une instanciation partielle est globalement cohérente, il faut évaluer le sous-espace de recherche associé, ce qui est une tâche difficile en pratique. Pour parfaire notre algorithme de recherche, il est donc possible d'imposer à l'instanciation partielle courante d'être localement cohérente. Si une instanciation partielle n'est pas localement cohérente, alors elle ne l'est pas globalement. Ceci permet de ne pas évaluer un sous-espace de recherche inutilement, c'est-à-dire d'*élaguer* l'arbre de recherche. Pour cela, nous présentons un aperçu des formes les plus célèbres de cohérences locales.

La cohérence d'arcs

La forme de cohérence locale la plus étudiée et celle qui s'est imposée comme une référence pour la résolution de problèmes de satisfiabilité est la *cohérence d'arcs* [MACKWORTH 1977]. Historiquement, le mot « arc » fait référence aux contraintes binaires d'un RCD, on parle donc de cohérence d'arcs AC pour les réseaux de contraintes binaires et de *cohérence d'arcs généralisée* GAC pour les réseaux de contraintes d'arité supérieure.

Définition 20 (Cohérence d'arcs) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD :

- un couple $(x, v) \in \mathcal{X} \times D_x$ est arcs-cohérent dans \mathcal{P} si et seulement si pour toute contrainte $c \in \mathcal{C}$, telle que $x \in \text{scp}(c)$, il existe un tuple $\tau \in \text{rel}(c) \cap \pi(\text{scp}(c))$ tel que $\tau[x] = v$;
- une variable $x \in \mathcal{X}$ est arcs-cohérente dans \mathcal{P} si et seulement si il existe au moins une valeur $v \in D_x$, telle que (x, v) soit arcs-cohérent ;
- \mathcal{P} est arcs-cohérent si et seulement si toute variable $x \in \mathcal{X}$ est arcs-cohérente ;
- \mathcal{P} est fermé par cohérence d'arcs si et seulement si tout couple $(x, v) \in \mathcal{X} \times D_x$ est arcs-cohérent.

Posons $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD. Si \mathcal{P} est fermé par cohérence d'arcs, alors quel que soit le choix d'affectation $x = v$ avec $x \in \mathcal{X}$ et $v \in D_x$, nous pouvons affirmer qu'aucune contrainte de $\mathcal{P}_{x=v}$ ne sera trivialement insatisfaite. Si un RCD est fermé par cohérence d'arcs, nous avons la garantie que la prochaine affectation choisie n'occasionera pas d'incohérence triviale dans le RCD. Ceci motive l'utilisation de la cohérence locale dans l'algorithme *Generate and Test*. Supposons maintenant que le couple (x, v) ne soit pas arcs-cohérent dans \mathcal{P} . Il existe au moins une contrainte insatisfaite dans $\mathcal{P}_{x=v}$. Par conséquent, l'affectation $x = v$ dans \mathcal{P} doit être évitée, et par conséquent v est supprimée du domaine de x . La cohérence d'arcs est une forme de cohérence locale *basée sur les domaines*.

L'algorithme 3 présente une approche simple pour obtenir la fermeture par cohérence d'arcs. L'ensemble des couples variable/valeur sont énumérés et placés dans une *file de révision*, c'est-à-dire un ensemble de couple variable/valeur pour lesquels il est nécessaire de vérifier s'ils satisfont la définition de cohérence d'arcs. Si un couple (x, v) n'est pas arcs-cohérent, alors v est réfuté du domaine de x . Or, la suppression de cette valeur peut avoir un impact sur les valeurs des variables voisines, et certaines ne vérifient peut être plus la cohérence d'arcs. Il est nécessaire de (re)vérifier la cohérence d'arcs pour les valeurs des variables voisines, qui sont donc placées dans la file de révision. Les vérifications se poursuivent jusqu'à atteindre un *point fixe*, c'est-à-dire jusqu'à ce qu'il n'y ait plus de suppressions de valeurs. Notons que s'il existe un domaine n'ayant aucune valeur arcs-cohérente alors le RCD est incohérent.

La complexité en temps de cet algorithme est $\mathcal{O}(enr^2d^{r+1})$. En particulier, la complexité en temps dans le pire des cas de cet algorithme pour un RCD binaire est $\mathcal{O}(end^3)$, et la complexité en espace dans le pire des cas est $\mathcal{O}(nd)$. Nous proposons, en fin de section, un aperçu des algorithmes optimaux en temps en $\mathcal{O}(erd^r)$ pour GAC et $\mathcal{O}(ed^2)$ pour AC.

Les figures 1.6 et 1.7 reposent sur le RCD de l'exemple 1, noté \mathcal{P} , et illustrent la cohérence d'arcs. La figure 1.6 correspond $\mathcal{P}_{|\mathcal{I}}$ avec $\mathcal{I} = \{(E_1, 1), (E_3, 2)\}$. Chaque contrainte décrit une relation de différence entre les valeurs du domaine des variables. Ce RCD n'est pas arcs-cohérent puisque quatre valeurs, représentées en rouge, ne vérifient pas la cohérence d'arcs. La contrainte entre E_1 et E_4 impose la différence des valeurs. Ainsi, il n'existe pas de tuples entre la valeur 1 du domaine de E_4 et une valeur du domaine de E_1 . De la même façon, la valeur 2 du domaine de E_4 n'admet pas de tuple support avec une valeur du domaine de la variable E_3 . En exploitant l'algorithme de fermeture par cohérence d'arcs, nous obtenons le RCD AC($\mathcal{P}_{|\mathcal{I}}$) fermé par cohérence d'arcs, illustré Figure 1.7.

Par la définition 14, une instantiation \mathcal{I}_X de $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ sur $X \subseteq \mathcal{X}$ est cohérente si et seulement si chaque contrainte impliquée par les variables de X sont satisfaites. Puisque la cohérence d'arcs vérifie que chaque couple variable/valeur ait un support dans chaque contrainte voisine, elle permet donc de déduire la cohérence d'une instantiation et, par conséquent, déduire si une instantiation complète est une

Algorithme 3: Fermeture par cohérence d'arcs

Entrées : $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$: un RCD
Sortie : GAC(\mathcal{P})

```

1  début
2  | file  $\leftarrow \{(x, v) : x \in \mathcal{X} \wedge v \in D_x\}$ 
3  | tant que (file  $\neq \emptyset$ ) faire
4  |   Choisir et Supprimer  $(x_i, v_{i_k}) \in$  file
5  |   si (Réviser( $\mathcal{P}, (x_i, v_{i_k})$ ) répond Oui) alors
6  |     si ( $D_{x_i} = \emptyset$ ) alors
7  |       | retourner  $\perp$ 
8  |     sinon
9  |       | file  $\leftarrow$  file  $\cup \{(x_j, v_{j_l}) : i \neq j \wedge \exists c \in \mathcal{C} : \{x_i, x_j\} \subseteq scp(c)\}$ 
10 |   retourner  $\mathcal{P}$ 

```

```

11 fonction Réviser( $\mathcal{P}, (x, v)$ ) : Booléen
12 début
13 | modification  $\leftarrow$  Non
14 | si (arcs-cohérence( $\mathcal{P}, (x, v)$ ) répond Non) alors
15 |   |  $D_x \leftarrow D_x \setminus \{v\}$ 
16 |   | modification  $\leftarrow$  Oui
17 | retourner modification

```

```

18 fonction arcs-cohérence( $\mathcal{P}, (x, v)$ ) : Booléen
19 début
20 | pour chaque ( $c \in \mathcal{C} : x \in scp(c)$ ) faire
21 |   | si ( $\nexists \tau \in rel(c) \cap \pi(scp(c)) : \tau[x] = v$ ) alors retourner Non
22 | retourner Oui

```

solution. La cohérence d'arcs est la forme de cohérence locale la plus faible permettant de déterminer la cohérence d'un RCD atomique. Puisqu'il est possible de répondre au problème de la cohérence d'un RCD atomique en temps polynomial dans la taille du problème, les RCD atomiques sont dits *traitables*.

L'approche présentée dans la recherche d'une solution vise à vérifier la cohérence d'une instantiation complète, nous pouvons intégrer la cohérence d'arcs à l'algorithme *Generate and Test*. De plus, nous souhaitons anticiper d'éventuelle incohérence locale durant la recherche. Pour cela, il suffit d'appliquer la fermeture par cohérence d'arcs afin d'obtenir un RCD $\mathcal{P}' = AC(\mathcal{P}|_{\mathcal{I}_X})$ qu'elle que soit \mathcal{I}_X . Il est donc garanti que, quelle que soit l'affectation $x = v$ choisie avec $x \in \mathcal{X} \setminus X$ et $v \in D_x$, nous avons une instantiation $\mathcal{I}_X \cup (x = v)$ de \mathcal{P} ne provoquant l'insatisfaction d'une contrainte. L'algorithme 4 intègre la fermeture par cohérence d'arcs dans l'algorithme *Generate and Test*. Celui-ci est un algorithme de recherche exhaustive avec *maintien de la cohérence d'arcs* ou MAC, pour *Maintain Arc-Consistency* [SABIN & FREUDER 1994]. La ligne 2 correspond à une phase de prétraitement du RCD, afin de garantir que le RCD soit fermé par cohérence d'arcs pour le premier choix. Après chaque affectation appliquée au RCD, la fermeture par cohérence d'arcs est maintenue (Ligne 14). Si le RCD qui en résulte n'est pas arcs-cohérent, alors il est inutile de continuer la recherche d'une solution sur ce sous-réseau puisque celui-ci n'est pas localement cohérent, un retour-arrière est donc nécessaire (Ligne 17). Dans le cas contraire, la recherche se poursuit (Ligne 15) jusqu'à atteindre un RCD atomique arcs-cohérent donc cohérent.

L'exemple 5 illustre l'application de MAC sur le problème de la salle de classe.

Exemple 5 Appliquons l'algorithme MAC au problème de l'exemple 1. Quelle que soit l'affectation $E_i = v$ choisie, avec $E_i \in \mathcal{X}$ et $v \in D_{E_i}$, la fermeture par cohérence d'arcs va déduire $E_j \neq v$ pour tout $E_j \in \mathcal{X}$ tel que $i \neq j$. Ceci impose à chaque étudiant de ne pas s'asseoir sur une chaise qui est

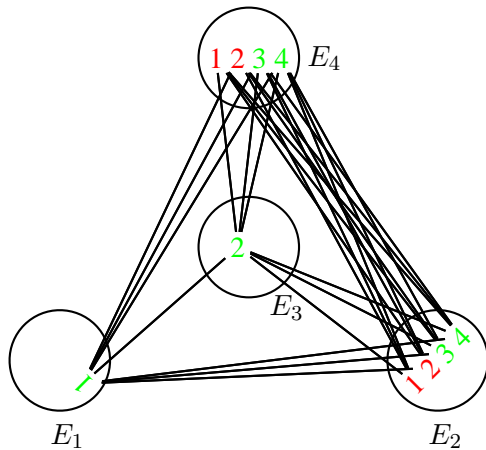


FIGURE 1.6 – $\mathcal{P}_{\mathcal{I}_{\{E_2, E_3\}}}$

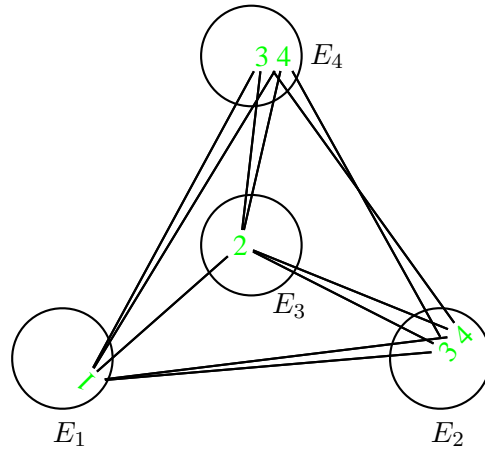


FIGURE 1.7 – $\mathcal{P}_{\mathcal{I}'_{\{E_1, E_3\}}}$

occupée. Si $n \leq d$ alors, l'un après l'autre, les étudiants pourront s'asseoir sur une des chaises libres. Le problème est donc satisfiable, et une solution a été trouvée sans qu'aucun étudiant n'ait à se relever, c'est-à-dire sans retour-arrière sur les choix effectués. Il suffit d'appliquer n fois l'algorithme GAC pour trouver une solution à ce problème. Ce problème est donc traitable en pratique et peut être résolu par un algorithme en temps polynomial dans le nombre de variables. Cependant, ceci est vrai uniquement pour $n \leq d$ pour l'algorithme MAC.

Remarquons qu'il est possible d'appliquer et de maintenir toute forme de cohérence locale pendant la recherche. Cependant, ceci est difficilement envisageable pour certaines formes de cohérence locale dont la complexité algorithmique est élevée. Il est impossible de prévoir avec certitude si une forme de cohérence locale sera pertinente pour la résolution d'un RCD, c'est-à-dire si l'élagage sera fort ou non. Notons que le prétraitement peut, lui aussi, être une forme de cohérence locale quelconque, et peut être une forme différente de celle maintenue durant la recherche.

Puisque la cohérence d'arcs est maintenue pendant la recherche d'une solution et donc appliquée à chaque décision, celle-ci suscite de nombreux travaux [BESSIÈRE 2006, LECOUTRE 2009]. L'algorithme 3 présenté est un algorithme de base, proche de l'algorithme AC3 [MACKWORTH 1977], bien que celui-ci place dans la file de révisions des couples variables/contraintes. L'inconvénient de ces approches est qu'elles ne sont pas réfléchies, dans le sens où chaque couple de la file de révisions est parcouru sans distinction, et quel que soit l'état du réseau de contraintes. Ainsi, l'algorithme AC4 [MOHR & HENDERSON 1986] et GAC4 [MOHR & MASINI 1988] atteint la complexité algorithmique optimale en temps dans le pire de cas pour un réseau de contraintes binaires en $\mathcal{O}(erd^r)$, en sauvegardant des informations telles que le nombre de supports pour un couple variable/valeur avec les autres variables. Ce qui permet d'affiner et d'accélérer les révisions. Cet algorithme est dit à « grain fin » car les révisions sont effectuées au niveau des valeurs, à l'opposé des approches proposant des révisions sur les contraintes. AC6 [BESSIÈRE 1994] permet de conserver la complexité optimale en temps en sauvegardant moins d'informations que AC4. AC2001 [BESSIÈRE & RÉGIN 2001] est une extension permettant d'affiner l'algorithme AC3.

De plus, dans la file de révision de AC3, une contrainte peut être révisée à partir de chacune des variables impliquées dans celle-ci. Afin d'éviter plusieurs traitements d'une même contrainte, de nombreux algorithmes intègrent la multidirectionnalité des contraintes. AC7 [BESSIÈRE *et al.* 1999] permet d'exploiter la bidirectionnalité en se basant sur AC6, et deux extensions de AC2001 exploitent partiellement

Algorithme 4: MAC

Entrées : $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$: un RCD
Sortie : Oui si \mathcal{P} est cohérent, Non sinon

```

1 début
2    $\mathcal{P}' \leftarrow \text{GAC}(\mathcal{P})$ 
3   si ( $\mathcal{P}' = \perp$ ) alors
4     | retourner Non
5   sinon
6     | retourner Recherche( $\mathcal{P}'$ )

```

7 fonction Recherche(\mathcal{P}) : **Booléen**

```

8 début
9   si ( $\mathcal{P}$  est atomique) alors
10    | retourner Oui
11  sinon
12    |  $x \leftarrow$  Choisir une variable dans  $\text{futures}(\mathcal{X})$ 
13    | pour chaque ( $v \in D_x$ ) faire
14      |  $\mathcal{P}' \leftarrow \text{GAC}(\mathcal{P}_{|x=v})$ 
15      | si ( $\mathcal{P}' \neq \perp$ ) et (Recherche( $\mathcal{P}'$ ) répond Oui) alors
16        | | retourner Oui
17    | retourner Non

```

avec AC3.2 ou complètement comme AC3.3 la bidirectionnalité [LECOUTRE *et al.* 2003]. Lecoutre et Hemery [LECOUTRE & HEMERY 2006] proposent d'intégrer la notion de supports résiduels aux algorithmes de cohérence d'arcs, c'est-à-dire pour chaque triplet contrainte/variable/valeur, le dernier support trouvé pour le couple variable/valeur dans la contrainte. Les algorithmes AC3 et AC3.2 intègrent la gestion de supports résiduels respectivement unidirectionnels avec AC3*r* ou multidirectionnel avec AC3*rm*. AC3*rm* se révélant être l'algorithme le plus efficace en pratique. Enfin, Boussemart *et al.* [BOUSSEMART *et al.* 2004b] définissent des stratégies de choix de révisions de la file de révisions à partir de critères heuristiques sur des caractéristiques des variables ou des contraintes. La diversité des algorithmes de cohérences d'arcs induit également une grande diversité des algorithmes de maintien de la cohérence d'arcs, notamment MAC2001/3.1 [BESSIÈRE *et al.* 2005] et MAC3*rm* [LECOUTRE *et al.* 2008].

La cohérence de chemins

La cohérence d'arcs permet d'affirmer, pour tout couple (x, v) et pour toute variable x' voisine de x , qu'il existe une valeur v' du domaine de x' telle que $\{(x, v), (x', v')\}$ est une instantiation partielle cohérente. Le raisonnement peut être étendu, en vérifiant qu'à partir de deux couples (x, v) et (x', v') et pour toute variable x'' voisine de x et x' , qu'il existe une valeur v'' du domaine de x'' telle que $\{(x, v), (x'', v'')\}$ et $\{(x', v'), (x'', v'')\}$ soient des instantiations partielles cohérentes. Cette forme de cohérence locale est appelée *cohérence de chemins*, notée PC pour *Path Consistency* [MONTANARI 1974].

Définition 21 (Cohérence de chemins) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD :

- $(x, v) \in \mathcal{X} \times D_x$ et $(x', v') \in \mathcal{X} \times D_{x'}$ sont chemins-cohérents si et seulement si, pour tout $x_i \in X$ telle que $x_i \neq x$ et $x_i \neq x'$, il existe une valeur $v_{i_k} \in D_{x_i}$ telle que $\{(x, v), (x_i, v_{i_k})\}$ et $\{(x_i, v_{i_k}), (x', v')\}$ soient des instantiations partielles cohérentes ;
- deux variables $x \in \mathcal{X}$ et $x' \in \mathcal{X}$ sont chemins-cohérente si et seulement si il existe deux valeurs $v \in D_x$ et $v' \in D_{x'}$ telles que (x, v) et (x', v') soient chemins-cohérents ;
- \mathcal{P} est chemins-cohérent si et seulement si tout couple $x \in \mathcal{X}$ et $x' \in \mathcal{X}$ est chemins-cohérent ;

- \mathcal{P} est fermé par chemin cohérence si et seulement tout couple $(x, v) \in \mathcal{X} \times D_x$ et $(x', v') \in \mathcal{X} \times D_{x'}$ est chemins-cohérent.

Historiquement, le terme « chemin » est utilisé au même sens qu'un chemin dans un graphe, dans lequel chaque couple variable/valeur décrit un nœud et chaque tuple support des contraintes décrit une arête entre ces nœuds. Cependant, Montanari a montré l'équivalence avec la définition 21.

L'algorithme 5 présente la fermeture par cohérence de chemins. Contrairement à la cohérence d'arcs, ce sont des ensembles de deux couples $\{(x, v), (x', v')\}$ qui sont placés dans une file de révisions. Deux couples sont chemins-cohérents si il est possible d'étendre chacun, pour chaque variable voisine de x et x' , à une instantiation de taille 2. Si le couple est chemins-cohérent, alors l'évaluation se poursuit. Dans le cas contraire, l'ensemble $\{(x, v), (x', v')\}$ ne peut être étendu à une solution. $\{(x, v), (x', v')\}$ est localement incohérent. Pour éviter de faire le choix de ces deux couples pour de futures affectations, il faut les interdire mutuellement. Par conséquent, un tuple conflit correspondant à $\{(x, v), (x', v')\}$ est créé. Si il existe une contrainte ne possédant plus de tuples supports, alors le RCD est incohérent. Dans le cas contraire, il faut vérifier si x et x' sont chemins-cohérentes avec leurs voisines, et ainsi de nouveaux couples sont ajoutés à la file de révision.

Algorithme 5: Fermeture par cohérence de chemins

Entrées : $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$: un RCD
Sortie : $\text{PC}(\mathcal{P})$

```

1  début
2  file  $\leftarrow \{\{(x, v), (x', v')\} : x, x' \in \mathcal{X}, v \in D_x, v' \in D_{x'}, x \neq x'\}$ 
3  tant que (file  $\neq \emptyset$ ) faire
4      Choisir et Supprimer  $\{(x, v), (x', v')\} \in \text{Queue}$ 
5      si (Réviser( $\mathcal{P}$ ,  $\{(x, v), (x', v')\}$ ) répond Oui) alors
6          si ( $c_{(x, x')} = \emptyset$ ) alors
7              retourner  $\perp$ 
8          sinon
9              file  $\leftarrow$  file  $\cup \{\{(x, v), (x_i, v_{i_k})\} : x_i \in \mathcal{X} \wedge v_{i_k} \in D_{x_i} \wedge x_i \text{ voisine de } x\} \cup$ 
                 $\{\{(x', v'), (x_i, v_{i_k})\} : x_i \in \mathcal{X} \wedge v_{i_k} \in D_{x_i} \wedge x_i \text{ voisine de } x'\}$ 
10     retourner  $\mathcal{P}$ 

```

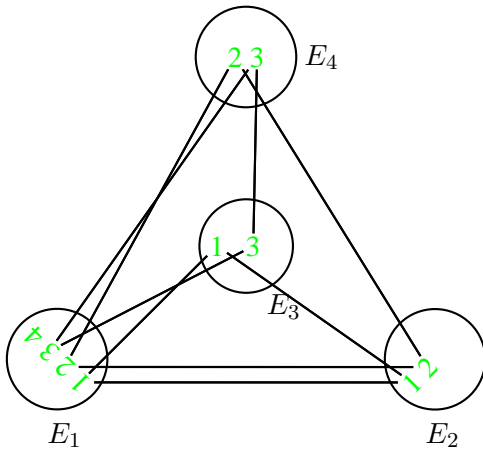
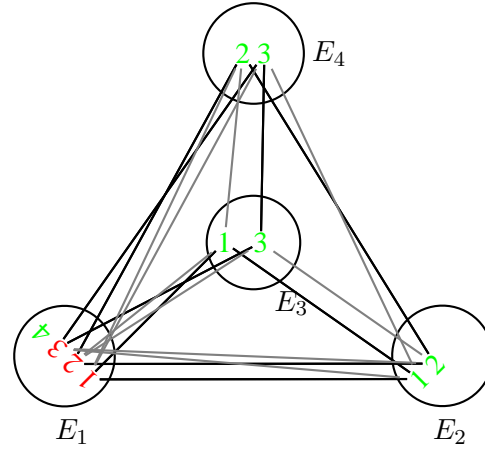
11 **fonction** Réviser(\mathcal{P} , $\{(x, v), (x', v')\}$) : Booléen
12 **début**
13 modification \leftarrow Non
14 si (chemins-cohérence(\mathcal{P} , $\{(x, v), (x', v')\}$) répond Non) alors
15 $\bar{c}_{(x, x')} \leftarrow \bar{c}_{(x, x')} \cup \{(v, v')\}$
16 modification \leftarrow Oui
17 retourner modification

18 **fonction** chemins-cohérence(\mathcal{P} , $\{(x, v), (x', v')\}$) : Booléen
19 **début**
20 pour chaque $(x_i \in \mathcal{X} : x_i \neq x \wedge x_i \neq x')$ faire
21 si ($\nexists v_{i_k} \in D_{x_i}$ telle que $(v, v_{i_k}) \in c_{(x, x_i)} \wedge (v_{i_k}, v') \in c_{(x_i, x')}$) alors
22 retourner Non
23 retourner Oui

La complexité en espace dans le pire des cas de cet algorithme coïncide avec l'énumération des instantiations de taille 2, soit $\mathcal{O}(n^2 d^2)$. Pour chacune, il est nécessaire de vérifier un chemin pour chaque couple variable/valeur des variables voisines. Pour cela, il faut trouver un tuple supports parmi les d^r

tuples. Chaque fois que l'on détermine qu'une instantiation de taille 2 n'est pas chemins-cohérente, ces voisines sont ajoutés. Dans le cas de réseaux de contraintes binaires, la complexité de cet algorithme est donc $\mathcal{O}(n^4 d^5)$.

Les figures 1.8 et 1.9 étendent l'exemple 1 afin d'illustrer la cohérence de chemins. La figure 1.8 présente le RCD \mathcal{P}' tel que $\mathcal{P}' \sqsubseteq \mathcal{P}$ avec \mathcal{P} le RCD modélisant le problème initial Exemple 1. Remarquons que les arcs expriment cette fois les tuples conflits entre les valeurs du domaine des variables. Sur \mathcal{P}' , certains étudiants n'ont pas l'autorisation de s'asseoir sur certaines chaises. De l'application de la fermeture par cohérence de chemins sur \mathcal{P}' résulte le RCD $PC(\mathcal{P}')$ présenté Figure 1.9. Les arcs grisés correspondent aux tuples conflits ajoutés par cohérence de chemins. Ainsi, si $E_1 = 3$ et $E_2 = 1$, alors il n'existe pas de valeurs de E_3 permettant de satisfaire la définition de la cohérence de chemins. Par conséquent, le tuple $(3, 1)$ de la contrainte $c_{(E_1, E_2)}$ est un conflit. De même, si $E_1 = 3$ et $E_2 = 2$, alors il n'existe pas de valeurs de E_4 permettant de satisfaire la définition de la cohérence de chemins. Par conséquent, tous les tuples entre E_1 et E_2 tel que $E_1 = 3$ sont des tuples conflits. La valeur 3 du domaine de E_1 n'a plus de supports dans E_2 , par conséquent cette valeur n'est pas arcs-cohérente. Les valeurs représentées en rouge sont les valeurs qui, à titre indicative, ne sont plus arcs-cohérentes après l'application de la fermeture par cohérence de chemins. Seuls les arcs grisés sont les effets de la fermeture par cohérence de chemins.


 FIGURE 1.8 – \mathcal{P}'

 FIGURE 1.9 – $PC(\mathcal{P}')$

Au même titre que la cohérence d'arcs, de nombreux algorithmes permettent de rendre un réseau de contraintes chemins-cohérent, et sont souvent inspirés, voire étendus, des algorithmes de cohérence d'arcs. Ainsi, PC2 est l'extension de AC3 [MACKWORTH 1977] permettant de maintenir une file de révision sur les triplets de variables. Au même titre que AC4, l'optimalité en temps dans le pire des cas est atteinte par deux extensions permettant de sauvegarder une liste de supports avec PC3 [MOHR & HENDERSON 1986]. PC5 est une extension de AC6, et PC5++ exploite la bidirectionnalité [SINGH 1996].

k -cohérence forte et $(i - j)$ -cohérence

Les cohérences d'arcs et de chemins peuvent être généralisées. La généralisation de ces approches est appelée k -cohérence, avec $k > 0$ [FREUDER 1978].

Définition 22 (k -cohérence) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD :

- un ensemble de $k - 1$ couples variable/valeur $\{(x_1, v_1), \dots, (x_{k-1}, v_{k-1})\}$ avec $x_i \in \mathcal{X}$, $v_i \in D_{x_i}$ et $x_i \neq x_j$ pour tout $1 \leq i, j \leq k - 1$, est k -cohérent si et seulement si, pour tout $x' \in \mathcal{X}$ avec

- $x' \neq x_i$ pour tout $1 \leq i \leq k-1$, il existe une valeur $v' \in D_{x'}$ telle que, $\{(x_i, v_i), (x', v')\}$ est une instantiation partielle cohérente pour tout $1 \leq i \leq k-1$;
- $X \subset \mathcal{X}$, avec $|X| = k-1$, est k -cohérent si et seulement si il existe un ensemble $\{(x_1, v_1), \dots, (x_{k-1}, v_{k-1})\}$ avec $x_i \in X$ et $v_i \in D_{x_i}$ pour tout $1 \leq i \leq k-1$ qui soit k -cohérente ;
- \mathcal{P} est k -cohérent si et seulement si tout ensemble de variables $X \subset \mathcal{X}$, avec $|X| = k-1$, est k -cohérent ;
- \mathcal{P} est fermé par k -cohérence si et seulement si pour tout $X \subset \mathcal{X}$ tel que $|X| = k-1$, tout ensemble $\{(x_1, v_1), \dots, (x_{k-1}, v_{k-1})\}$ avec $x_i \in X$ et $v_i \in D_{x_i}$ est k -cohérent.

Lorsqu'un ensemble de $k-1$ couples variable/valeur est k -incohérente, une contrainte d'arité ($k-1$) est créée et cet ensemble devient un tuple conflit. Les cohérences d'arcs et de chemins sont donc équivalentes respectivement à la 2-cohérence et 3-cohérence. La k -cohérence a été étendue à la k -cohérence forte.

Définition 23 (k -cohérence forte) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD. \mathcal{P} est fortement k -cohérent si et seulement il est k' -cohérent pour tout $1 \leq k' \leq k$.

L'idée d'étendre un ensemble de $k-1$ couple variable/valeur à 1 variable supplémentaire amène la question du procédé inverse. Est-il possible de déterminer une instantiation localement cohérente de taille 1 qui puisse être étendu à $k-1$ variables supplémentaires ? Cette forme de cohérence locale, nommée *inverse k -cohérence*, laisse entrevoir une généralisation sur la taille de l'instanciation initiale et son extension. Cette forme de cohérence s'appelle la (i, j) -cohérence, c'est-à-dire est-il possible d'étendre i affectations à une instantiation de taille j .

Définition 24 ((i, j) -cohérence) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un réseau de contraintes :

- un ensemble de i couples variable/valeur $\{(x_1, v_1), \dots, (x_i, v_i)\}$ avec $x_k \in \mathcal{X}$, $v_k \in D_{x_k}$ et $x_k \neq x_l$ pour tout $1 \leq k, l \leq i$, est (i, j) -cohérent si et seulement si, pour tout ensemble de variables $X \subset \mathcal{X}$ tel que $|X| = j$ et $x_k \notin X$ pour $1 \leq k \leq i$, il est possible de construire une instantiation partielle cohérente \mathcal{I}_X telle que $\{(x_k, v_k)\} \cup \mathcal{I}_X$ est une instantiation partielle cohérente pour tout $1 \leq k \leq i$;
- $X \subset \mathcal{X}$, avec $|X| = i$, est (i, j) -cohérent si et seulement si il existe un ensemble de i couples variable/valeur $\{(x_1, v_1), \dots, (x_i, v_i)\}$, avec $x_k \in X$, $v_k \in D_{x_k}$ et $x_k \neq x_l$ pour tout $1 \leq k, l \leq i$, qui soit (i, j) -cohérent ;
- \mathcal{P} est (i, j) -cohérent si et seulement si tout ensemble $X \subset \mathcal{X}$, tel que $|X| = i$, est (i, j) -cohérent ;
- \mathcal{P} est fermé par (i, j) -cohérence si et seulement toute ensemble de i couples variable/valeur $\{(x_1, v_1), \dots, (x_i, v_i)\}$, avec $x_k \in \mathcal{X}$, $v_k \in D_{x_k}$ et $x_k \neq x_l$ pour tout $1 \leq k, l \leq i$, est (i, j) -cohérent.

La k -cohérence est donc équivalente à la $(k-1, 1)$ -cohérence et l'inverse k -cohérence est équivalente à la $(1, k-1)$ -cohérence. Dans le cas général, pour appliquer la k -cohérence, il faut énumérer les instantiations de taille $k-1$, c'est-à-dire en d^{k-1} . Ce qui est une tâche difficile lorsque k est grand. Appliquer la k -cohérence forte est équivalent à développer un arbre de recherche de hauteur k à l'aide d'un parcours en largeur d'abord, où pour chaque niveau $1 \leq i \leq k$ la i -cohérence est vérifiée, afin de garantir la cohérence locale de chaque niveau. L'algorithme optimale est en $\mathcal{O}(n^k d^k)$ [COOPER 1989].

En pratique, les formes de cohérences locales au-delà de la 3-cohérence, forte ou non, sont très peu utilisées, notamment pour le maintien de la cohérence locale pendant la recherche. Comme énoncé précédemment, il est difficile de déterminer la pertinence du maintien de certaines formes de cohérence locale. De plus, une forme de (i, j) -cohérence impose de manipuler des tuples conflits de taille i lorsqu'une instantiation n'est pas (i, j) -cohérente. Une tâche qui s'avère être coûteuse en espace, notamment si le RCD

n'est pas i -cohérent. Cependant, il est possible d'imposer la fermeture par une forme de cohérence locale forte, de n'être effectuée qu'une seule fois en prétraitement du RCD.

(k, ϕ) -cohérence

La (k, ϕ) -cohérence est une généralisation de cohérences locales bien connues telles que la cohérence d'arcs de singletons SAC [DEBRUYNE & BESSIÈRE 1997] ou la cohérence d'arcs de tuples TAC [CHMEISS *et al.* 2008]. ϕ représente toute forme de cohérence locale qui peuvent être définie, tel que AC ou PC.

Définition 25 Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD :

- une instantiation \mathcal{I}_X de \mathcal{P} sur $X \subseteq \mathcal{X}$, avec $|X| = k$, est (k, ϕ) -cohérente si et seulement si $\phi(\mathcal{P}_{|\mathcal{I}_X}) \neq \perp$;
- $X \subseteq \mathcal{X}$, avec $|X| = k$, est (k, ϕ) -cohérent si et seulement si il existe une instantiation \mathcal{I}_X de \mathcal{P} qui soit (k, ϕ) -cohérente.
- \mathcal{P} est (k, ϕ) -cohérent si et seulement si tout ensemble de variables $X \subseteq \mathcal{X}$, avec $|X| = k$, est (k, ϕ) -cohérent.
- \mathcal{P} est (k, ϕ) -fermé si et seulement si pour tout ensemble de variables $X \subseteq \mathcal{X}$, avec $|X| = k$, toute instantiation \mathcal{I}_X de \mathcal{P} est (k, ϕ) -cohérente.

Pour schématiser cette approche, il suffit de construire toute instantiation partielle sur k variables distinctes, appliquer celle-ci au réseau de contraintes et vérifier la ϕ -cohérence du réseau de contraintes. Le principe est donc similaire au cadre affectations/propagations de la recherche exhaustive afin d'éliminer, *a priori*, les instantiations partielles de taille k globalement incohérentes. Elle permet ainsi de vérifier la cohérence locale au-delà du voisinage des variables de l'instanciation appliquée, ce qui diffère des techniques de cohérence locale présentées précédemment.

Dans le cas général, l'algorithme appliquant la (k, ϕ) -fermeture énumère les instantiations de taille k et applique la ϕ -cohérence. Si une instantiation n'est pas (k, ϕ) -cohérente, un tuple conflit de taille k est créé. Le procédé est réitéré jusqu'à obtenir un point fixe, c'est-à-dire qu'il n'y a plus de modifications dans le RCD. Pour illustrer la (k, ϕ) -cohérence, nous pouvons évoquer la $(1, \text{AC})$ -cohérence, plus connue sous le nom de *cohérence d'arcs de singletons*, notée SAC pour *Singleton Arc-Consistency*, présenté Algorithme 6. Pour chaque couple $(x, v) \in \mathcal{X} \times D_x$, si $\text{AC}(\mathcal{P}_{|x=v}) = \perp$ alors $x = v$ est une affectation globalement incohérente dans \mathcal{P} . Il convient donc de réfuter v du domaine de la variable x . Le procédé est réitéré jusqu'à atteindre un point fixe. La complexité algorithmique en temps dans le pire des cas de l'algorithme 6 est en $\mathcal{O}(n^2 d^2 \mathcal{O}(\text{AC}))$ et la complexité spatiale dans le pire des cas est $\mathcal{O}(nd)$.

Les algorithmes SAC sont également nombreux. Parmi les extensions, nous pouvons citer SAC – SDS [BESSIERE & DEBRUYNE 2008] et SAC3+ [LECOUTRE & CARDON 2005] qui exploitent incrémentalement la cohérence d'arcs de singletons en développant un arbre de recherche.

1.3.3 Choix heuristiques

Les techniques de cohérence locale sont un atout majeur pour répondre au problème de satisfiabilité. Néanmoins, il existe certaines catégories de problèmes sur lesquelles seules les formes de cohérences locales très fortes pourront être pertinentes. Cependant, la complexité algorithmique de telles approches est très élevée, voire aussi difficile que le problème de satisfiabilité lui-même. Il faut donc se concentrer sur d'autres approches pour améliorer l'algorithme MAC. L'exemple 6 fait mention d'un problème majeur : quels choix d'affectations faut-il effectuer pour évaluer l'arbre de recherche ?

Algorithme 6: Fermeture par Cohérence d'Arcs de Singletons

Entrées : $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$: un RCD
 Sortie : SAC(\mathcal{P})

```

1  début
2  |   fini ← Non
3  |   tant que (fini est Non) faire
4  |   |   fini ← Oui
5  |   |   file ←  $\{(x, v) : x \in \mathcal{X} \wedge v \in D_x\}$ 
6  |   |   tant que (file  $\neq \emptyset$ ) faire
7  |   |   |   Choisir et Supprimer  $(x, v) \in$  file
8  |   |   |   si  $(AC(\mathcal{P}_{|x,v}) = \perp)$  alors
9  |   |   |   |   Supprimer  $v$  de  $D_x$ 
10  |   |   |   |   fini ← Non
11  |   |   |   |   si  $(D_{x_i} = \emptyset)$  alors
12  |   |   |   |   |   retourner  $\perp$ 
13  |   |   retourner  $\mathcal{P}$ 
    
```

Exemple 6 À partir de l'exemple 1, considérons 50 étudiants et 50 chaises. Nous faisons évoluer ce problème puisque 5 étudiants à surveiller doivent être placés au premier rang composé de 4 chaises. Ce problème n'a trivialement pas de solutions puisqu'il n'y a pas assez de chaises au premier rang, malgré le fait qu'il y ait assez de chaises pour les 50 étudiants. Seule les techniques de vérification de la cohérence locale plus forte que la 5-cohérence peuvent montrer l'aspect incohérent de ce problème.

Ce problème peut pourtant être facilement divisé en deux sous-problèmes. Les élèves souhaitant se mettre devant forment le problème (5, 4) et le reste forme un second sous-problème (45, 46). En appliquant la recherche exhaustive, deux cas se présentent. Si les élèves du premier rang prennent places en premier, alors ils pourront « rapidement » déduire que le sous-problème (5, 4) est incohérent. S'ils sont choisis en dernier, 45 étudiants vont s'asseoir avant que le problème soit montré incohérent par les 5 autres.

L'algorithme MAC (Algorithme 4) présente le choix d'une variable parmi les variables futures et le parcours de chaque valeur du domaine de cette variable. Comment faire ce choix ? Quels sont les critères de choix qui permettent d'améliorer les algorithmes de recherche, sans connaître le sens du problème modélisé ? Ces deux lignes font référence à des *choix heuristiques* des variables et des valeurs du RCD permettant de guider la recherche.

Définition 26 (Heuristique de choix de variables) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD. Une heuristique de choix de variables $H_{\mathcal{X}}$ est une application de \mathcal{X} dans \mathbb{R} , associée à une relation d'ordre totale \leq_H sur \mathcal{X} définie, pour tout $x_i, x_j \in \mathcal{X}$, par : $x_i \leq_H x_j$ si et seulement si $H(x_i) \leq H(x_j)$.

La définition d'une heuristique de choix de valeurs est analogue à la définition 26, associant à chaque valeur du domaine d'une variable un nombre réel, et associé à une relation d'ordre totale sur chacune des valeurs. L'ordre \leq_H désigne une préférence, c'est-à-dire $x_i \leq x_j$ signifie « x_j est préférée à x_i » (où inversement suivant le choix effectué). Le choix de variables ou de valeurs se portent donc sur la variable ou la valeur préférée. Il est nécessaire d'avoir une relation d'ordre strict pour faire un choix de variables ou un choix de valeurs. Dans le cas général, les heuristiques peuvent ainsi être combinées, afin d'affiner le choix effectué ou départager les variables aussi préférée. Dans la suite, nous considérons qu'une fois l'heuristique H calculée, l'ordre \leq_H est établi en $\mathcal{O}(n^2)$ à l'aide d'un tri-rapide [HOARE 1961].

L'exemple 6 montre clairement que, pour un problème incohérent, il est nécessaire de choisir les variables d'un sous-problème incohérent en premier. En effet, si un problème est incohérent, alors il y

a une cause, qui peut être caractérisée comme un sous-problème du problème d'origine. Développer la recherche vers ce sous-problème, à proximité de la racine de l'arbre de recherche, permet de prouver « rapidement » que le problème est insatisfiable. Dans notre exemple, parcourir 4^5 instanciations complètes est suffisant pour prouver l'incohérence, alors que 50^{50} peuvent être parcourues dans le pire des cas. Cette stratégie de recherche est appelée l'*échec d'abord*, ou *fail-first* [HARALICK & ELLIOTT 1979].

Notre objectif pour caractériser l'heuristique de choix de variables est donc de préférer les variables du sous-problème incohérent. Cependant, dans le cas général, trouver le sous-problème incohérent est une tâche plus difficile que le problème CSP. L'heuristique H et l'ordre associée \leq_H doivent être décidées à chaque niveau de l'arbre. Une heuristique difficile à calculer pourrait rendre la résolution du problème difficile, mais permettrait peut-être de faire des choix appréciables. À l'inverse, une heuristique facile à calculer pourrait proposer de mauvais choix. Les heuristiques peuvent être classées en deux catégories : les heuristiques statiques et les heuristiques dynamiques. Les heuristiques statiques sont calculées avant l'évaluation de l'arbre de recherche, et l'ordre choisi est donc toujours le même. Une heuristique dynamique permet de proposer des choix en fonction de l'état de la recherche. Elles permettent d'adapter les choix à chaque instant de la recherche.

Heuristique de choix de variables

lexico La première heuristique, parmi les plus triviales, est l'ordre lexicographique, notée *lexico*. Chaque variable ou chaque valeur est définie par un nom unique, ce qui permet de décrire une relation d'ordre strict. Cette heuristique est très souvent utilisée pour départager les heuristiques associées à une relation d'ordre non strict. Puisque les noms ne changent pas en cours de recherche, cette heuristique est statique. La valeur associée à chaque variable est donc calculée en $\mathcal{O}(n)$.

random Tout aussi triviale, le choix aléatoire d'une variable, notée *random*. Une variable est choisie et considérée comme préférée à tout autre. Puisque *random* est calculée en $\mathcal{O}(1)$, elle peut s'avérer plus efficace que d'autres approches « intelligente », notamment pour certains problèmes difficiles pour lequel l'ordre du choix des variables n'est pas important. L'inconvénient de cette approche est qu'elle est non déterministe. Sur un même problème et avec les mêmes algorithmes de recherche, les résultats peuvent être totalement différents.

dom La première heuristique dynamique utilise la taille du domaine, notée *dom* [HARALICK & ELLIOTT 1979]. Elle est définie par l'application $dom(x) = -|D_x|$. Notons que $dom(x)$ est toujours négatif, ce qui permet de préférer les variables ayant les domaines les plus petits, par notre définition de \leq_{deg} . L'idée derrière cette stratégie est d'avoir un nombre de branchement (ou facteur de branchement) faible en haut de l'arbre de recherche. S'il y a maintien de la cohérence locale dans le réseau de contraintes, la taille du domaine est modifiée dynamiquement, en fonction des choix précédents. Dans le pire des cas, *dom* est calculée en $\mathcal{O}(n)$.

ddeg L'analyse structurelle d'un RCD peut conduire à la définition d'une autre heuristique, s'appuyant sur le nombre de contraintes impliquant une variable, notée *deg*. Bien qu'elle puisse être calculée de manière statique, cette heuristique est plus généralement utilisée de manière dynamique, notée *ddeg* [DECHTER & MEIRI 1989]. Ainsi, $ddeg(x) = |\bigcup\{c \in \mathcal{C} : x \in scp(c) \wedge |futures(scpc)| > 1\}|$. Choisir les variables en fonction du degré permet de réduire le nombre de contraintes à traiter. Dans le pire des cas, l'heuristique *ddeg* est calculée en $\mathcal{O}(en)$.

wdeg Pour détecter des sous-problèmes incohérents afin de diriger la recherche, il est possible d'utiliser les algorithmes de cohérence locale. L'heuristique *wdeg* [BOUSSEMART *et al.* 2004a], pour *weighted degree*, requiert une légère adaptation de l'algorithme de cohérence locale. Un poids w_c est défini pour chaque contrainte $c \in \mathcal{C}$, tel que $w_c = 1$ avant la recherche. Lorsque le problème est localement incohérent, il existe une ou plusieurs contraintes décrite par la relation vide, et le poids de chacune de ces contraintes est incrémenté. Au cours de la recherche, plus les contraintes ont été pondérées, plus elles ont été impliquées dans les incohérences locales. Cette heuristique repose donc sur la probabilité que les contraintes impliquées dans une incohérence lors de la fermeture par cohérence locale fassent partie d'un sous-problème incohérent. Elles permettent alors l'approximation d'un sous-problème incohérent dans le RCD. L'heuristique *wdeg* est également fonction du degré des variables, puisque si la variable est la seule non affectée dans une contrainte, le degré n'est pas comptabilisé. Celle-ci est définie par $wdeg(x) = \sum_{c \in \mathcal{C}: x \in scp(c) \wedge |future(scpc)| > 1} w_c$. Elle est calculée en $\mathcal{O}(en)$.

Combinaison heuristiques Comme énoncé précédemment, certaines heuristiques peuvent être utilisées pour départager des variables aussi préférées. Pour cela, on utilise l'opérateur \oplus . Par exemple, $dom \oplus ddeg$ signifie la variable est choisie à l'aide de *dom* puis les variables « également préférées » sont départagées par *ddeg*.

Les combinaisons d'heuristiques sont, en pratique, plus efficaces que les heuristiques présentées seules. L'heuristique *dom/deg* [BESSIÈRE & RÉGIN 1996] et *dom/ddeg* [SMITH & GRANT 1998] sont des combinaisons des heuristiques permettant de minimiser le domaine des variables tout en maximisant le degré. Elles sont, en moyenne, plus efficaces que les heuristiques *dom* ou *ddeg* seules. L'heuristique *dom/wdeg*, qui choisit une variable en minimisant son domaine et maximisant le degré d'incohérence, permet d'obtenir de très bons résultats en pratique. Elle est notamment utilisée pour l'approximation de sous-problèmes incohérents, et correspond à notre description précédente basée sur une stratégie d'échec d'abord. Cette heuristique est illustrée dans l'exemple 7.

Exemple 7 Revenons sur l'exemple 6. Au cours de la recherche, les contraintes entre ces 5 étudiants vont être impliquées dans l'incohérence du sous-problème (5, 4). Par conséquent, le degré d'incohérence w_c de chaque contrainte va augmenter et les variables choisies en priorité seront parmi ces 5 variables après chaque retour-arrière. Si ces 5 étudiants s'installent après leurs 45 camarades, ceux-ci vont progressivement faire remonter leur sous-problème dans l'arbre de recherche. Il suffit donc, dans le pire des cas de prouver $(1 + \sum_{i=1}^{46} i)$ fois l'incohérence de ce sous-problème afin de démontrer l'incohérence du problème. Avec une heuristique classique, l'incohérence de ce sous-problème devrait être prouvée 46^{45} fois dans le pire des cas.

Heuristique de choix de valeurs

Les algorithmes de recherche s'appuient moins sur les heuristiques de choix de valeurs, notamment du fait du coût pour les calculer. Les heuristiques élémentaires telles que *lexico* et *random* sont également définies, et sont très souvent utilisées. Parmi les heuristiques les plus connues, *conflict* associe à chaque valeur le nombre de conflits dans lesquels celle-ci apparaît. Comme pour *wdeg*, cette heuristique s'appuie sur un raisonnement statistique. La probabilité pour qu'une valeur fasse partie d'un sous-problème incohérent est plus forte lorsque cette valeur est impliquée dans beaucoup de conflits. À l'inverse, minimiser le nombre de conflits [FROST & DECHTER 1995] suggère que la valeur puisse apparaître plus facilement dans une solution. L'inconvénient de cette approche est qu'elle nécessite de maintenir le nombre de tuples dans lequel une valeur apparaît. Par conséquent, le calcul d'une telle heuristique requiert le parcours des tuples de chaque contrainte dans laquelle la valeur peut être impliquée. $conflict(v \in D_x) = \bigcup \{\tau \in c : c \in \mathcal{C} \wedge x \in scp(c) \wedge \tau[x] = v\}$. Pour calculer l'heuristique associée

aux valeurs du domaine d'une variable, la complexité algorithmique dans le pire des cas est $\mathcal{O}(ed^r)$. Ceci en fait une heuristique difficile à calculer, et nécessite également la mise à jour des contraintes à chaque point de décisions de la recherche. D'autres heuristiques sont adaptées des heuristiques utiliser pour la résolution du problème SAT [JEROSLOW & WANG 1990] à la résolution du problème CSP [LECOUTRE *et al.* 2007].

Exploiter les heuristiques

L'exemple 7 fait mention de la pondération d'un sous-problème incohérent, permettant de rapprocher celui-ci de la racine de l'arbre. Cependant, faire remonter un sous-problème incohérent peut être une étape très « lente ». Pour l'approche par maintien de la cohérence d'arcs (Algorithme 4), cela peut être dû à l'évaluation de chaque valeur du domaine d'une variable à chaque niveau de l'arbre.

Pour palier à ce problème, l'algorithme 7 se propose de donner plus d'importance aux heuristiques [HWANG & MITCHELL 2005]. Dans celui-ci une variable puis une valeur de son domaine sont choisies par critère heuristique. À la différence de l'algorithme 4, l'ensemble des valeurs du domaine n'est pas évalué avant un retour-arrière. L'affectation est appliquée au RCD et la fermeture par cohérence locale est appliquée. Si celui-ci n'est pas localement incohérent, alors la recherche se poursuit. Sinon, le couple (x, v) est réfuté, et la cohérence locale est de nouveau vérifiée. Quel que soit \mathcal{P} un RCD, si $\mathcal{P}_{|x=v}$ et $\mathcal{P}_{|x \neq v}$ sont incohérents, alors \mathcal{P} est incohérent et un retour-arrière doit être effectué. Néanmoins, si le RCD issu de la réfutation et de la fermeture par cohérence locale n'est pas localement cohérent, alors chacune des étapes est réitérée. Ainsi, un nouveau choix de variable et de valeur est effectué après la réfutation d'une valeur. Cet algorithme est un algorithme de recherche avec *branchements binaires*, ou *binary branching* en anglais, avec maintien de la cohérence d'arcs. Celui-ci simule des branchements de largeur 2 à chaque point de choix d'une variable : la décision positive pour un couple (x, v) et, si celle-ci n'est pas cohérente, la décision négative pour ce couple. Par analogie, l'algorithme de recherche exhaustive 4 est à *branchements dway* pour *branchements de largeur d*, ou *dway branching* en anglais, puisque les branchements sont effectués sur toutes les valeurs du domaine des variables choisies.

Considérons $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD et l'algorithme MAC avec branchement *dway*. Lorsque le choix de la variable se porte sur $x \in \mathcal{X}$, alors la première valeur $v \in D_x$ est évaluée. Si celle-ci est globalement incohérente, elle doit être supprimée, c'est-à-dire réfutée. Cette opération va être implicitement réalisée en affectant la deuxième valeur $v' \in D_x \setminus \{v\}$ à la variable x . Cependant, appliquer l'affectation de $x = v'$ suppose que la variable courante x est toujours préférée. Or, la modification de la microstructure du RCD par maintien de la cohérence locale peut contredire ce choix si l'heuristique était recalculée. Dans le cas d'une stratégie à branchements binaires, le choix d'une variable et d'une valeur est proposé de nouveau après chaque réfutation. Cette approche exploite donc beaucoup plus les choix heuristiques. Cependant, cela ne signifie pas que l'approche avec branchements binaires est meilleure que l'approche avec branchements de largeur d . En contre-partie, si les heuristiques ne sont pas adaptées au problème, alors l'impact des mauvais choix sera plus important dans le cas d'un branchement binaire. Remarquons néanmoins que le choix de l'affectation du premier couple variable/valeur peut être déterminant. En commençant l'évaluation de l'arbre de recherche, les heuristiques peuvent être complètement inefficaces. Les domaines, le degré ou encore les poids ont des valeurs initiales, qui ne sont pas adaptés à ce stade de la recherche. Pour l'heuristique *dom/wdeg*, par exemple, le premier choix ne dirige pas la recherche vers les conflits mais vers le domaine le plus petit.

Pour palier aux lacunes de la stratégie de branchements *dway* et améliorer la stratégie de branchements binaires, il est possible d'utiliser une autre approche permettant la mise en avant des heuristiques : les *redémarrages* [GOMES *et al.* 2000]. Cette approche consiste simplement à arrêter l'évaluation de l'instanciation courante et à reprendre le problème depuis la racine de l'arbre de recherche. Les valeurs globalement incohérentes et les pondérations appliquées en cours de recherche, par exemple celle

Algorithme 7: MAC avec branchement binaire

Entrées : $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$: un RCD
Sortie : Oui si \mathcal{P} est cohérent, Non sinon

```

1 début
2    $\mathcal{P}' \leftarrow \text{GAC}(\mathcal{P})$ 
3   si ( $\mathcal{P}' = \perp$ ) alors
4     | retourner Non
5   sinon
6     | retourner Recherche( $\mathcal{P}'$ )

```

```

7 fonction Recherche( $\mathcal{P}$ ) : Booléen
8 début
9   si ( $\mathcal{P}$  est atomique) alors
10    | retourner Oui
11  sinon
12     $x \leftarrow$  Choisir une variable dans futures( $\mathcal{X}$ )
13     $v \leftarrow$  Choisir une valeur dans  $D_x$ 
14     $\mathcal{P}' \leftarrow \text{GAC}(\mathcal{P}_{|x=v})$ 
15    si ( $\mathcal{P}' \neq \perp$ ) et (Recherche( $\mathcal{P}'$ ) répond Oui) alors
16      | retourner Oui
17    sinon
18      |  $\mathcal{P} \leftarrow \text{GAC}(\mathcal{P}_{|x \neq v})$ 
19      | si ( $\mathcal{P} = \perp$ ) alors
20        | retourner Non
21      | sinon
22        | Aller Ligne 9

```

utilisée pour *wdeg*, sont conservées. Cela permet de recommencer la recherche avec certaines connaissances, de démarrer l'évaluation par un sous-problème plus pertinent. Appliquer des redémarrages à la recherche évite les évaluations infructueuses de sous-problèmes profondément dans l'arbre de recherche, appelés *phénomène heavy-tail*. Cette approche suscite un grand intérêt, cependant elle est fortement dépendante de la stratégie utilisée pour les redémarrages. En pratique, les redémarrages sont fonctions du nombre de retours-arrière, donc d'incohérence locale, effectué. La première évaluation s'arrête alors rapidement, après quelques retours-arrière. L'utilisation des redémarrages permet donc d'atténuer le mauvais choix du premier couple variable/valeur en initialisant les pondérations de certaines heuristiques, tel que *dom/wdeg*

1.4 Conclusion

Dans ce chapitre, nous avons défini les notions de base permettant de répondre au problème CSP pour les Réseaux de Contraintes Discrets. Les RCD constituent un outil simple et compact pour la modélisation de problèmes à domaine fini. En exploitant les notions de la théorie des graphes, de nombreuses approches ont conduit à définir des outils algorithmiques efficaces pour décider de la cohérence des problèmes modélisés. L'approche la plus exploitée consiste à développer un arbre de recherche, afin de parcourir exhaustivement l'espace de recherche du problème, et de vérifier si une solution existe. Pour réduire cet espace de recherche, différentes formes de cohérences locales ont été définies. La cohérence d'arcs est la plus exploitée. Elle offre un outil rapide et efficace pour maintenir la cohérence locale en cours de recherche. Certaines formes de cohérence locale, permettant un élagage plus fort de l'espace de recherche mais plus difficile à calculer, sont exploitées en prétraitement du RCD, dans le but de faci-

liter la recherche. Cependant, exploiter la cohérence locale ne suffit pas à faciliter la résolution de tout problème. Il est nécessaire de faire de bons choix d'affectations pour ne pas parcourir un sous-espace de recherche ne contenant pas de solutions. Une analyse structurale du RCD peut être utilisée pour définir des critères heuristiques de choix de variables et de valeurs. D'autres approches consistent à identifier des sous-problèmes souvent impliqués dans l'incohérence locale du réseau de contraintes, potentiellement incohérents. En commençant la recherche pas ces sous-problèmes, il est possible de prouver rapidement l'incohérence du problème. Diriger la recherche vers les conflits montre les résultats les plus intéressants en pratique sur de nombreux problèmes. Afin d'exploiter au mieux ces choix heuristiques, le choix de l'algorithme de recherche se porte vers le maintien de la cohérence d'arcs avec branchements binaires. En intégrant une stratégie de redémarrage pour profiter rapidement des choix heuristiques, ces algorithmes constituent les outils les plus efficaces pour décider de la cohérence de RCD. De nombreuses améliorations peuvent être mentionnées. *L'analyse de conflits* consiste à déterminer la source de l'incohérence locale d'un RCD après affectation et maintien de la cohérence locale. Pour exploiter l'analyse de conflits, *l'apprentissage* permet de sauvegarder des instanciations globalement incohérentes, afin d'éviter de les reconsidérer durant la recherche. D'autres approches se concentrent sur une analyse structurale du RCD afin de détecter des formes de symétries ou de décomposer le problème en sous-problème afin de faciliter la recherche d'une solution.

De nombreuses applications logicielles, appelées *solveur*, ont été développées afin de décider de la cohérence des RCD et font l'objet de compétitions internationales. Parmi ces solveurs, nous pouvons citer Abscon que nous utiliserons pour développer et étudier nos approches [LECOUTRE & TABARY 2008] et qui figure parmi les meilleurs solveurs génériques. Abscon est un solveur générique écrit en langage JAVA. Il propose un catalogue très fourni d'algorithmes de recherche et de techniques d'inférences, mais permet également de développer de nouveaux outils pour répondre au problème CSP.

Chapitre 2

Formalismes qualitatifs et réseaux de contraintes qualitatives

Sommaire

2.1	Introduction	35
2.2	Les formalismes qualitatifs	37
2.2.1	Définition générale	37
2.2.2	Opérations sur les relations	38
2.2.3	Quelques formalismes qualitatifs	39
2.3	Les réseaux de contraintes qualitatives	45
2.3.1	Définition générale	45
2.3.2	Problème de satisfiabilité de contraintes qualitatives	47
2.3.3	Cohérence locale	48
2.3.4	Algorithme de recherche et techniques d'inférence	50
2.3.5	Les applications	54
2.4	Quelques fragments traitables de l'algèbre des intervalles	54
2.4.1	Relations pointisables	54
2.4.2	Relations ORD-Horn	56
2.4.3	Relations convexes	58
2.4.4	Relations préconvexes	60
2.5	Conclusion	63

2.1 Introduction

En fonction des connaissances à modéliser, il est nécessaire d'adapter l'outil à utiliser. Pour certaines connaissances définies sur des domaines non finis, le raisonnement quantitatif peut être limité en pratique (modélisation complexe, problèmes de mémoire, *etc.*). Dans le but de décider de la cohérence de connaissances temporelles ou spatiales, ont été définis les *formalismes qualitatifs* permettant de les modéliser. Les formalismes qualitatifs s'attachent à l'aspect qualitatif des informations et non à l'aspect quantitatif, au sens numérique. Un formalisme qualitatif décrit les positions relatives entre des entités temporelles ou spatiales - telles que des actions, des événements, des régions dans l'espace - dont le domaine de définition peut être infini et dense. Par exemple, un intervalle de temps sur une droite dense, une région dans un plan cartésien, *etc.* Les positions relatives entre ces entités décrivent un ensemble fini de relations entre les entités, appelées *relations atomiques*. Par exemple, un intervalle de temps se déroule *avant*

un autre, ou une région d'un plan *intersecte* une autre. Les connaissances temporelles ou spatiales sont ainsi modélisées par un ensemble d'informations qualitatives mettant en relation les entités entre elles, en faisant abstraction de l'aspect quantitatif.

Dans les années 1980, pour la résolution de problèmes difficiles à représenter dans le raisonnement quantitatif, telles que la compréhension du langage naturelle ou l'ordonnement temporelle, Allen imagine un formalisme qualitatif dédié à la représentation d'informations temporelles : *l'algèbre des intervalles* [ALLEN 1981, ALLEN 1983]. Pour ce formalisme qualitatif, les entités considérées sont des intervalles définies sur une droite représentant le temps. En considérant les positions relatives entre deux intervalles d'une droite, il est possible de définir treize relations binaires, telles que tout couple d'intervalles est mis en relation par l'une d'elles. Par exemple, la phrase « Anatole fait la vaisselle pendant que Béatrice regarde un match de football » exprime des connaissances temporelles qualitatives. Les entités temporelles sont les activités « Anatole fait la vaisselle » et « Béatrice regarde un match de football », et la relation entre ces entités exprime que l'action d'Anatole se déroule « pendant » celle de Béatrice. Il n'y a ni notion de durée des activités, ni précision quantitative sur le début ou la fin des activités pour caractériser les événements. Le cadre qualitatif ne s'attache pas à figer les entités, mais à proposer une relation atomique entre les entités, et chaque relation atomique décrit une infinité de positions absolues. Cependant, la modélisation des connaissances amène à faire un choix parmi un ensemble de relations atomiques proposées entre les entités, appelé *relation complexe*. Ainsi, l'activité d'Anatole peut « commencer *ou* terminer » ou encore « ne pas commencer *ou* ne pas terminer » l'activité de Béatrice. Une contrainte implicite des connaissances est que les relations atomiques sont complètes et mutuellement exclusives. Par conséquent, les entités ne peuvent être mises en relation que par une et une seule des relations atomiques qui composent la relation complexe. Dans le cadre de l'algèbre des intervalles, l'action d'Anatole peut « commencer *ou* terminer » l'activité de Béatrice mais pas « commencer *et* terminer ». Ce formalisme constitue un outil de représentation simple et synthétique des données temporelles, ce qui contribue au succès de ce modèle, qui est toujours utilisé de nos jours dans de nombreux domaines.

De nombreux formalismes étendent la représentation des intervalles du formalisme d'Allen. *L'algèbre des n -pavés* [BALBIANI *et al.* 2003a] décrit des pavés dans un espace euclidien de dimension n ainsi que les positions relatives entre ceux-ci. Ce formalisme qualitatif permet de généraliser le formalisme d'Allen qui coïncide avec l'algèbre des 1-pavés. Balbiani et Osmani [BALBIANI & OSMANI 2000] s'inspirent du cadre d'Allen et proposent, par exemple, l'*algèbre des arcs cycliques* permettant la représentation d'arcs orientés sur un cercle. Mais les formalismes qualitatifs proposent la représentation d'informations de natures diverses, autre que des entités intervalles de temps. Vilain et Kautz [VILAIN & KAUTZ 1986, VILAIN *et al.* 1990] définissent un formalisme qualitatif pour la représentation d'entités ponctuelles sur une droite temporelle : *l'algèbre des points*. Ligozat [LIGOZAT 1998b] propose l'*algèbre des points cardinaux* décrivant les points dans un espace cartésien à deux dimensions, mis en relation par les relations cardinales. Balbiani et Condotta [BALBIANI & CONDOTTA 2002] généralise cette algèbre à un espace cartésien à n dimensions, avec $n > 2$. Pour la représentation d'informations spatiales, Randell et Cohn [RANDELL & COHN 1989, RANDELL *et al.* 1992] définissent le formalisme RCC, pour *Region Connection Calculus*. RCC permet la description de positions relatives entre des régions d'un espace topologique. Les régions sont décrites par des formules topologiques, et une même région peut être composée de plusieurs parties déconnectées. Ce formalisme qualitatif est l'un des plus utilisés pour la représentation d'informations spatiales.

Pour modéliser les connaissances temporelles ou spatiales et pour vérifier la cohérence de celles-ci, des réseaux de contraintes particuliers sont utilisés dans le cadre des formalismes qualitatifs, appelés *réseaux de contraintes qualitatives* (noté RCQ en abrégé). On dit alors que le RCQ est construit sur un formalisme qualitatif en particulier. Les variables du réseau de contraintes correspondent aux entités temporelles ou spatiales et le domaine des variables est fonction du formalisme qualitatif utilisé. Les contraintes d'un RCQ, appelée *contraintes qualitatives*, sont constituées d'un ensemble de positions

relatives entre ces entités à l'aide d'un ensemble de *relations atomiques*. Puisque le domaine des entités peut être infini, énumérer ou parcourir l'espace de recherche en temps fini est impossible. Pour palier à ce problème, il est nécessaire de se concentrer sur l'aspect qualitatif des informations et d'exploiter les propriétés des formalismes qualitatifs utilisés. Ainsi, l'algorithme de recherche exhaustive est adapté et, plutôt que de choisir une valeur du domaine de chaque variable, le choix se porte sur une et une seule des positions relatives proposées par chacune des contraintes entre les entités. Cela permet de construire un RCQ atomique, appelé *scénario*, dans lequel chaque contrainte ne décrit qu'une seule position relative. Pour de nombreux formalismes qualitatifs, il est possible de répondre au problème de la cohérence d'un RCQ à l'aide d'un scénario. Le problème de décision consistant à décider de la satisfiabilité d'un RCQ est le problème CSP qualitatif. Pour de nombreux formalismes qualitatifs, le problème CSP qualitatif est un problème de décision NP-complet, dans le cas général. Cependant, de nombreux travaux exploitent la restriction de certains formalismes qualitatifs pour mettre en évidence des classes particulières de relations complexes, appelées *classes traitables*, telles qu'il est possible de répondre à la cohérence de tout RCQ défini à partir de l'une de ces classes en temps polynomial dans la taille du problème. Pour définir de telles classes, certaines approches visent à réduire le formalisme qualitatif étudié à un formalisme qualitatif plus élémentaire, tel que l'algèbre des points. D'autres tendent à représenter les formalismes qualitatifs en logique propositionnelle [NEBEL & BÜRCKERT 1995], afin d'extraire des fragments traitables. Les RCQ bénéficient ainsi d'un cadre théorique permettant d'exploiter les propriétés des formalismes qualitatifs dans le but de définir des outils algorithmiques efficaces pour répondre à la cohérence de connaissances temporelles ou spatiales [BESSIÈRE *et al.* 1996].

Dans ce chapitre, nous étudions la représentation d'informations temporelles et spatiales à l'aide des RCQ. Pour cela, nous définissons formellement les formalismes qualitatifs, puis nous présentons quelques formalismes qualitatifs bien connus tels que l'algèbre des points, l'algèbre des intervalles ou encore l'algèbre des arcs cycliques. Nous définissons ensuite les RCQ et présentons les techniques algorithmiques utilisées pour répondre à la cohérence des RCQ. Enfin, nous proposons un aperçu des classes traitables de l'algèbre des intervalles.

2.2 Les formalismes qualitatifs

2.2.1 Définition générale

Les formalismes qualitatifs permettent la description d'informations temporelles ou spatiales. Pour cela, il faut pouvoir décrire le domaine des entités, mais également les relations possibles entre ces entités. Dans notre étude, nous nous intéressons principalement aux formalismes qualitatifs binaires, c'est-à-dire les formalismes qualitatifs dont les relations atomiques considérées sont binaires. Dans la suite, le terme « formalisme qualitatif » mentionnera implicitement les formalismes qualitatifs binaires, et nous préciserons le cas échéant.

Définition 27 (Formalisme qualitatif) *Un formalisme qualitatif \mathcal{F} est défini par :*

- le domaine des entités \mathbb{D} ;
- un ensemble fini de relations atomiques binaires, noté \mathcal{B} , tel que \mathcal{B} soient une partition de \mathbb{D}^2 .

Le domaine des entités temporelles et spatiales, noté \mathbb{D} , décrit des intervalles de temps, des points sur un repère cartésien, des régions dans un espace cartésien, *etc.* Il représente communément un ensemble infini d'éléments. Chaque position relative entre deux entités est décrit par une relation atomique r telle que $r \subseteq \mathbb{D}^2$. L'ensemble des relations atomiques, noté \mathcal{B} , décrit donc l'ensemble des positions relatives entre les entités sur \mathbb{D}^2 . Les relations atomiques sont complètes et mutuellement exclusives. Par conséquent, pour tout $(x, y) \in \mathbb{D}^2$, il existe une et une seule relation $r \in \mathcal{B}$ telle que $(x, y) \in r$. L'ensemble \mathcal{B} forme donc une partition de \mathbb{D}^2 [LIGOZAT & RENZ 2004].

Les entités peuvent être mises en relation par des *relations complexes*. Une relation complexe est définie par un ensemble $R = \{r_1, \dots, r_k\}$ où $r_i \in \mathcal{B}$ pour tout $i \in [1 ; k]$. Pour deux entités $x, y \in \mathbb{D}$, nous noterons $(x, y) \in R$ le fait qu'il existe une relation atomique $r \in R$ telle que $(x, y) \in r$. L'ensemble des relations complexes d'un formalisme qualitatif est donc désigné par l'ensemble des parties de \mathcal{B} , que nous noterons $2^{\mathcal{B}}$ dans le cas général. Notons qu'il est courant dans le cadre qualitatif de confondre la relation complexe $R = \{r_1, \dots, r_k\}$ et la relation binaire $r_1 \cup \dots \cup r_k$. Ainsi, $(x, y) \in R$ avec $R = \{r_1, \dots, r_k\}$ si et seulement si $(x, y) \in r_1 \cup \dots \cup r_k$.

2.2.2 Opérations sur les relations

De l'ensemble $2^{\mathcal{B}}$, nous pouvons dégager des relations remarquables pour chaque formalisme qualitatif. La relation *identité*, notée usuellement id , est une relation atomique précisant l'identité entre entités (relation réflexive et symétrique), telle que $\text{id} \in \mathcal{B}$.

Définition 28 (Relation identité) Soit \mathcal{F} un formalisme qualitatif. La relation identité id est définie par $\text{id} = \{(x, y) \in \mathbb{D}^2 : x = y\}$.

La *relation totale*, notée ψ , est la relation complexe permettant de spécifier que toutes les positions relatives entre deux entités sont admises. Par conséquent, $\psi = \mathcal{B}$. Aussi, une contrainte naturelle liée aux formalismes qualitatifs est de mettre les entités en relation par un ordre total. Par conséquent, la *relation vide*, notée \emptyset , permet de spécifier que deux entités ne peuvent être mises en relations, ce qui dénote une incohérence dans les connaissances représentées.

Dans les formalismes qualitatifs, l'ensemble des relations complexes $2^{\mathcal{B}}$ est muni d'un ensemble d'opérateurs.

Définition 29 (Stabilité) Soient \mathcal{F} un formalisme qualitatif, k relations complexes $R_1, \dots, R_k \in 2^{\mathcal{B}}$ et OP un opérateur binaire de $2^{\mathcal{B}} \times 2^{\mathcal{B}}$ dans $2^{\mathcal{B}}$. $\{R_1, \dots, R_k\}$ est stable par OP si pour tout $i, j \in [1 ; k]$, $\text{OP}(R_i, R_j) \in \{R_1, \dots, R_k\}$.

Tout d'abord, les opérateurs ensemblistes usuelles sont définies sur $2^{\mathcal{B}}$. L'intersection (\cap) a pour élément absorbant la relation \emptyset et pour élément neutre la relation ψ . L'union (\cup) a pour élément absorbant la relation ψ et pour élément neutre la relation \emptyset . Ces opérateurs sont associatifs, commutatifs et distributifs. Il est également possible d'utiliser l'opérateur *complémentaire* sur \mathcal{B} , tel que $\complement_{\mathcal{B}} R = \{r : r \in \mathcal{B}, r \notin R\}$. Puisque $2^{\mathcal{B}}$ est l'ensemble des parties de \mathcal{B} , il est clair que ces opérateurs sont stables.

Sur un formalisme qualitatif \mathcal{F} et pour toute relation atomique $r \in \mathcal{B}$, l'inverse de r est définie par $r^{-1} = \{(y, x) \in \mathbb{D}^2 : (x, y) \in r\}$. Dans la suite, nous supposons que l'ensemble \mathcal{B} est stable par inverse, et nous étendons cette définition à l'ensemble $2^{\mathcal{B}}$, en conséquence stable par inverse également.

Définition 30 (Inverse d'une relation complexe) Soient \mathcal{F} un formalisme qualitatif et $R \in 2^{\mathcal{B}}$ une relation complexe. $R^{-1} = \{r^{-1} : r \in R\}$.

De plus, les formalismes qualitatifs possèdent deux lois de composition sur $2^{\mathcal{B}} \times 2^{\mathcal{B}}$. La *composition forte*, notée \circ , fait référence à la composition mathématique usuelle.

Définition 31 (Composition forte) Soient \mathcal{F} un formalisme qualitatif et $R_1, R_2 \in 2^{\mathcal{B}}$. La composition forte est définie par : $R_1 \circ R_2 = \{(x, z) \in \mathbb{D}^2 : \exists y \in \mathbb{D}, (x, y) \in R_1 \wedge (y, z) \in R_2\}$.

Cette loi de composition n'est pas totalement satisfaisante puisque celle-ci n'est pas une loi de composition interne en règle générale, et donc il n'existe pas toujours une relation $R \in 2^{\mathcal{B}}$ telle que $(x, y) \in R$

si et seulement si $(x, y) \in \circ(R_1, R_2)$. Si des connaissances sont exprimées à l'aide d'un formalisme muni de cette loi de composition, il est possible de devoir traiter un ensemble infini de relations, ce qui sort du cadre des formalismes qualitatifs, défini sur un ensemble fini de relations atomiques. Dans le cas général, il est nécessaire d'affaiblir la définition de la composition pour que l'ensemble $2^{\mathcal{B}}$ soit stable pour cette opération de composition. La *composition faible* est ainsi une approximation de la composition mathématique.

Définition 32 (Composition faible) Soient \mathcal{F} un formalisme qualitatif et $R_1, R_2 \in 2^{\mathcal{B}}$. La composition faible est définie par : $R_1 \diamond R_2 = \bigcap \{R \in 2^{\mathcal{B}} : R_1 \circ R_2 \subseteq R\}$.

La composition faible $R_1 \diamond R_2$ est donc définie comme la plus petite relation de $2^{\mathcal{B}}$ contenant $R_1 \circ R_2$. Par définition, l'ensemble $2^{\mathcal{B}}$ est stable par composition faible, donc \diamond est une loi de composition interne dans $2^{\mathcal{B}}$, et il est évident que $R_1 \circ R_2 \subseteq R_1 \diamond R_2$. Pour la composition faible, la relation ψ est l'élément neutre et la relation \emptyset est l'élément absorbant. Notons que, pour de nombreux formalismes qualitatifs, la composition faible est associative, c'est-à-dire $R_1 \diamond R_2 \diamond R_3 = R_1 \diamond (R_2 \diamond R_3) = (R_1 \diamond R_2) \diamond R_3$, pour $R_1, R_2, R_3 \in 2^{\mathcal{B}}$, mais n'est pas commutative dans le cas général. Remarquons, au même titre que la composition mathématique de relations binaires, que $(R_1 \diamond R_2)^{-1} = R_2^{-1} \diamond R_1^{-1}$, pour tout $R_1, R_2 \in 2^{\mathcal{B}}$. De plus, nous avons la propriété $\text{id} \in r \diamond r^{-1}$, pour tout $r \in \mathcal{B}$, qui suit directement la définition de composition mathématique. On appelle *table de composition*, abrégée TC, le tableau à deux entrées permettant de représenter le résultat des $|\mathcal{B}|^2$ compositions faibles des relations singletons d'un formalisme qualitatif.

Pour certains formalismes qualitatifs, la structure $(2^{\mathcal{B}}, \cup, \cap, \emptyset, \psi, \diamond, ^{-1}, \text{id})$ décrit une algèbre non-associative ou une algèbre relationnelle. Ces formalismes qualitatifs sont également appelées *algèbres qualitatives* [LIGOZAT & RENZ 2004].

2.2.3 Quelques formalismes qualitatifs

Pour modéliser des connaissances temporelles ou spatiales, de nombreux formalismes qualitatifs ont été développés. Nous décrivons dans la suite quelques formalismes qualitatifs très étudiés. En particulier, nous présentons l'*algèbre des points*, l'un des formalismes qualitatifs les plus élémentaires. Nous présentons également l'*algèbre des intervalles*, l'un des formalismes qualitatifs pour le temps les plus étudiés. Nous proposons un aperçu de l'*algèbre des arcs cycliques*, un autre formalisme qualitatif basé sur des intervalles temporelles ayant des propriétés particulières. Puis nous terminerons par le formalisme qualitatif RCC (pour *Région Connection Calculus*) le plus connu pour la représentation spatiale.

L'algèbre des points

L'une des algèbres qualitatives les plus élémentaires est l'*algèbre des points*, définie par Vilain et Kautz [VILAIN & KAUTZ 1986, VILAIN *et al.* 1990]. C'est un formalisme qualitatif binaire permettant de décrire des entités ponctuelles sur une droite, dont les relations décrivent un pré-ordre totale sur les entités. Le tableau 2.1 donne une définition des relations atomiques $\mathcal{B}_{pt} = \{<, =, >\}$ de l'algèbre des points et le tableau 2.2 en donne une illustration. Dans ce formalisme qualitatif, la relation identité est décrite par $=$. L'ensemble $2^{\mathcal{B}_{pt}}$ est donc composé de 8 relations complexes, tel que $2^{\mathcal{B}_{pt}} = \{\emptyset, \{<\}, \{=\}, \{>\}, \{<, =\}, \{=, >\}, \{<, >\}, \{<, =, >\}\}$.

Le tableau 2.3 donne la table de composition faible pour l'algèbre des points.

L'exemple 8 illustre la modélisation d'un problème à l'aide de l'algèbre de points.

Exemple 8 Trois étudiants, E_1 , E_2 et E_3 , doivent prendre place dans une salle de classe. Chaque chaise de la salle possède un numéro unique. Nous souhaitons modéliser les informations suivantes à l'aide de

Description	Relation	Définition
Précède	<	$\{(x, y) \in \mathbb{Q}^2 : x < y\}$
Est confondu avec	=	$\{(x, y) \in \mathbb{Q}^2 : x = y\}$
Succède	>	$\{(x, y) \in \mathbb{Q}^2 : x > y\}$

TABLE 2.1 – Définition des relations de l’algèbre des points sur l’ensemble \mathbb{Q}

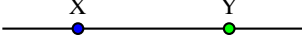
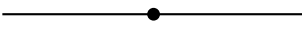
Relation	Illustration	Relation inverse
X précède Y <		Y succède X >
X est égal à Y =		Y est égal à X =

TABLE 2.2 – Illustration des relations atomiques de l’algèbre des points

l’algèbre des points : « deux étudiants ne peuvent avoir la même chaise », « E_1 est avant E_2 » et « E_2 est avant E_3 ».

Chaque étudiant représente une entité dont le domaine est défini par la numérotation des chaises. Dans ce cas, nous supposons que la numérotation décrit l’ensemble \mathbb{Z} . Les étudiants sont mis en relation par $\mathcal{B}_{pt} = \{<, =, >\}$ sur \mathbb{Z} .

En conséquence, la conjonction de ces informations est modélisée par :

$$E_1\{<\}E_2, E_1\{<, >\}E_3, E_2\{<\}E_3$$

En nous appuyant sur l’exemple 8, nous pouvons également illustrer la différence entre composition faible et composition forte.

Exemple 9 Reprenons l’exemple 8. Nous savons que $E_1\{<\}E_2$ et $E_2\{<\}E_3$. La composition de ses informations permet de déduire la relation entre E_1 et E_3 .

En utilisant la définition 31 de la composition forte, nous obtenons :

$$\{<\} \circ \{<\} = \{(x, z) \in \mathbb{Z}^2 : \exists y \in \mathbb{Z}, x\{<\}y, y\{<\}z\}$$

Autrement écrit :

$$\{<\} \circ \{<\} = \{(i, j) \in \mathbb{Z}^2 : j - i \geq 2\}$$

En vérifiant la définition des relations (Tableau 2.1), nous pouvons remarquer que $\{<\} \circ \{<\} \notin 2^{\mathcal{B}_{pt}}$. L’ensemble $2^{\mathcal{B}_{pt}}$ n’est donc pas stable par composition forte. La composition engendre une nouvelle relation hors de $2^{\mathcal{B}_{pt}}$, qui par composition forte pourrait engendrer de nouvelles relations. Il est donc impossible d’utiliser la composition forte en garantissant de rester dans l’algèbre des points.

◇	=	<	>
=	=	<	>
<	<	<	{=, <, >}
>	>	{=, <, >}	>

TABLE 2.3 – Table de composition faible pour l’algèbre des points

Information	Modélisation
« E_1 et E_2 ne peuvent avoir la même chaise »	$E_1\{\langle, \rangle\}E_2$
« E_1 et E_3 ne peuvent avoir la même chaise »	$E_1\{\langle, \rangle\}E_3$
« E_2 et E_3 ne peuvent avoir la même chaise »	$E_2\{\langle, \rangle\}E_3$
« E_1 est avant E_2 »	$E_1\{\langle\}E_2$
« E_2 est avant E_3 »	$E_2\{\langle\}E_3$

Par définition, la plus petite relation de $2^{\mathcal{B}}$ contenant $\{\langle\} \diamond \{\langle\}$ est $\{\langle\}$. Par conséquent, $\{\langle\} \diamond \{\langle\} = \{\langle\}$. Par composition faible, nous pouvons donc déduire que si $E_1\{\langle\}E_2$ et $E_2\{\langle\}E_3$ alors $E_1\{\langle\}E_3$.

L'exemple 9 montre que l'utilisation de la composition faible est bien une approximation de la composition forte. L'information obtenue par composition forte exprime « E_1 doit être au minimum 2 places avant E_3 », tandis que la composition permet de déduire « E_1 est avant l'étudiant E_2 ». L'aspect qualitatif prime dans ce cadre et l'information obtenue par composition faible n'est pas erronée. Notons cependant que si le choix du domaine pour le formalisme qualitatif se porte sur un ensemble dense, tels que \mathbb{R} ou \mathbb{Q} , alors la composition forte et la composition faible sont équivalentes. Il n'y a donc plus d'approximation par composition faible.

De nombreux formalismes qualitatifs basés sur les entités ponctuelles étendent le cadre de l'algèbre des points. Balbiani et Condotta [BALBIANI & CONDOTTA 2002] généralisent l'algèbre des points à un espace cartésien de dimension n , avec $n \leq 1$, ce qui impose 3^n relations atomiques. L'algèbre des relations cardinales [LIGOZAT 1998b] permet de décrire les points d'un plan cartésien. La position relative entre deux points est décrite par une direction par rapport à l'origine, dénotée : Nord, Sud, Est, Ouest, Nord-Est, Nord-Ouest, Sud-Est et Sud-Ouest. Par exemple, un point se situe au même endroit, au Nord ou au Sud-Ouest d'un autre point. L'algèbre des n -star [MITRA 2002] est une généralisation de l'algèbre des relations cardinales. Condotta et Saade [SAADE 2008] définissent l'algèbre des directions cardinales sur \mathbb{N}^2 en se basant sur l'algèbre des points cardinaux afin de décrire le problème académique des n -reines par un formalisme qualitatif. Ils complètent les positions relatives de cette algèbre en intégrant la notion de distance par rapport à l'origine. Ainsi, les relations atomiques *Nord*, *Nord-Ouest*, *Ouest*, *etc* dénotent les directions cardinales Nord, Nord-Ouest, Ouest, *etc* avec une distance de 1 par rapport à l'origine, et les relations atomiques *Plus au Nord*, *Plus au Nord-Ouest*, *Plus à l'Ouest*, *etc* dénotent les mêmes directions cardinales avec une distance à l'origine supérieure à 1. Cette algèbre est donc composée de $3n + 1$ relations atomiques. Nous pouvons enfin citer [BROXVALL & JONSSON 2003] qui proposent une généralisation de l'algèbre des points en ajoutant une relation atomique \parallel décrivant l'incomparabilité de deux entités.

L'algèbre des intervalles

Pour mettre en relations des entités sous forme d'intervalles de temps, Allen définit l'algèbre des intervalles [ALLEN 1981, ALLEN 1983]. Un intervalle de temps $X = [x^- ; x^+]$ est défini sur une droite, par sa borne inférieure $x^- \in \mathbb{Q}$ et sa borne supérieure $x^+ \in \mathbb{Q}$ avec $x^- < x^+$. Le formalisme d'Allen est binaire, et 13 relations atomiques décrivent les positions relatives entre deux intervalles de temps : $\mathcal{B}_{int} = \{b, m, o, s, fi, d, eq, si, di, f, oi, mi, a\}$. L'ensemble $2^{\mathcal{B}_{int}}$ est donc composé de 8192 relations complexes. La relation $\{eq\}$ est la relation identité de ce formalisme qualitatif. Le tableau 2.4 illustre les relations atomiques et leurs inverses ; le tableau 2.5 définit chaque relation atomique entre deux intervalles définis sur la droite des nombres rationnels. L'annexe A permet de visualiser la table de composition faible pour le formalisme d'Allen, contenant $13^2 = 169$ relations complexes. L'utilisation d'un ensemble dense permet d'obtenir l'équivalence entre composition faible et composition forte.


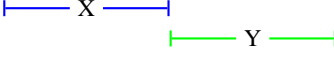

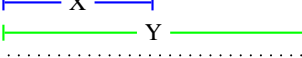
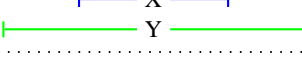
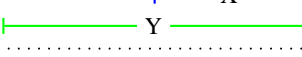
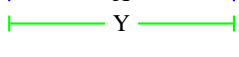
Relation	Illustration	Relation inverse
X précède Y <i>b</i>		Y succède X <i>a</i>
X rencontre Y <i>m</i>		Y est rencontré par X <i>mi</i>
X chevauche Y <i>o</i>		Y est chevauché par X <i>oi</i>
X commence Y <i>s</i>		Y est commencé par X <i>si</i>
X est contenu par Y <i>d</i>		Y contient X <i>di</i>
X fini Y <i>f</i>		Y est fini par X <i>fi</i>
X est égal à Y <i>eq</i>		Y est égal à X <i>eq</i>

TABLE 2.4 – Illustration des relations atomiques de l’algèbre des intervalles

L’exemple 10 donne un aperçu de la modélisation d’informations temporelles dans le formalisme d’Allen. Remarquons que certaines précautions doivent être prises car chacune des informations est traduite depuis le langage naturel. Par exemple, le mot « avant » du langage naturel peut sous-entendre la relation $\{b\}$ (*avant*), $\{m\}$ (*rencontre*) ou $\{b, m\}$. Il est donc nécessaire de clarifier les informations.

Exemple 10 *Lors d’une séance de cours, il est demandé à Anatole, Béatrice et Charles de se répartir la correction d’un exercice chacun et séparément. Nous souhaitons donc modéliser les informations suivantes : « chaque correction est réalisée durant la séance » et « deux corrections sont disjointes ».*

Tout d’abord, nous pouvons entrevoir quatre intervalles de temps : la séance de cours (T), la correction d’Anatole (A), la correction de Béatrice (B) et la correction de Charles (C). Par conséquent, le formalisme d’Allen est adapté à la modélisation. Pour modéliser la première information, il est nécessaire que l’intervalle représentant la correction soit inclus dans l’intervalle de temps du cours. Quatre relations atomiques du tableau 2.4 correspondent à ce cadre : « la correction commence, est contenu par, termine ou est égale à la séance de cours ». De même, en nous basant sur la même figure pour la seconde information, quatre relations atomiques correspondent : « une correction est effectuée avant, rencontre, est rencontrée ou est effectuée après une autre correction ».

Information	Modélisation
« la correction A est réalisée durant la séance T »	$A \{s, d, f, eq\} T$
« la correction B est réalisée durant la séance T »	$B \{s, d, f, eq\} T$
« la correction C est réalisée durant la séance T »	$C \{s, d, f, eq\} T$
« les corrections A et B sont disjointes »	$A \{b, m, mi, a\} B$
« les corrections A et C sont disjointes »	$A \{b, m, mi, a\} C$
« les corrections B et C sont disjointes »	$B \{b, m, mi, a\} C$

Par conséquent, la modélisation de ces informations à l’aide de l’algèbre des intervalles est :

$$A \{s, d, f, eq\} T, \quad B \{s, d, f, eq\} T, \quad C \{s, d, f, eq\} T, \\ A \{b, m, mi, a\} B, \quad A \{b, m, mi, a\} C, \quad B \{b, m, mi, a\} C$$

Description	Relation	Ensemble
Précède	b	$\{(X, Y) \in \mathbb{Q}^2 \times \mathbb{Q}^2 : x^+ < y^-\}$
Rencontre	m	$\{(X, Y) \in \mathbb{Q}^2 \times \mathbb{Q}^2 : x^+ = y^-\}$
Chevauche	o	$\{(X, Y) \in \mathbb{Q}^2 \times \mathbb{Q}^2 : x^- < y^-, x^+ > y^-, x^+ < y^+\}$
Commence	s	$\{(X, Y) \in \mathbb{Q}^2 \times \mathbb{Q}^2 : x^- = y^-, x^+ < y^+\}$
Est contenu dans	d	$\{(X, Y) \in \mathbb{Q}^2 \times \mathbb{Q}^2 : x^- > y^-, x^+ < y^+\}$
Fini	f	$\{(X, Y) \in \mathbb{Q}^2 \times \mathbb{Q}^2 : x^- > y^-, x^+ = y^+\}$
Est confondu avec	eq	$\{(X, Y) \in \mathbb{Q}^2 \times \mathbb{Q}^2 : x^- = y^-, x^+ = y^+\}$
Est fini par	fi	f^{-1}
Contient	di	d^{-1}
Est commencé par	si	s^{-1}
Est chevauché par	oi	o^{-1}
Est rencontré par	mi	m^{-1}
Succède	a	b^{-1}

TABLE 2.5 – Définition des relations atomiques de l’algèbre des intervalles sur \mathbb{Q}^2 pour $X = [x^- ; x^+]$ et $Y = [y^- ; y^+]$

L’algèbre des intervalles est l’un des formalismes qualitatifs les plus utilisés. La simplicité et l’expressivité de sa représentation contribue à son succès. Allen a défini ce formalisme pour des problèmes de compréhension de langage naturel et d’ordonnancement alors que la modélisation sous forme de formalisme quantitatif devenait complexe et posait des problèmes de mémoire pour sauvegarder les informations. Cette nécessité d’augmenter les données quantitatives dans de nombreuses applications a conduit à la modélisation des connaissances temporelles par ce formalisme qualitatif.

Le formalisme d’Allen a lui aussi été raffiné et généralisé. L’algèbre des rectangles [BALBIANI *et al.* 1998] permet la modélisation de rectangle dans le plan cartésien. Un rectangle R est décrit par deux intervalles R_x et R_y , représentant sa projection sur les deux axes du plan. Chaque position relative entre deux rectangles est décrite par un couple de $\mathcal{B}_{int} \times \mathcal{B}_{int}$, correspondant aux positions relatives des intervalles issu de la projection de ce rectangle sur les deux axes du plan. Par exemple, $R1\{(d, m)\}R2$ sera vérifié lorsque $R1_x\{d\}R2_x$ et $R1_y\{m\}R2_y$. 169 relations atomiques composent donc cette algèbre. L’algèbre des n -pavé [BALBIANI *et al.* 2003a] généralise le raisonnement à un espace cartésien à n dimensions. Ladkin [LADKIN 1986] définit un formalisme qualitatif permettant de représenter des intervalles non convexes, appelé p -intervalle, c’est-à-dire qu’un même intervalle est composé de p parties distinctes. Pour définir les positions relatives d’un p -intervalle et d’un q -intervalle, chaque relation atomique r est décrite par une matrice $p \times q$ pour laquelle chaque élément est une relation atomique de \mathcal{B}_{int} . L’élément $r(i, j)$ décrit alors la position relative entre le i^e intervalle du p -intervalle et le j^e intervalle du q -intervalle. Ligozat [LIGOZAT 1991] décrit le formalisme qualitatif en exploitant les bornes des intervalles. Pour étendre l’algèbre des intervalles en considérant la taille relative entre intervalles en plus de la position relative, [PUJARI *et al.* 1999] proposent l’algèbre *INDU*. Dans ce formalisme qualitatif, un intervalle peut donc « être avant et plus petit », « rencontrer et être de même taille » ou encore « chevaucher et être plus petit » qu’un autre intervalle. Ce formalisme compte 25 relations atomiques en couplant les 13 relations atomiques de l’algèbre des intervalles aux ordre de grandeurs.

L’algèbre des intervalles cycliques

L’algèbre des intervalles cycliques [BALBIANI & OSMANI 2000] est un formalisme qualitatif permettant également de raisonner sur des entités représentées par des intervalles. Cependant, chaque entité représente un arc $[x^- ; x^+]$ sur cercle orienté \mathcal{C} tel que $(x^-, x^+) \in \mathcal{C} \times \mathcal{C}$ avec $x^- \neq x^+$. La figure

2.1 permet de visualiser les 16 positions relatives entre deux arcs orientés AB et CD d'un cercle C décrite par l'ensemble $\mathcal{B}_{arcs} = \{ppi, m, o, s, fi, d, eq, di, f, si, oi, mi, ooi, moi, omi, mmi\}$. L'ensemble $2^{\mathcal{B}_{arcs}}$ contient ainsi 65536 relations complexes.

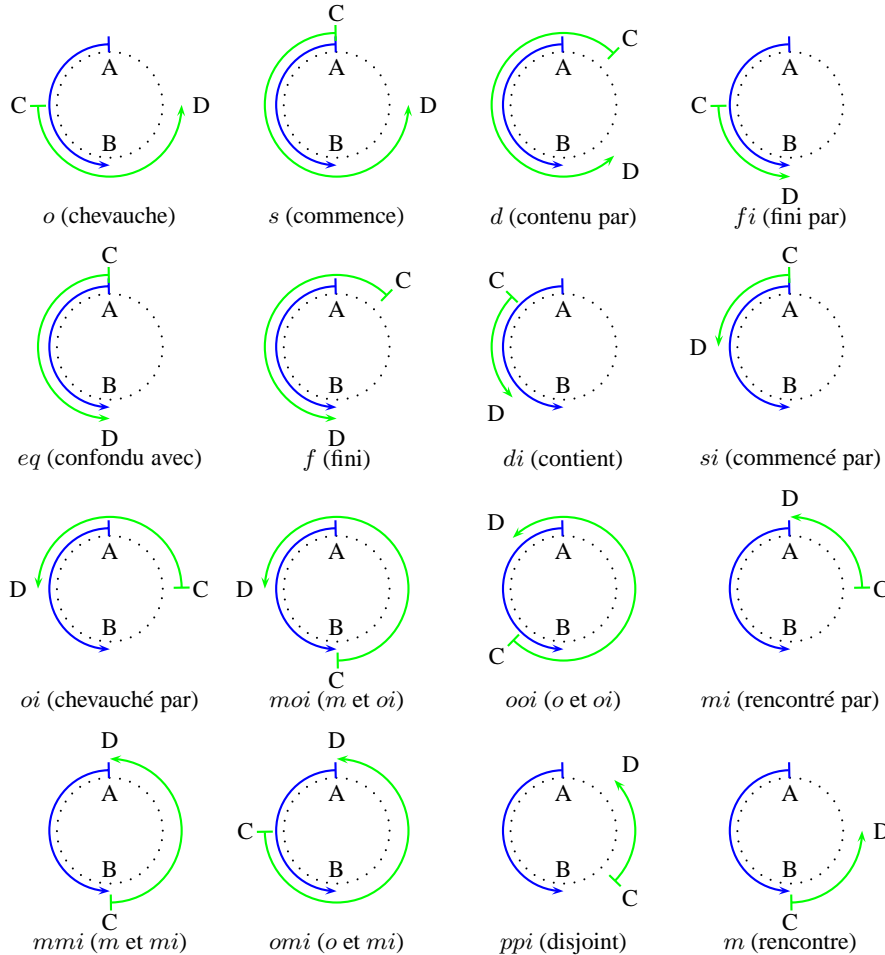


FIGURE 2.1 – Relations atomiques de l'algèbre des intervalles cycliques

Le formalisme RCC

RCC pour *Region Connection Calculus* est le principal formalisme qualitatif basé sur les relations topologiques [RANDELL & COHN 1989, RANDELL *et al.* 1992]. Dans un plan cartésien, des régions convexes ou concaves sont définies. Les positions relatives entre ces régions constituent les relations atomiques du formalisme qualitatif. RCC5 caractérisent les cinq relations atomiques suivante : deux régions sont égales deux régions sont égales, deux régions sont incluses l'une dans l'autre, deux régions sont disjointes ou deux régions se chevauchent. RCC8 étend ce formalisme qualitatif et affine les positions relatives distinguant que les régions peuvent être tangentes. Huit relations atomiques décrivent alors les positions relatives : l'égalité (EQ), la disconnection (DC), la connection externe (EC), le chevauchement partiel (PO), la partie propre tangente (TPP) et son inverse ($TPPI$), la partie propre non-tangente ($NTPP$) et son inverse ($NTPPI$) Les relations $\mathcal{B}_{RCC8} = \{EQ, DC, EC, PO, TPP, TPPI, NTPP, NTPPI\}$ sont illustrées Figure 2.2. L'ensemble $2^{\mathcal{B}_{RCC8}}$ est donc décrit par 256 relations complexes.

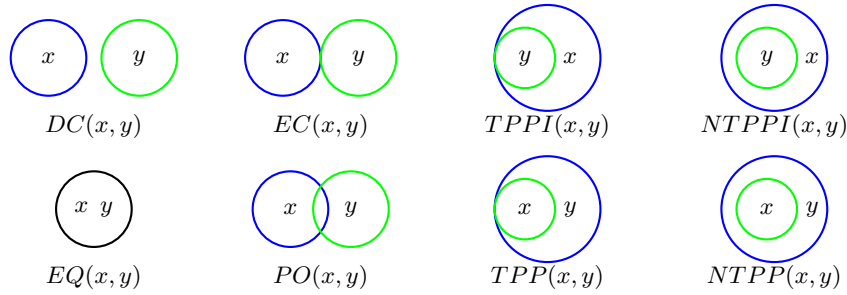


FIGURE 2.2 – Illustration des relations atomiques de RCC8

2.3 Les réseaux de contraintes qualitatives

Les formalismes qualitatifs permettent de modéliser des connaissances temporelles et spatiales de façon naturelle. La motivation derrière cette modélisation est de pouvoir vérifier la cohérence de ces connaissances ou encore de déterminer la cause de l'incohérence de celles-ci. Pour cela, le résultat de la modélisation à l'aide des formalismes qualitatifs a été introduit dans un réseau de contraintes pour former un *réseau de contraintes qualitatives*, noté RCQ. Décider de la cohérence d'un RCQ est le problème de satisfaction de contraintes qualitatives, que nous nommerons problème CSP qualitatif dans la suite. Contrairement au cadre des RCD, le domaine des entités n'est pas fini dans le cas général. Il n'est donc pas possible d'énumérer ou de parcourir l'espace de recherche afin de déterminer la cohérence ou non des connaissances. Néanmoins, en s'appuyant sur l'aspect qualitatif du formalisme, il est possible de vérifier que les positions relatives des entités sont cohérentes entre elles. Pour vérifier la cohérence d'un RCQ, il est possible de manipuler les contraintes définies, et non le domaine comme dans l'approche discrète. Une seule position relative est choisie entre les entités parmi la relation complexe modélisée, afin d'obtenir un RCQ atomique appelé *scénario*. Pour de nombreux formalismes qualitatifs, si un tel scénario vérifie les propriétés d'inverse et de composition, alors le RCQ est cohérent. Le problème consistant à déterminer s'il existe un scénario cohérent est NP-complet dans le cas général. Le nombre de scénario est exponentiel dans la taille du problème mais fini. Il est nécessaire d'adopter une approche similaire au cadre des RCD, c'est-à-dire énumérer l'ensemble des scénarios à l'aide d'un arbre de recherche sur les relations atomiques, et de vérifier s'il existe un scénario vérifiant les propriétés du formalisme qualitatif. Comme pour le cadre des RCD, il est ainsi possible de définir des techniques d'inférence afin de réduire l'espace de recherche à évaluer.

2.3.1 Définition générale

Définition 33 (Réseau de Contraintes Qualitatives) Un Réseau de Contraintes Qualitatives, noté RCQ, construit sur un formalisme qualitatif \mathcal{F} est défini par un couple $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ où :

- $\mathcal{V} = \{v_1, \dots, v_n\}$ est un ensemble fini de variables représentant les entités temporelles ou spatiales ;
- $\mathcal{Q} = \{Q_1, \dots, Q_e\}$ décrit un ensemble fini de contraintes qualitatives binaires entre les variables de \mathcal{V} .

Un RCQ est un outil de représentation des connaissances qui sont retranscrites à l'aide d'un formalisme qualitatif défini sur un domaine \mathbb{D} de relations atomiques \mathcal{B} . Les entités temporelles ou spatiales sont caractérisées par l'ensemble de variables \mathcal{V} . Ainsi, pour tout $v \in \mathcal{V}$, le domaine de v est $D_v = \mathbb{D}$. C'est la raison pour laquelle, contrairement aux RCD, l'ensemble des domaines n'est pas explicité pour décrire la microstructure du RCQ. L'ensemble \mathcal{Q} décrit les contraintes qualitatives entre les variables

de \mathcal{V} . Dans le cas général, les contraintes d'un RCQ sont de la même arité que le formalisme qualitatif utilisé. Nous étudierons donc uniquement les RCQ binaires. L'application scp associe la portée à toute contrainte $Q \in \mathcal{Q}$, telle que $scp(Q) \subseteq \mathcal{V}$ et $scp(Q) \neq \emptyset$. Tout couple de variables est impliqué dans une contrainte de \mathcal{Q} , par conséquent le RCQ est *complet*. Toute contrainte $Q \in \mathcal{Q}$ caractérise une relation complexe, décrite par l'application rel , telle que $rel(Q) \in 2^{\mathcal{B}}$. Une contrainte qualitative $Q \in \mathcal{Q}$ peut donc être considérée comme un ensemble de tuples supports décrit par $\{\bigcup_{r \in rel(Q)} r\}$. Dans la suite, afin de faciliter la lecture, $Q_{(v_i, v_j)}$ décrira la relation complexe $rel(Q)$ de la contrainte $Q \in \mathcal{Q}$ telle que $scp(Q) = \{v_i, v_j\}$.

L'analogie entre « relations complexes » et « contraintes qualitatives » permet l'utilisation du lexique défini pour les relations complexes. Les relations remarquables jouent un rôle significatif dans les RCQ. Pour tout $v_i, v_j \in \mathcal{V}$, $Q_{(v_i, v_j)} = \psi$ décrit une *contrainte totale* ou *contrainte universelle*, et $Q_{(v_i, v_j)} = \{id\}$ décrit la *contrainte identité*. De plus, la *contrainte vide*, décrite par $Q_{(v_i, v_j)} = \emptyset$, dénote une incohérence dans le RCQ et les connaissances modélisées. Puisque une contrainte est caractérisée par une relation complexe, les opérations d'union, d'intersection et d'inverse sont bien définies. Comme dans le cadre des RCD, nous pouvons définir la normalisation d'un RCQ.

Définition 34 (RCQ normalisé) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ construit sur un formalisme \mathcal{F} . \mathcal{N} est normalisé si et seulement si \mathcal{N} vérifie, pour tout $v_i, v_j \in \mathcal{V}$:

- $Q_{(v_i, v_i)} = \{id\}$ (*identité*);
- $Q_{(v_i, v_j)} = Q_{(v_j, v_i)}^{-1}$ (*inverse*).

Dans le cas général, quel que soit le RCQ et le formalisme qualitatif étudié, vérifier ces propriétés ou les appliquer est réalisable en temps polynomiale dans le nombre de variables du RCQ. Par conséquent, nous nous concentrerons uniquement sur des RCQ normalisés dans la suite.

La figure 2.3 présente un RCQ binaire construit sur le formalisme d'Allen, qui est la représentation de l'exemple 10, modélisant le problème des étudiants corrigeant chacun un exercice lors d'une séance de cours. Le RCQ est assimilé à un graphe orienté, où chaque arc est étiqueté d'une contrainte entre les variables représentant la microstructure. Par exemple, l'arc $Q_{(A, T)} = \{s, d, f, eq\}$ décrit l'information « A commence, est contenu dans, fini, ou est égal à T ». Puisque chaque relation atomique est complète et mutuellement exclusive, une et une seule des relations atomiques peut décrire la position relative entre A et T . Puisque le RCQ est normalisé, l'intégralité des contraintes n'est pas représentée sur les illustrations par soucis de clarté. La contrainte identité entre une variable et elle-même n'est pas pertinente. De plus, une contrainte peut être déduite lorsque son inverse est mentionné. Par exemple, la contrainte $Q_{(A, T)} = \{s, d, f, eq\}$ a pour inverse $Q_{(T, A)} = (Q_{(A, T)})^{-1} = \{s, d, f, eq\}^{-1} = \{si, di, fi, eq\}$. Enfin, dans le cas général, si il existe une contrainte totale entre deux entités, celle-ci n'est pas illustrée.

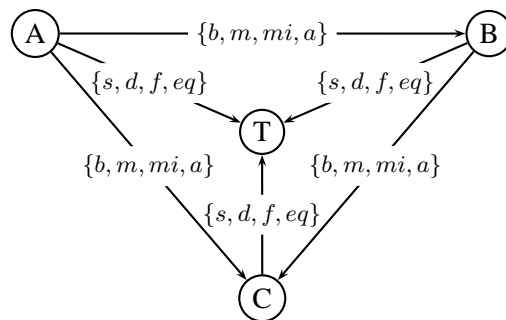


FIGURE 2.3 – Anatole (A), Béatrice (B) et Charles (C) doivent se répartir la correction d'un exercice chacun et séparément lors d'une séance de cours (T)

2.3.2 Problème de satisfiabilité de contraintes qualitatives

La modélisation de connaissances par l'utilisation des RCQ est motivée par la vérification de l'intégrité des connaissances. L'objectif est donc de décider de la satisfiabilité des contraintes qualitatives, c'est-à-dire déterminer l'existence d'une solution du RCQ.

Définition 35 (Instanciation complète) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ construit sur un formalisme qualitatif \mathcal{F} . Une instanciation complète \mathcal{I} de \mathcal{N} est une application de \mathcal{V} dans \mathbb{D} , qui à tout $v \in \mathcal{V}$ associe une valeur $m \in \mathbb{D}$.

Définition 36 (solution) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ construit sur un formalisme qualitatif \mathcal{F} . Une solution est une instanciation complète \mathcal{I} telle que, pour tout $v_i, v_j \in \mathcal{V}$, $(\mathcal{I}(v_i), \mathcal{I}(v_j)) \in Q_{(v_i, v_j)}$.

Dans le cadre discret, un algorithme de recherche exhaustive prévoit le parcours de chaque instanciation complète, afin de vérifier la cohérence de chacune. Dans le cadre de domaines non finis, l'espace de recherche n'est pas borné, et il n'est évidemment pas possible de considérer chaque élément du domaine des entités pour parcourir les instanciations complètes. Pour répondre au problème de la cohérence d'un RCQ, il faut dans un premier temps imposer la contrainte implicite des formalismes qualitatifs : les relations atomiques de \mathcal{B} sont complètes et mutuellement exclusives. En d'autres termes, tout couple de valeurs de \mathbb{D}^2 n'appartient qu'à une et une seule relation atomique de \mathcal{B} . Cela nous conduit à définir un RCQ atomique appelé scénario.

Définition 37 (Scénario) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ construit sur un formalisme qualitatif \mathcal{F} . Un scénario de \mathcal{N} est un RCQ $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ tel que, pour tout $v_i, v_j \in \mathcal{V}$, $Q'_{(v_i, v_j)} \subseteq Q_{(v_i, v_j)}$ et $|Q'_{(v_i, v_j)}| = 1$.

La définition d'un scénario illustre la relation d'inclusion entre RCQ.

Définition 38 (Inclusion de RCQ) Soient $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ et $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ deux RCQ construits sur un même formalisme qualitatif \mathcal{F} . $\mathcal{N} \sqsubseteq \mathcal{N}'$ si et seulement si, pour tout $v_i, v_j \in \mathcal{V}$, $Q_{(v_i, v_j)} \subseteq Q'_{(v_i, v_j)}$. \mathcal{N} est appelé sous-RCQ de \mathcal{N}' .

Par la définition d'inclusion des RCQ, tout scénario d'un RCQ est un sous-réseau de celui-ci.

Puisque les relations sont complètes et mutuellement exclusives, l'espace de recherche associé à chaque scénario est inclus dans l'espace de recherche du RCQ complet, et deux scénarios distincts ont des espaces de recherche disjoints. En d'autres termes, un scénario permet de décrire une partie de l'espace de recherche d'un RCQ et l'ensemble des scénarios d'un RCQ forme une partition de l'espace de recherche du RCQ. Comme le nombre de relations atomiques de chaque contrainte et le nombre de variables sont finis, nous pouvons déduire qu'il existe un nombre fini de scénarios, bien qu'exponentiel dans le nombre de variables du RCQ. Par conséquent, puisque chaque scénario factorise un ensemble d'instanciations complètes, il est ainsi possible de « survoler » l'espace de recherche d'un RCQ en énumérant les scénarios d'un RCQ. Puisqu'un scénario n'est autre qu'un RCQ, nous pouvons définir la cohérence d'un scénario.

Définition 39 (Scénario cohérent) Soient $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un scénario construit sur un formalisme qualitatif \mathcal{F} . \mathcal{N} est cohérent si et seulement si il admet une solution.

Pour répondre au problème de la cohérence d'un RCQ, nous pouvons mettre en évidence la cohérence d'un de ces scénarios. Puisque les relations atomiques d'un formalisme qualitatif sont complètes et mutuellement exclusives, la définition d'un scénario exprime qu'à toute solution est associée un seul

et unique scénario cohérent. Cependant, la réciproque n'est pas vraie car un scénario cohérent peut être associé à un ensemble de solutions. Il convient donc de pouvoir déterminer si un scénario possède au moins une solution. Or, dans le cas de domaine non fini, l'ensemble des instanciations complètes d'un scénario est également infini. Pour palier à ce problème, plusieurs approches peuvent être envisagées suivant le formalisme qualitatif considéré. Pour la majeure partie des formalismes qualitatifs, vérifier la cohérence globale d'un scénario peut être effectué en vérifiant sa cohérence locale, au même titre que la cohérence d'arcs vérifie la cohérence d'une instanciation complète. La clé pour vérifier la cohérence d'un scénario repose donc sur les propriétés des formalismes qualitatifs.

2.3.3 Cohérence locale

Dans le cas général, considérons que le domaine des formalismes qualitatifs soit un ensemble non fini. Puisqu'à tout élément a de \mathbb{D} , il existe un élément b de \mathbb{D} tel que $(a, b) \in r$ quelle que soit la relation $r \in \mathcal{B}$, un RCQ est fermé par cohérence d'arcs. Intéressons-nous à la cohérence de chemins d'un réseau de contraintes. Considérons $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ, et deux variables v_i et v_j de \mathcal{V} . En nous appuyant sur la définition 21 de la cohérence de chemins des RCD, les variables $v_i, v_j \in \mathcal{V}$ sont chemins-cohérentes si et seulement si il existe deux valeurs $a \in D_{v_i}$ et $c \in D_{v_j}$ telle que $(a, c) \in Q_{(v_i, v_j)}$ et pour toute variable $v_k \in \mathcal{V}$, il existe une valeur $b \in D_{v_k}$ telle que $(a, b) \in Q_{(v_i, v_k)}$ et $(b, c) \in Q_{(v_k, v_j)}$. L'ensemble des valeurs chemins-cohérentes de v_i, v_j est donc décrit par l'ensemble $\{(a, c) \in Q_{(v_i, v_j)} : \exists b \in D_{v_k}, (a, b) \in Q_{(v_i, v_k)} \wedge (b, c) \in Q_{(v_k, v_j)}\}$. Or, cet ensemble peut se réécrire $\{(a, c) \in Q_{(v_i, v_j)} : (a, c) \in Q_{(v_i, v_k)} \circ Q_{(v_k, v_j)}\}$. La définition de la cohérence de chemins s'appuie sur la propriété de composition forte. Cependant, comme énoncé précédemment, cet ensemble ne coïncide pas toujours à une relation de $2^{\mathcal{B}}$. Par conséquent, nous exploitons la composition faible afin de définir la *fermeture par composition faible*.

Définition 40 (Fermeture par composition faible RCQ) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ construit sur un formalisme qualitatif \mathcal{F} . \mathcal{N} est fermé par composition faible, si et seulement si $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$, pour tout $v_i, v_j, v_k \in \mathcal{V}$.

La cohérence locale définie est appelée \diamond -cohérence. La fermeture par composition faible est donc également appelé *fermeture par \diamond -cohérence* ou \diamond -fermeture. Puisque nous traitons des RCQ normalisés, il est à noter que la vérification de la \diamond -fermeture peut être restreinte à tout $v_i, v_j, v_k \in \mathcal{V}$ avec $1 \leq i < k < j \leq n$, en exploitant les propriétés de composition faible et d'inverse.

Dans le cas général, si un RCQ est chemins-cohérent alors celui-ci est \diamond -cohérent. La cohérence de chemins et la \diamond -cohérence sont équivalentes si et seulement si la composition forte et la composition faible sont équivalentes. Par extension, la fermeture par cohérence de chemins et la \diamond -fermeture sont également équivalentes. Pour l'algèbre des intervalles, Valdès-Pérez [VALDÉS-PÉREZ 1987] donne la preuve que la \diamond -cohérence d'un scénario est suffisante pour déterminer la cohérence d'un scénario. Ainsi, un algorithme de \diamond -cohérence permet de répondre au problème de la cohérence d'un scénario sur l'algèbre des intervalles. Succinctement, parmi les formalismes qualitatifs présentés, la vérification de \diamond -cohérence est *complète* pour déterminer la cohérence de scénarios construits sur l'algèbre des points, l'algèbre des intervalles et RCC8. Pour les autres formalismes qualitatifs, il est nécessaire d'utiliser une étape supplémentaire afin de vérifier la cohérence du scénario. Par exemple, nous pouvons construire une instanciation des variables du RCQ, variables par variable, en s'attachant à satisfaire la relation atomique.

En exploitant la définition des RCQ normalisés, nous pouvons faciliter la vérification de la \diamond -cohérence d'un scénario à un ensemble de relations. L'intérêt de cette proposition réside dans le nombre de vérifications d'inclusion à effectuer en pratique, mais cela facilite également l'écriture des preuves en théorie.

Proposition 1 Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un scénario normalisé. Si $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$ pour tout $v_i, v_j, v_k \in \mathcal{V}$, avec $1 \leq i < k < j \leq n$, alors \mathcal{N} est \diamond -cohérent.

Preuve Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un scénario normalisé tel que $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$ pour tout $v_i, v_j, v_k \in \mathcal{V}$, avec $1 \leq i < k < j \leq n$. Il existe trois valeurs $d_i, d_j, d_k \in \mathbb{D}$ telles que $(d_i, d_j) \in Q_{(v_i, v_j)}$, $(d_i, d_k) \in Q_{(v_i, v_k)}$, $(d_k, d_j) \in Q_{(v_k, v_j)}$. Puisque \mathcal{N} est normalisé, l'inverse d'une relation permet de déduire $Q_{(v_i, v_j)}^{-1} = Q_{(v_j, v_i)}$, $Q_{(v_i, v_k)}^{-1} = Q_{(v_k, v_i)}$ et $Q_{(v_k, v_j)}^{-1} = Q_{(v_j, v_k)}$. Nous en déduisons $(d_j, d_i) \in Q_{(v_j, v_i)}$, $(d_k, d_i) \in Q_{(v_k, v_i)}$, $(d_j, d_k) \in Q_{(v_j, v_k)}$. De plus, puisque $Q_{(v_i, v_i)} = Q_{(v_j, v_j)} = Q_{(v_k, v_k)} = \{\text{id}\}$, $(d_i, d_i) \in Q_{(v_i, v_i)}$, $(d_j, d_j) \in Q_{(v_j, v_j)}$ et $(d_k, d_k) \in Q_{(v_k, v_k)}$. Donc \mathcal{N} est cohérent sur chaque triplet de variables distinctes $v_i, v_j, v_k \in \mathcal{V}$. Par conséquent, $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$ pour tout triplet de variables distinctes $v_i, v_j, v_k \in \mathcal{V}$.

Avant de poursuivre, rappelons, par l'inverse de la composition faible des relations, que si $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$, alors $Q_{(v_i, v_j)}^{-1} \subseteq (Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)})^{-1}$, et donc $Q_{(v_j, v_i)} \subseteq Q_{(v_j, v_k)} \diamond Q_{(v_k, v_i)}$. Considérons maintenant le cas de l'identité. Trivialement, puisque $\{\text{id}\} = \{\text{id}\} \diamond \{\text{id}\}$, si $i = j = k$ alors $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$. Posons maintenant $1 \leq i, j, k \leq n$ avec $i = j$. Nous savons que $Q_{(v_i, v_j)} = \{\text{id}\}$. Or, par les propriétés de la composition faible, $\{\text{id}\} \subseteq r \diamond r^{-1}$ pour tout $r \in \mathcal{B}$. Puisque $Q_{(v_k, v_j)} = Q_{(v_j, v_k)}^{-1}$, $\{\text{id}\} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_j, v_k)}^{-1}$ et donc $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$ pour tout $1 \leq i = j < k \leq n$. Par l'inverse de la composition faible, nous avons donc $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$ pour tout $1 \leq i, j, k \leq n$ avec $i = j$. De plus, $R = \{\text{id}\} \diamond R = R \diamond \{\text{id}\}$ pour toute relation $R \in 2^{\mathcal{B}}$. Si $i = k$, alors $Q_{(v_i, v_j)} \subseteq \{\text{id}\} \diamond Q_{(v_k, v_j)}$ et donc $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$ pour tout $1 \leq i = k < j \leq n$. En utilisant la propriété d'inverse, $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$ pour tout $1 \leq i, j, k \leq n$ avec $i = k$. Par un raisonnement similaire pour $j = k$, nous montrons $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$ pour tout $1 \leq i, j, k \leq n$ avec $j = k$.

Nous pouvons donc conclure que $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$ pour tout $1 \leq i, j, k \leq n$. Et donc \mathcal{S} est fermé par composition faible. \square

Le cas de certains formalismes qualitatifs amène immédiatement un corollaire à la proposition 1.

Corollaire 1 Soit \mathcal{N} un scénario construit sur un formalisme qualitatif \mathcal{F} pour lequel la fermeture par composition faible est complète pour décider de la cohérence d'un scénario. Si \mathcal{N} est \diamond -fermé alors \mathcal{N} est cohérent.

La figure 2.4 présente un scénario cohérent issu du RCQ présenté Figure 2.3 (à gauche), et un ensemble de solutions du problème modélisé (à droite). En utilisant la table de composition, il est facile de voir que ce scénario est \diamond -cohérent. Chaque relation atomique entre les variables permet de représenter une position relative entre les intervalles de temps. L'aspect qualitatif se dégage du fait qu'il n'y ait ni repère ni échelle, seul la position relative entre les intervalles importe. Il est donc possible d'imaginer un ensemble infini de solutions associé à ce scénario.

Comme énoncé précédemment, la \diamond -fermeture est complète pour décider de la cohérence d'un scénario pour de nombreux formalismes qualitatifs (algèbre des intervalles, algèbre des points, RCC8, etc). La figure 2.5 présente néanmoins le cas de l'algèbre des intervalles cycliques, pour laquelle la méthode de fermeture par composition n'est pas suffisante pour décider de la cohérence d'un scénario. Dans cet exemple, trois intervalles A, B et C forment une partition du cercle \mathcal{C} et nous souhaitons qu'un quatrième intervalle X soit distinct des trois autres. Il est clair que cela est impossible, pourtant le scénario est \diamond -cohérent. Dans le cas de l'algèbre des intervalles cycliques, il est alors nécessaire d'utiliser d'autres procédés pour vérifier la cohérence d'un scénario, notamment en s'appuyant sur les bornes des intervalles.

Renz et Ligozat [RENZ & LIGOZAT 2005] ont déterminé certaines conditions permettant de caractériser un formalisme qualitatif pour lequel la \diamond est suffisante pour décider de la cohérence d'un scénario.

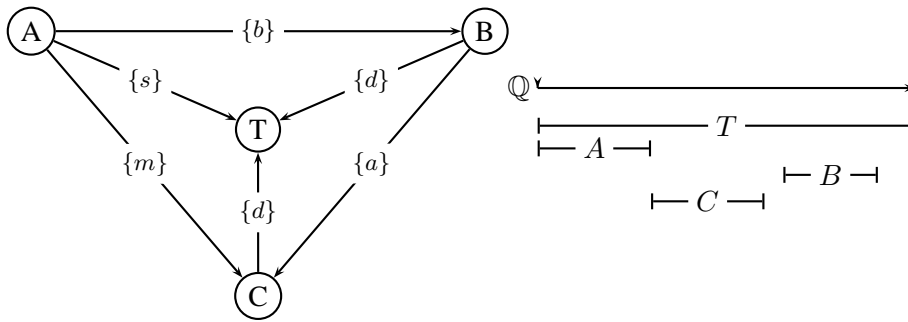


FIGURE 2.4 – Scénario cohérent et instanciations cohérentes pour l'exemple 10

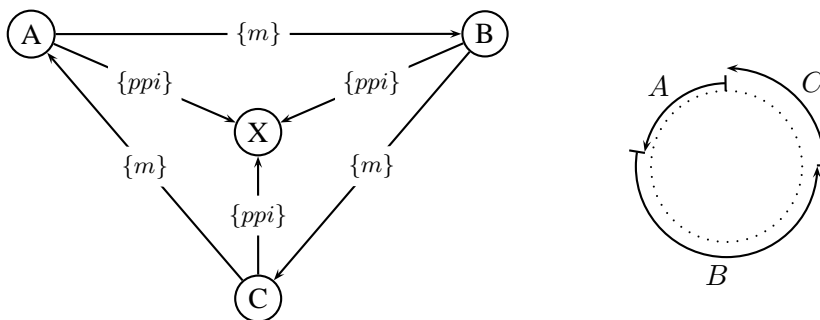


FIGURE 2.5 – Scénario \diamond -cohérent et incohérent de l'algèbre des intervalles cycliques

Ils montrent que cela n'est pas dû à la différence entre composition faible et composition forte, ainsi l'approximation par composition faible n'est pas en cause lorsque les formalismes qualitatifs sont définies sur des ensembles denses. L'un des facteurs pour lequel la \diamond -cohérence n'est pas suffisante pour décider de la cohérence d'un scénario est la possibilité de raffiner les relations atomiques du formalisme considéré en de plus petites relations, ce qui empêche de détecter l'incohérence. L'annexe A.2 illustre le problème des relations qui ne sont pas raffinées. Nous reprenons l'exemple 10 et y imposons un scénario en nous inspirant de l'exemple 2.5 pour l'algèbre des intervalles cycliques. Les trois corrections A , B et C forment une partition du temps de cours T . Nous souhaitons qu'une quatrième correction D ait lieu pendant temps de cours T et qu'elle soit distincte de A , B et C . À l'aide du formalisme d'Allen, il n'existe pas de scénarios \diamond -fermés permettant la description de ce problème, il est donc incohérent. Cependant, nous modifions le formalisme d'Allen, afin de grouper la relation complexe $\{b, a\}$ en une seule relation atomique ppi dénotant la disjonction des intervalles. Nous avons toujours un formalisme qualitatif, mais comme pour l'algèbre des intervalles cycliques, nous pouvons construire un scénario \diamond -fermé mais incohérent.

2.3.4 Algorithme de recherche et techniques d'inférence

Quel que soit le formalisme qualitatif utilisé, si un RCQ n'admet pas de scénario \diamond -cohérent, alors ce RCQ est incohérent. L'approche consiste donc à énumérer les scénarios \diamond -cohérent. Il est ainsi possible de mettre à profit les connaissances développées dans le cadre des RCD pour définir les algorithmes de recherches et les techniques d'inférence dans le raisonnement qualitatif. En s'appuyant sur un algorithme de type MAC, la première étape consiste à énumérer l'ensemble des scénarios d'un RCQ tout en

maintenant la \diamond -fermeture du RCQ.

L'algorithme 8 présente l'application de la \diamond -fermeture à un RCQ, proposée par Vilain et Kautz [VILAIN & KAUTZ 1986] dans le cadre de formalismes qualitatifs. L'objectif est, pour tout triplet de variables, d'appliquer l'opération de *triangulation* $Q_{(v_i, v_j)} \leftarrow Q_{(v_i, v_j)} \cap (Q_{(v_k, v_j)} \diamond Q_{(v_k, v_i)})$. Cette opération est réitérée jusqu'à ce qu'il n'y ait plus de modifications dans le RCQ. Pour cela, toute paire de variables (v_i, v_j) est ajoutée dans une file, afin de vérifier leur \diamond -cohérence pour toute variable v_k distincte de v_i et v_j . Pour chaque triplet v_i, v_j, v_k est appliquée l'opération de fermeture. S'il y eu modification de $Q_{(v_i, v_j)}$, deux cas se présentent. Si la contrainte est vide, alors le RCQ n'est pas localement cohérent et donc incohérent. Dans le cas contraire, une relation atomique a été supprimée dans $Q_{(v_i, v_j)}$ lors de l'évaluation d'une variable v_k , il est alors nécessaire d'évaluer l'une des contraintes $Q_{(v_i, v_k)}$ ou $Q_{(v_k, v_j)}$, choisie arbitrairement. La complexité temporelle et spatiale dans le pire des cas de l'algorithme 8 est donc respectivement en $\mathcal{O}(|\mathcal{B}|n^3)$ et $\mathcal{O}(n^2)$.

Algorithme 8: Algorithme de \diamond -fermeture d'un RCQ binaire

Entrées : $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$: un réseau de contraintes qualitatives
Sortie : \diamond -fermeture(\mathcal{N})

```

1  début
2  |  $S \leftarrow \{(i, j) : 1 \leq i < j \leq n\}$ 
3  | tant que ( $S \neq \emptyset$ ) faire
4  | | Choisir et Supprimer  $(i, j) \in S$ 
5  | | pour chaque ( $k \in [1 ; n] : k \neq i \wedge k \neq j$ ) faire
6  | | | si (Réviser( $\mathcal{N}, v_i, v_j, v_k$ ) répond Oui) alors
7  | | | | si ( $c_{(v_k, v_j)} = \emptyset$ ) alors
8  | | | | | retourner  $\perp$ 
9  | | | | sinon
10 | | | | |  $S \leftarrow S \cup \{(k, j)\}$ 
11 | retourner  $\mathcal{P}$ 

```

```

12 fonction Réviser( $\mathcal{N}, v_i, v_j, v_k$ ) : Booléen
13 début
14 | Modification  $\leftarrow$  Non
15 |  $Q \leftarrow Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$ 
16 | si ( $Q \not\subseteq Q_{(v_i, v_j)}$ ) alors
17 | |  $Q_{(v_i, v_j)} \leftarrow Q \cap Q_{(v_i, v_j)}$ 
18 | | Modification  $\leftarrow$  Oui
19 | retourner Modification

```

L'analogie entre les formes de cohérence locale des RCD et des RCQ ne s'arrête pas à la cohérence de chemins et à la \diamond -cohérence. Cependant, l'utilisation de domaines non finis dans le cadre qualitatif peut être une difficulté à la définition d'autres formes de cohérence locale. Dans le cadre des RCD, plusieurs formes de cohérence locale s'appuient sur les domaines. Puisqu'il n'y a pas de manipulation des domaines dans le cadre qualitatif, il faut se concentrer sur la manipulation de tuples. Nous pouvons remarquer, lorsqu'une relation atomique n'est pas \diamond -cohérente, que c'est en réalité l'ensemble des tuples qu'elle décrit qui ne le sont pas. Sur le même schéma, il est possible d'imaginer une forme de k -cohérence. Cette approche nécessiterait l'utilisation d'une table à $k - 1$ dimensions permettant d'associer à tout $(k - 1)$ -uplet de relations atomiques un RCQ admissible de $k - 1$ variables. Cette approche pourrait amener de grosses difficultés dans sa mise en œuvre. Ceci explique que la \diamond -cohérence soit la forme de cohérence locale essentiellement utilisée.

Nous avons défini un outil algorithmique permettant de vérifier la cohérence d'un scénario pour cer-

tains formalismes qualitatifs. Pour la construction d'un scénario, nous exploitons les notations des RCD, en substituant variable avec relation associée à une contrainte qualitative et valeur avec relation atomique. Ainsi, $Q_{(v_i, v_j)} = \{r\}$ représente l'affectation de la relation atomique r à $Q_{(v_i, v_j)}$, et $Q_{(v_i, v_j)} \neq r$ représente la réfutation de la relation atomique r dans $Q_{(v_i, v_j)}$. Pour tout RCQ $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$, $\mathcal{N}_{|Q_{(v_i, v_j)} = \{r\}}$ décrit un RCQ $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ tel que $Q'_{(v_i, v_j)} = \{r\}$ et $Q'_{(v_k, v_l)} = Q_{(v_k, v_l)}$ pour tout $k \neq i$ et $l \neq j$. L'algorithme 9 présente la vérification de la cohérence d'un RCQ lorsque la \diamond -cohérence est suffisante pour décider de la cohérence d'un scénario. L'approche est analogue à l'algorithme MAC avec branchement binaire pour les RCD. À chaque niveau de l'arbre de recherche, un couple de variables (v_i, v_j) est choisi, et une relation atomique r est choisie dans $Q_{(v_i, v_j)}$ afin de construire un scénario affectation par affectation. La fermeture par composition faible est maintenue après chaque affectation et chaque réfutation (dénotée par $\diamond(\mathcal{N})$).

Algorithme 9: Recherche d'un scénario cohérent dans un RCQ

Entrées : $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$: un réseau de contraintes
Sortie : Oui, si \mathcal{N} est cohérent, Non sinon.

```

1  début
2  |  $\mathcal{N}' \leftarrow \diamond(\mathcal{N})$ 
3  | si  $(\mathcal{N}' = \perp)$  alors
4  | | retourner Non
5  | sinon
6  | | retourner Recherche( $\mathcal{N}'$ )

```

```

7  fonction Recherche( $\mathcal{N}$ ) : Booléen
8  début
9  | si ( $\mathcal{N}$  est un scénario) alors
10 | | retourner Oui
11 | sinon
12 | |  $(v_i, v_j) \leftarrow$  Choisir un couple de variables dans  $v$ 
13 | |  $r \leftarrow$  Choisir une relation atomique dans  $Q_{(v_i, v_j)}$ 
14 | |  $\mathcal{N}' \leftarrow \diamond(\mathcal{N}_{|Q_{(v_i, v_j)} = \{r\}, Q_{(v_j, v_i)} = \{r^{-1}\}})$ 
15 | | si  $(\mathcal{N}' \neq \perp)$  et (Recherche( $\mathcal{N}'$ ) répond Oui) alors
16 | | | retourner Oui
17 | | sinon
18 | | |  $\mathcal{N} \leftarrow \diamond(\mathcal{N}_{|Q_{(v_i, v_j)} \neq r, Q_{(v_j, v_i)} \neq r^{-1}})$ 
19 | | | si  $(\mathcal{N} = \perp)$  alors
20 | | | | retourner Non
21 | | | sinon
22 | | | | Aller Ligne : 9

```

Pour faciliter la recherche d'un scénario \diamond -cohérent, l'une des approches consiste à réduire l'espace de recherche à évaluer, comme dans le cadre des RCD. Bien que la caractérisation de forme de cohérences locales au-delà de la \diamond -cohérence soit difficile à manipuler, il est possible d'exploiter un pré-traitement s'apparentant à l'arcs-cohérence de singleton. Cela consiste à affecter une relation atomique à une $Q_{(v_i, v_j)}$ puis à vérifier la \diamond -cohérence. Si le RCQ résultant n'est pas localement cohérent, alors la relation atomique n'appartient pas à un scénario cohérent et celle-ci est réfutée de $Q_{(v_i, v_j)}$. Dans le cas contraire, une autre relation atomique est affectée, et le procédé est réitéré jusqu'à atteindre un point fixe.

Comme pour les RCD, l'heuristique de choix de contraintes à affecter est très importante, et la recherche peut être guidée par les conflits. Puisque la \diamond -fermeture est maintenue, il est possible de pondérer les contraintes durant la recherche à la manière de *wdeg*. Le choix peut donc naturellement se porter vers

la contrainte mise en cause le plus souvent dans l'incohérence locale par composition faible pour adopter une stratégie d'échec d'abord en choisissant de réfuter rapidement les sous-problèmes incohérents (potentiels). La technique des redémarrages et l'analyse de conflits peuvent alors avoir leur importance afin d'accentuer cette approche [WESTPHAL *et al.* 2010]. Bien qu'elles nécessitent une adaptation causée par la difficulté à manipuler les domaines, de nombreuses techniques d'inférence du cadre des RCD peuvent être adaptées au cadre des RCQ.

D'autres améliorations des algorithmes de recherche ont spécialement été définies pour les RCQ en exploitant les propriétés de certains formalismes. Nebel [NEBEL 1997] propose, pour l'algèbre des intervalles, un algorithme de recherche pour lequel la relation caractérisant une contrainte est découpée en un ensemble de relations complexes particulières, appelé *split* $\subseteq 2^{\mathcal{B}_{int}}$.

Définition 41 (Ensemble de relations traitable) Soient \mathcal{F} un formalisme qualitatif et $R \subseteq 2^{\mathcal{B}}$. R est un ensemble de relations traitable pour \mathcal{F} si et seulement si pour tout RCQ $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ construit sur \mathcal{F} tel que toute contrainte Q de \mathcal{Q} vérifie $Q \in R$, il existe un algorithme polynomial en fonction de $|\mathcal{V}|$ permet de répondre à la cohérence de \mathcal{P} .

split est un ensemble de relations traitable, la \diamond -cohérence permet de répondre à la cohérence de tout RCQ composé uniquement des relations de cet ensemble. L'intérêt de cette approche réside dans le choix heuristique effectué avant affectation. Avec un algorithme classique, dans le pire des cas, 13 relations atomiques peuvent être choisies dans $Q_{(v_i, v_j)}$ afin d'être affectée. Avec l'approche de Nebel, $Q_{(v_i, v_j)}$ est découpée en plusieurs relations complexes de l'ensemble *split*, et le choix se porte parmi ce découpage. Cette approche réduit considérablement le facteur de branchement, c'est-à-dire le nombre de choix à effectuer avant d'effectuer un retour-arrière. L'algorithme 10 intègre cette amélioration.

Algorithme 10: Recherche d'un scénario cohérent dans un RCQ avec division des contraintes

Entrées : $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$: un réseau de contraintes
Sortie : \mathcal{N} modélise-t-il un problème satisfiable ?

```

1  début
2  |  $\mathcal{N}' \leftarrow \diamond(\mathcal{N})$ 
3  | si  $\mathcal{N}' = \perp$  alors
4  | | retourner Non
5  | sinon
6  | | retourner Recherche( $\mathcal{N}'$ )

7  fonction Recherche( $\mathcal{N}$ ) : Booléen
8  début
9  | si ( $\mathcal{N}$  est un scénario) alors
10 | | retourner Oui
11 | sinon
12 | |  $(v_i, v_j) \leftarrow$  Choisir un couple de variables dans  $\mathcal{V}$ 
13 | | Découper  $Q_{(v_i, v_j)}$  en  $\{R_1, \dots, R_k\} \subseteq 2^{\mathcal{B}}$  tel que  $R_l \in \text{split}$ , pour tout  $l \in [1; k]$ 
14 | | pour chaque ( $l \in [1; k]$ ) faire
15 | | |  $\mathcal{N}' \leftarrow \diamond(\mathcal{N}_{|_{Q_{(v_i, v_j)}=R_l, Q_{(v_j, v_i)}=R_l^{-1}}})$ 
16 | | | si ( $\mathcal{N}' \neq \perp$ ) et (Recherche( $\mathcal{N}'$ ) répond Oui) alors
17 | | | | retourner Oui
18 | | | retourner Non

```

Condotta et Saade [CONDOTTA *et al.* 2007b, SAADE 2008] propose de figer certaines relations dans un RCQ en exploitant les relations de l'ensemble *split*. Ils définissent la notion de contraintes *éligibles*

et de contraintes *gelées*, permettant respectivement de caractériser des contraintes qui ne doivent pas être évaluée et des contraintes pour lesquelles il n'y a pas la nécessité de les mettre à jour. Définir ces ensembles de relations traitables a donc un intérêt pratique certain pour répondre au problème de la cohérence des RCQ. S'il n'est plus nécessaire de se restreindre à un scénario pour vérifier la cohérence, il est possible d'adapter les algorithmes de recherches.

2.3.5 Les applications

Pour répondre au problème de la cohérence des RCQ, de nombreuses applications, appelées *solveurs*, ont été développées. [VAN BEEK & MANCHAK 1996] adaptent les techniques algorithmiques proposées pour les RCD pour la résolution de RCQ construits sur l'algèbre des intervalles. [NEBEL 1997] se concentre également sur l'algèbre des intervalles, mais il met en avant l'exploitation des classes traitables. [RENZ & NEBEL 2001] définissent une approche similaire pour le formalisme qualitatif RCC8, en adaptant toutes les approches proposées pour l'algèbre des intervalles. L'inconvénient de ces solveurs est qu'ils ne traitent que des formalismes qualitatifs binaires très spécifiques.

Plus récemment, [WALLGRÜN *et al.* 2007] ont proposé Spar-Q (pour *Spatial reasoning done Qualitatively*) et [GANTNER *et al.* 2008] ont développé GQR (pour *Generic Qualitative Reasoner*), deux solveurs génériques proposant de nombreux formalismes qualitatifs et offrant la possibilité d'intégrer simplement les techniques de résolution.

Pour réaliser nos études expérimentales, nous utiliserons QAT (pour *Qualitative Algebra Toolkit*) [CONDOTTA *et al.* 2006b]. QAT est un solveur générique très complet développé en langage JAVA. Il offre la possibilité d'utiliser n'importe quel formalisme qualitatif (temporels ou spatiaux), quelle que soit l'arité, à la seule condition que la définition des formalismes qualitatifs soient respectée. Il implémente de nombreuses techniques algorithmiques (heuristiques, algorithme de propagation de contraintes, exploitation des classes traitables), et permet d'en définir facilement de nouvelles. Il offre un outil intéressant pour la manipulation de RCQ.

2.4 Quelques fragments traitables de l'algèbre des intervalles

Vérifier la cohérence de connaissances exprimées à l'aide de l'algèbre des intervalles est un problème NP-complet dans le cas général. Néanmoins, nous avons énoncé précédemment qu'un scénario, représentant un RCQ pour lequel chaque contrainte décrit une relation atomique, présente une propriété intéressante. Un scénario est \diamond -cohérent si et seulement si il est globalement cohérent. En d'autres termes, l'ensemble des relations complexes composé uniquement de relations singletons forme un ensemble de relations, dit *traitables*. Il est possible de décider de la cohérence en temps polynomial de tout RCQ dont les contraintes qualitatives sont décrites par des relations d'un ensemble de relations traitable. Déterminer de tels ensembles de relations peut permettre de faciliter la recherche [NEBEL 1997, CONDOTTA *et al.* 2007b, SAADE 2008]. Pour comprendre ce qu'est un ensemble de relations traitable, étudions les approches permettant d'en caractériser certains. Nous nous focaliserons sur des ensembles de l'algèbre des intervalles contenant les relations singletons et la relation totale.

2.4.1 Relations pointisables

Vilain et Kautz [VILAIN & KAUTZ 1986] montrent que la \diamond -fermeture est complète pour décider de la cohérence de tout RCQ construit sur l'algèbre des points. Ainsi, il existe un algorithme en $\mathcal{O}(n^3)$ permettant de répondre au problème de satisfiabilité de tout problème modélisé dans cette algèbre. Puisque l'ensemble $2^{\mathcal{B}_{pt}}$ forme un ensemble de relations traitable, l'algèbre des points toute entière est dite *traitable*. Van Beek et Cohen [VAN BEEK & COHEN 1990] proposent un algorithme en $\mathcal{O}(n^2)$ permettant

de répondre au problème de la cohérence d'un RCQ de l'algèbre des points. Pour cela, ils transforment le RCQ construit sur l'algèbre des points en un graphe orienté étiqueté des relations $\{=\}$, $\{<\}$ et $\{\leq\}$, puis ils déterminent les composantes fortement connexes de ce graphe afin de les factoriser et de les trier. Pour l'algèbre des intervalles, Valdès-Pérez [VALDÉS-PÉREZ 1987] énonce que, bien qu'il soit possible de vérifier la cohérence d'un scénario à l'aide de la \diamond -cohérence, il est également possible de vérifier sa cohérence en $\mathcal{O}(n^2)$, par la construction d'un graphe sur les bornes des intervalles.

En se basant sur ces études, van Beek et Cohen [VAN BEEK & COHEN 1990] définissent un ensemble de relations traitable de l'algèbre des intervalles : les *relations pointisables*. Puisqu'un intervalle x est décrit par deux bornes $[x^- ; x^+]$ telles que $x^- < x^+$, une approche naturelle basée sur la définition des relations atomiques (Tableau 2.5) consiste à transformer, en temps polynomial dans le nombre de variables, un RCQ de l'algèbre des intervalles en RCQ sur l'algèbre des points. Le tableau 2.6 présente la *pointisation* des relations atomiques de l'algèbre des intervalles. Chaque relation atomique décrit la position relative entre deux intervalles $x = [x^- ; x^+]$ et $y = [y^- ; y^+]$. Pour chaque relation atomique $r \in \mathcal{B}_{int}$ telle que $(x, y) \in r$, les relations atomiques $r^{--} \in \mathcal{B}_{pt}$, $r^{-+} \in \mathcal{B}_{pt}$, $r^{+-} \in \mathcal{B}_{pt}$ et $r^{++} \in \mathcal{B}_{pt}$ décrivent les positions relatives entre les bornes (x^-, y^-) , (x^-, y^+) , (x^+, y^-) et (x^+, y^+) respectivement. Par exemple, si un intervalle x rencontre l'intervalle y , alors $x^- < y^-$, $x^- < y^+$, $x^+ = y^-$ et $x^+ < y^+$.

Algèbre des intervalles		Algèbre des points			
Description	Relation	$(x^-, y^-) \in r^{--}$	$(x^-, y^+) \in r^{-+}$	$(x^+, y^-) \in r^{+-}$	$(x^+, y^+) \in r^{++}$
x est avant y	b	<	<	<	<
x rencontre y	m	<	<	=	<
x chevauche y	o	<	<	>	<
x commence y	s	=	<	>	<
x contient y	d	>	<	>	<
x fini y	f	>	<	>	=
x est égal à y	eq	=	<	>	=
x est fini par y	fi	<	<	>	=
x contient y	di	<	<	>	>
x est commencé par y	si	=	<	>	>
x est chevauché par y	oi	>	<	>	>
x est rencontré par y	mi	>	=	>	>
x est après y	a	>	>	>	>

TABLE 2.6 – Pointisation des relations atomiques de l'algèbre des intervalles

La pointisation d'une relation complexe de l'algèbre des intervalles est définie par l'union des relations atomiques de l'algèbre des points obtenues par pointisation sur les bornes des intervalles. Le tableau 2.7 illustre la pointisation de la relation $\{m, fi, di\} \in 2^{\mathcal{B}_{int}}$. Chaque relation atomique qui la compose est pointisée, et l'union du résultat de la pointisation pour chaque borne décrit les relations entre les bornes pour cette relation complexe.

Algèbre des intervalles	Algèbre des points				
	Relation	$(x^-, y^-) \in r^{--}$	$(x^-, y^+) \in r^{-+}$	$(x^+, y^-) \in r^{+-}$	$(x^+, y^+) \in r^{++}$
m		<	<	=	<
fi		<	<	>	=
di		<	<	>	>
$\{m, fi, di\}$		$\{<\}$	$\{<\}$	$\{=, >\}$	$\{<, =, >\}$

 TABLE 2.7 – Pointisation de la relation complexe $\{m, fi, di\}$ de l'algèbre des intervalles

L'opération de pointisation peut être appliquée à toute relation $R \in 2^{\mathcal{B}_{int}}$, mais toute relation n'est

pas pointisable. Pour deux relations complexes différentes, l'opération de pointisation peut donner des résultats identiques. Cependant, il est nécessaire de garantir qu'une seule relation de $2^{\mathcal{B}_{int}}$ est associée à un ensemble de relations entre les bornes de deux intervalles.

Définition 42 (Relation pointisable) Une relation $R \in 2^{\mathcal{B}_{int}}$ est pointisable si et seulement si, pour toute relation $R' \in 2^{\mathcal{B}_{int}}$ telle que $R' \not\subseteq R$, le résultat de l'opération pointisation sur R diffère du résultat de l'opération pointisation sur R' .

Le tableau 2.8 illustre l'opération de pointisation sur la relation $\{m, o, fi, di\}$. Le résultat est identique à celui présenté Tableau 2.8 sur la relation $\{m, fi, di\}$. Puisque $\{m, o, fi, di\} \not\subseteq \{m, fi, di\}$, nous pouvons affirmer que $\{m, fi, di\}$ n'est pas une relation pointisable. À l'inverse, il n'existe pas de relations $R \in 2^{\mathcal{B}_{int}}$, telles que $R \not\subseteq \{m, o, fi, di\}$, permettant d'obtenir le même résultat par l'opération de pointisation de R . Par conséquent, $\{m, o, fi, di\}$ est une relation pointisable.

Algèbre des intervalles	Algèbre des points			
	Relation	$(x^-, y^-) \in r^{--}$	$(x^-, y^+) \in r^{-+}$	$(x^+, y^-) \in r^{+-}$
o	<	<	>	<
m	<	<	=	<
fi	<	<	>	=
di	<	<	>	>
{m, o, fi, di}	{<}	{<}	{=, >}	{<, =, >}

TABLE 2.8 – Pointisation de la relation complexe $\{m, o, fi, di\}$ de l'algèbre des intervalles

L'ensemble des relations pointisables, noté \mathcal{P} , est donc le plus grand sous-ensemble de $2^{\mathcal{B}_{int}}$ contenant uniquement des relations pointisables. En considérant 8 relations complexes de l'algèbre des points, 4 positions relatives entre les bornes des intervalles, il existe 4096 possibilités pour mettre en relation deux intervalles à l'aide des bornes. Certaines ne peuvent trivialement pas être étendue à une relation de $2^{\mathcal{B}_{int}}$, et d'autres ne correspondent pas à des relations pointisables. Au total, 188 des 8192 relations de $2^{\mathcal{B}_{int}}$ composent l'ensemble \mathcal{P} . L'ensemble des relations \mathcal{P} est appelé une sous-classe de $2^{\mathcal{B}_{int}}$ car il est stable par inverse, intersection et composition dans $2^{\mathcal{B}_{int}}$. La sous-classe de relations \mathcal{P} est donc appelée *classe traitable* pour l'algèbre des intervalles.

Définition 43 (RCQ pointisable) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ construit sur l'algèbre des intervalles. \mathcal{N} est un RCQ pointisable si et seulement si $Q_{(v_i, v_j)} \in \mathcal{P}$ pour tout $v_i, v_j \in \mathcal{V}$.

L'opération permettant de transformer un RCQ pointisable en RCQ construit sur l'algèbre des points est une opération polynomiale en temps dans la taille du RCQ initial. Comme énoncé précédemment, décider de la cohérence d'un RCQ construit sur l'algèbre des points est également une opération polynomiale en temps dans la taille de ce RCQ. Il existe donc un algorithme polynomial permettant de répondre au problème de la cohérence d'un RCQ pointisable.

Propriété 3 Soit \mathcal{N} un RCQ construit sur l'algèbre des intervalles. Si \mathcal{N} est pointisable, alors il existe un algorithme polynomial en temps dans la taille de \mathcal{N} permettant de décider de la cohérence de \mathcal{N} .

2.4.2 Relations ORD-Horn

Nebel et Bürckert [NEBEL & BÜRCKERT 1995] proposent de décrire un RCQ construit sur l'algèbre des intervalles en manipulant également les bornes des intervalles au travers des contraintes linéaires \leq ,

= et \neq entre les bornes. Ainsi, les relations atomiques $a < b$, $a = b$ et $a > b$ de l'algèbre des points peuvent s'écrire par les contraintes « $a \leq b$ et $a \neq b$ », « $a = b$ » et « $b \leq a$ et $a \neq b$ » respectivement. En s'appuyant sur le tableau décrivant la pointisation 2.6 et en exploitant cette réécriture, il est possible de décrire toute relation atomique de l'algèbre des intervalles. Par exemple, considérons deux intervalles $x = [x^- ; x^+]$ et $y = [y^- ; y^+]$. Si « x rencontre y », alors $x^- = y^-$, $x^- < y^+$, $x^+ > y^-$ et $x^+ < y^+$ par la pointisation. Ce qui donne les contraintes $(x^- = y^-)$, $(x^- \leq y^+ \text{ et } x^- \neq y^+)$, $(y^- \leq x^+ \text{ et } x^+ \neq y^-)$ et $(x^+ \leq y^+ \text{ et } x^+ \neq y^+)$ pour la représentation de Nebel et Bürckert. Toute relation atomique de \mathcal{B}_{int} peut donc être associée à une conjonction de contraintes sur les bornes.

Une relation complexe R est une disjonction de relation atomique. Par conséquent, si deux intervalles x et y sont en relation par R, alors ils sont en relation par l'une des relations atomiques $r \in R$. La réécriture d'une relation complexe de $2^{\mathcal{B}_{int}}$ est donc la disjonction de contraintes linéaires. L'exemple 11 illustre la réécriture de la relation $\{b, o\}$. Le développement permet d'obtenir une conjonction de contraintes linéaires entre les bornes.

Exemple 11 *Considérons deux intervalles $x = [x^- ; x^+]$ et $y = [y^- ; y^+]$ mis en relation par $\{b, o\}$. En utilisant l'opération de pointisation sur b et o , nous obtenons.*

$$\begin{aligned} x \ b \ y : & (x^- < y^-), (x^- < y^+), (x^+ < y^-) \text{ et } (x^+ < y^+) \\ x \ o \ y : & (x^- < y^-), (x^- < y^+), (x^+ > y^-) \text{ et } (x^+ < y^+) \end{aligned}$$

Avec le système de réécriture défini, chaque relation atomique peut s'écrire :

$$\begin{aligned} x \ b \ y : & x^- \leq y^-, x^- \neq y^-, x^- \leq y^+, x^- \neq y^+, x^+ \leq y^-, x^+ \neq y^-, x^+ \leq y^+ \text{ et } x^+ \neq y^+ \\ x \ o \ y : & x^- \leq y^-, x^- \neq y^-, x^- \leq y^+, x^- \neq y^+, y^- \leq x^+, x^+ \neq y^-, x^+ \leq y^+ \text{ et } x^+ \neq y^+ \end{aligned}$$

Après développement, la disjonction de ces contraintes sur les bornes permet d'obtenir :

$$\{b, o\} \left\{ \begin{array}{l} x^- \leq y^-, x^- \neq y^-, \\ x^- \leq y^+, x^- \neq y^+, \\ x^+ \neq y^-, (x^+ \leq y^- \text{ ou } y^- \leq x^+), \\ x^+ \leq y^+, x^+ \neq y^+ \end{array} \right.$$

Remarquons que la contrainte $x^+ \leq y^-$ ou $y^- \leq x^+$ entre les bornes y^- et x^+ est toujours vraie. Par conséquent, nous pouvons réécrire les contraintes sur les bornes de la façon suivante :

$$\{b, o\} \left\{ \begin{array}{l} x^- \leq y^-, x^- \neq y^-, \\ x^- \leq y^+, x^- \neq y^+, \\ x^+ \neq y^-, \\ x^+ \leq y^+, x^+ \neq y^+ \end{array} \right.$$

Nebel et Bürckert définissent les contraintes linéaires décrites uniquement par =, \leq et \neq comme des *contraintes ORD*. Une contrainte linéaire est une *contrainte Horn* lorsqu'elle ne contient au plus une contrainte \leq ou =. Une contrainte linéaire ORD et Horn est dite *ORD-Horn*. Dans l'exemple 11, la relation $\{b, o\}$ s'écrit comme une conjonction de contraintes ORD, et chacune s'écrit avec au plus une contrainte \leq ou =. Toutes les contraintes de cet exemple sont donc des contraintes ORD-Horn.

Définition 44 (Relation ORD-Horn) *Soit $R \in 2^{\mathcal{B}_{int}}$. R est une relation ORD – Horn si et seulement si R peut s'écrire comme une conjonction de contraintes ORD-Horn.*

L'ensemble des relations \mathcal{H} est un sous-ensemble de $2^{\mathcal{B}^{int}}$ contenant uniquement des relations ORD-Horn. \mathcal{H} forme une sous-classe de $2^{\mathcal{B}^{int}}$ puisqu'elle est stable par inverse, intersection et composition dans $2^{\mathcal{B}^{int}}$. $\mathcal{P} \subset \mathcal{H}$ puisque toute relation pointisable peut être transformée en une conjonction de contraintes ORD. La classe \mathcal{H} se compose de 868 relations, relations pointisables comprises.

Définition 45 (RCQ ORD-Horn) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ. \mathcal{N} est un RCQ ORD-Horn si et seulement si $Q_{(v_i, v_j)} \in \mathcal{H}$ pour toutes variables $v_i, v_j \in \mathcal{V}$.

Nebel et Bückert définissent la théorie ORD-Horn en proposant un schéma d'axiomes sur les contraintes linéaires (Tableau 2.9), exploitant les propriétés de transitivité, de réflexivité et d'antisymétrie sur les bornes des intervalles, puis en axiomatisant la relation d'équivalence $x = y$ entre deux bornes par $x \leq y$ et $y \leq x$.

$\forall x, y : x \leq y \wedge y \leq z$	\rightarrow	$x \leq z$	Transitivité
$\forall x : x \leq x$			Réflexivité
$\forall x, y : x \leq y \wedge y \leq x$	\rightarrow	$x = y$	Antisymétrie
$\forall x, y : x = y$	\rightarrow	$x \leq y$	
$\forall x, y : x = y$	\rightarrow	$y \leq x$	

TABLE 2.9 – Schéma d'axiomes de la théorie ORD

Les contraintes linéaires liées au schéma d'axiomes de la théorie ORD sont des contraintes ORD-Horn. Si les contraintes linéaires décrivant les relations d'un RCQ et la théorie ORD sont cohérentes, alors le RCQ admet un scénario cohérent. Vérifier la cohérence d'un système de contraintes linéaires composé uniquement de contraintes ORD-Horn est polynomial en temps. Or, la réécriture d'un RCQ ORD-Horn à l'aide d'un système de contraintes linéaires ORD-Horn est une opération polynomiale en temps dans le nombre de variables du RCQ. Il est donc possible de décider de la cohérence d'un RCQ ORD-Horn en temps polynomial dans le nombre de variables. \mathcal{H} est donc une classe de relations traitable de l'algèbre des intervalles.

Propriété 4 Soit \mathcal{N} un RCQ. Si \mathcal{N} est ORD-Horn, alors la \diamond -cohérence est complète pour décider de la cohérence de \mathcal{N} .

Notons que la sous-classe des relations ORD-Horn est la sous-classe de relations traitable la plus grande de l'algèbre des intervalles contenant les relations complexes de cardinalité 1 et la relation totale [NEBEL & BÜCKERT 1995, LIGOZAT 1998b]. Drakengren et Jonson [DRAKENGREN & JONSSON 1997] montrent qu'il existe des ensembles de relations traitables de cardinalités plus grandes, mais ne contenant pas toutes les relations atomiques. Pour s'en convaincre, nous pouvons construire une classe traitable très simple. Considérons l'ensemble de relations $R_{eq} \subseteq 2^{\mathcal{B}^{int}}$, telle que $\emptyset \in R_{eq}$ et $eq \in R$ pour tout $R \in R_{eq}$. Pour répondre au problème de la cohérence d'un RCQ construit sur les relations de R_{eq} , il existe un algorithme polynomial en temps dans la taille du problème en $\mathcal{O}(n^2)$ dans le pire des cas, nécessitant de parcourir chaque contrainte qualitative. S'il existe une contrainte vide, le RCQ est incohérent. Sinon, il existe au moins un scénario cohérent tel que tous les intervalles soient identiques. Ainsi R_{eq} est composé de 4097 relations et cet ensemble couvre donc plus de la moitié des relations de $2^{\mathcal{B}}$. Puisque eq est la relation d'identité, R_{eq} est stable par intersection, inverse et composition, elle est donc une sous-classe de relations traitable pour l'algèbre des intervalles. La taille d'une classe traitable ne peut pas être utilisée comme un argument unique pour justifier son exploitation dans les outils algorithmiques, puisque les informations temporelles qu'elle permet de décrire ne sont pas toujours pertinentes. Contrairement à R_{eq} , les relations de \mathcal{H} permettent de décrire tout scénario de l'algèbre des intervalles. Par conséquent elle peut être exploitée, quel que soit le RCQ, pour améliorer les algorithmes de recherches [NEBEL 1997, CONDOTTA *et al.* 2007b].

2.4.3 Relations convexes

Chaque relation atomique $r \in \mathcal{B}_{int}$ est associée à une position relative entre deux intervalles $X = [x^- ; x^+]$ et $Y = [y^- ; y^+]$. En fonction de la position relative de ces intervalles, l'opération de pointisation permet de décrire la position relatives entre les bornes. Pour les relations atomiques b , m et o , le tableau 2.6 ne montre une différence qu'entre les bornes x^+ et y^+ pour l'opération de pointisation. Intuitivement, ces relations atomiques montrent une certaine proximité. Sur ce constat, il est possible de caractériser un ordre sur les relations atomiques de l'algèbre des intervalles.

Un intervalle de référence $X_0 = [x_0^- ; x_0^+]$ avec $x_0^- < x_0^+$ est divisé en 5 zones (Figure 2.6), afin de comparer la position d'un point y par rapport à l'intervalle X . Si y se situe avant la borne inférieure de X , alors la valeur assignée est 0, si il est confondu avec la borne inférieure de X , alors sa valeur est 1, *etc.* Un couple d'entiers est alors attribué à chaque relation atomique $r \in \mathcal{B}_{int}$ en comparant la borne inférieure et la borne supérieure d'un intervalle $Y = [y^- ; y^+]$, tel que $(Y, X_0) \in r$. Considérons par exemple que Y commence X_0 . $(Y, X_0) \in \{s\}$ ssi $y^- = x_0^-$, ce qui correspond à la zone 1, et $x_0^- < y^+ < x_0^+$, ce qui correspond à la zone 2. En d'autres termes, le couple $(1, 2)$ est associé à la relation atomique s .

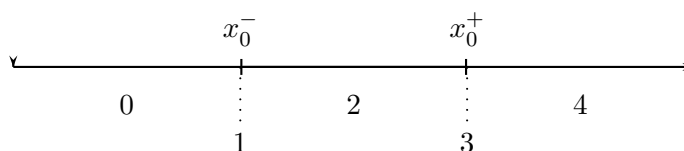


FIGURE 2.6 – Illustration des zones pour la comparaison des bornes avec un intervalle de référence $X_0 = [x_0^- ; x_0^+]$

La figure 2.7 illustre sur un plan les 13 relations atomiques de l'algèbre des intervalles. Les coordonnées de chaque relation atomique sont déduites par le couple d'entiers obtenu.

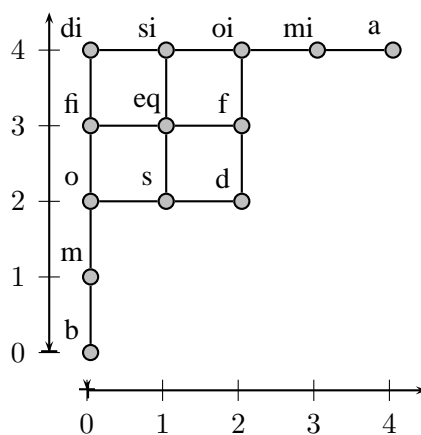


FIGURE 2.7 – Illustration graphique des relations atomiques de l'algèbre des intervalles

En s'appuyant sur ces couples d'entiers, il est ainsi possible de définir un ordre sur les relations atomiques de l'algèbre des intervalles [LIGOZAT 1991, LIGOZAT 1994, BESSIÈRE *et al.* 1996]. On admet qu'il existe un lien entre deux relations atomiques si les couples d'entiers de celles-ci ne diffèrent que pour l'une ou l'autre des bornes. Il s'ensuit un pré-ordre partiel \preceq qui peut être caractérisé comme suit :

Définition 46 (Ordre partielle sur les relations atomiques de l'algèbre des intervalles) Soient $r, r' \in \mathcal{B}_{int}$, (a, b) et (c, d) les couples d'entiers respectifs associés à r et r' . $r \preceq r'$ si et seulement si $a \leq c$ et

$b \leq d$.

En considérant l'ordre \preceq , \mathcal{B}_{int} admet pour infimum la relation atomique *précède* (b) et pour supremum la relation atomique *succède* (a). De plus, pour tous éléments $r, r', r'' \in \mathcal{B}_{int}$, \preceq vérifie les propriétés de réflexivité ($r \preceq r$), de transitivité ($(r \preceq r') \wedge (r' \preceq r'') \Rightarrow r \preceq r''$) et d'antisymétrie ($(r \preceq r') \wedge (r' \preceq r) \Leftrightarrow r = r'$) sur \mathcal{B}_{int} . Enfin, puisque tous éléments $r, r' \in \mathcal{B}_{int}$ admettent une borne supérieure $\inf\{r, r'\}$ ainsi qu'une borne inférieure $\sup\{r, r'\}$, nous pouvons constater que $(\mathcal{B}_{int}, \preceq)$ forme un treillis. Ce treillis est parfois appelé treillis conceptuel ou treillis qualitatif de l'algèbre des intervalles (Figure 2.8).

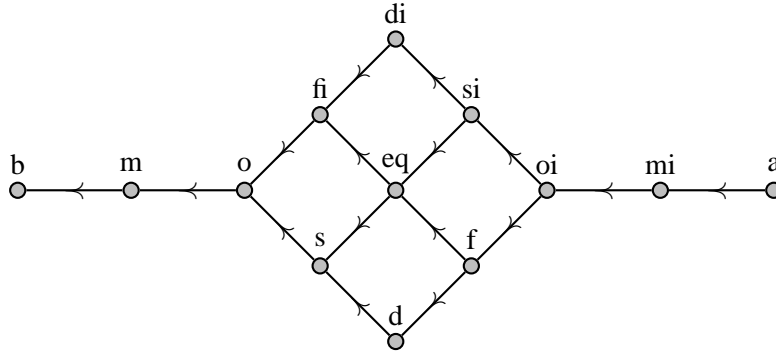


FIGURE 2.8 – Treillis conceptuel de l'algèbre des intervalles

À l'aide du treillis $(\mathcal{B}_{int}, \preceq)$, nous pouvons définir la notion d'*intervalles convexes* et de *relations convexes*.

Définition 47 (Intervalle convexe) Soient $r, r' \in \mathcal{B}_{int}$. L'intervalle convexe $[r ; r']$ est une relation de $2^{\mathcal{B}_{int}}$ tel que $[r ; r'] = \{r'' : r \preceq r'' \wedge r'' \preceq r'\}$.

Définition 48 (Relation convexe) Soit $R \in 2^{\mathcal{B}_{int}}$. R est une relation convexe si et seulement si elle peut s'écrire sous la forme d'un intervalle convexe.

La définition de la convexité met en avant un ensemble de relations particulier de l'algèbre des intervalles. L'ensemble des relations convexes de $2^{\mathcal{B}_{int}}$, noté \mathcal{C} , est composé de 82 relations complexes. Il est une sous-classe de $2^{\mathcal{B}_{int}}$ car il est stable par inverse, intersection et composition. \mathcal{C} est donc une classe traitable de l'algèbre des intervalles.

Définition 49 (RCQ convexe) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ. \mathcal{N} est un RCQ convexe si et seulement si $Q_{(v_i, v_j)} \in \mathcal{C}$ pour toutes variables $v_i, v_j \in \mathcal{V}$.

Notons que, quelle que soit la relation $R \in \mathcal{C}$, R peut s'écrire comme un ensemble de contraintes ORD-Horn, en temps polynomiale dans la taille de la relation. Puisque décider de la cohérence d'un RCQ ORD-Horn peut être réalisé par un algorithme polynomiale en temps dans la taille du RCQ, nous en déduisons qu'il est possible de décider de la cohérence de tout RCQ convexe en temps polynomiale dans la taille du RCQ. Notons qu'une autre preuve se concentre sur la stabilité de \mathcal{C} et le fait que la composition faible soit distributive sur l'intersection. Il est donc possible de répondre à la cohérence d'un RCQ convexe par un algorithme de fermeture par composition faible.

Propriété 5 Soit \mathcal{N} un RCQ. Si \mathcal{N} est convexe, alors il existe un algorithme polynomiale en temps dans la taille de \mathcal{N} permettant de décider de la cohérence de \mathcal{N} .

2.4.4 Relations préconvexes

Ligozat [LIGOZAT 1991, LIGOZAT 1994, BESSIÈRE *et al.* 1996] étend la notion de convexité afin de définir la préconvexité. Pour cela, il s'appuie dans un premier temps sur une représentation graphique des relations atomiques définie par Nökel [NÖKEL 1991]. Nökel illustre l'ordre sur les relations atomiques graphiquement en définissant une notion de région. Chaque point d'un plan $(y^-, y^+) \in \mathbb{R}^2$ est associé à un intervalle $Y = [y^- ; y^+]$. L'abscisse coïncide donc avec la borne inférieure y^- et l'ordonnée coïncide avec la borne supérieure y^+ . Un intervalle de référence $X_0 = [x_0^- ; x_0^+]$ est défini et permet de déduire tout intervalle $Y = [y^- ; y^+]$ en relation avec X_0 par une relation atomique de \mathcal{B}_{int} .

Définition 50 (Région) Soit $r \in \mathcal{B}_{int}$ et $X_0 = [x_0^- ; x_0^+]$ un intervalle de référence sur \mathbb{R} . $Reg(r, X_0) = \{(y^-, y^+) \in \mathbb{R}^2 : ([x_0^- ; x_0^+], [y^- ; y^+]) \in r\}$.

La figure 2.9 illustre les relations atomiques par la représentation de Nökel. Prenons par exemple la relation atomique « commence » (s). La région associée est $Reg(s, X_0) = \{(y^-, y^+) \in \mathbb{R}^2 : ([x_0^- ; x_0^+], [y^- ; y^+]) \in \{s\}\}$ Ce qui correspond à $Reg(s, X_0) = \{(y^-, y^+) \in \mathbb{R}^2 : (x_0^- = y^-, (x_0^+ < y^+))\}$. Pour tout $R \in 2^{\mathcal{B}_{int}}$, nous avons donc $Reg(R, X_0) = \bigcup_{r \in R} Reg(r, X_0)$.

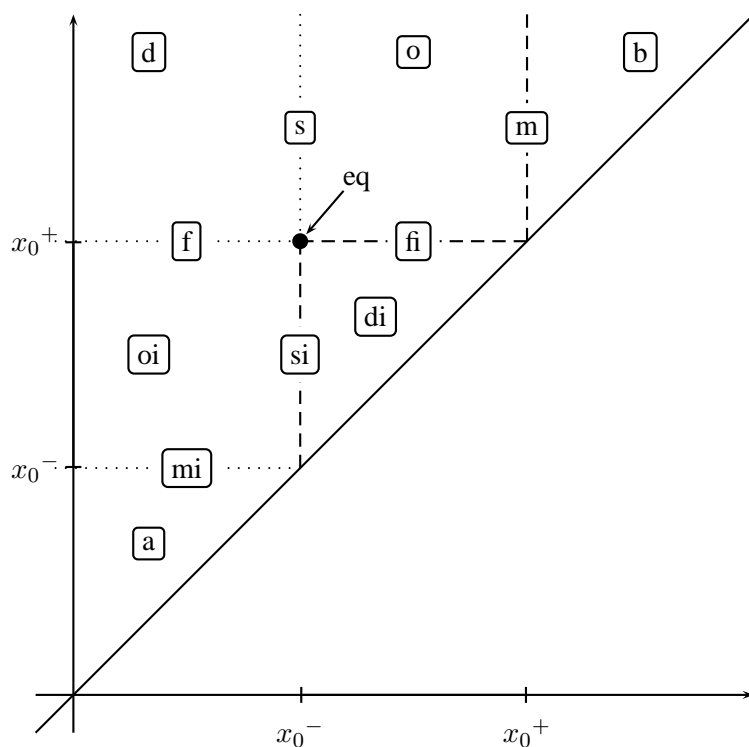


FIGURE 2.9 – Représentation des relations atomiques de l'algèbre des intervalles dans le plan.

Ligozat adapte la représentation de Nökel, afin de définir les notions de *dimension*, de *fermeture topologique* et de *fermeture convexe* d'une relation. La dimension d'une relation atomique est la dimension géométrique de la région par la représentation de Nökel. Celle-ci peut également être vue comme le nombre de bornes distinctes avec l'intervalle de référence X_0 .

Définition 51 (Dimension) Soient $r \in \mathcal{B}_{int}$ et $R \in 2^{\mathcal{B}_{int}}$:

- la dimension de r , notée $dim(r)$, est la dimension de la région $Reg(r, X_0)$;
- $dim(\emptyset) = -1$;

$$- \dim(R) = \max\{\dim(r) : r \in R\}.$$

Ligozat définit ensuite la fermeture topologique d'une relation atomique. Cette notion s'approche de la notion de fermeture ou d'adhérence d'un espace topologique. La fermeture d'une relation atomique r est donc une relation complexe R dont la région $Reg(R, X_0)$ est la fermeture de $Reg(r, X_0)$. En d'autres termes, en partant de la région d'une relation atomique r , il suffit de considérer les régions voisines de dimension strictement inférieure afin de fermer la région de r .

Définition 52 (Fermeture topologique) Soient $r \in \mathcal{B}$ et $R \in 2^{\mathcal{B}_{int}}$.

- la fermeture topologique de r , notée $C(r)$, est la relation $R' \in 2^{\mathcal{B}}$ telle que $Reg(R, X_0)$ est la fermeture topologique de $Reg(r, X_0)$;
- $C(R) = \bigcup\{C(r) : r \in R\}$.

Le tableau 2.10 énumère les relations atomiques de l'algèbre des intervalles, leur dimension et leur fermeture topologique.

$r \in \mathcal{B}$	$\dim(r)$	$C(r)$	$r \in \mathcal{B}$	$\dim(r)$	$C(r)$
b	2	$\{b, m\}$	a	2	$\{mi, a\}$
m	1	$\{m\}$	mi	1	$\{mi\}$
o	2	$\{m, o, s, fi, eq\}$	oi	2	$\{eq, f, si, oi, mi\}$
s	1	$\{s, eq\}$	si	1	$\{si, eq\}$
fi	1	$\{fi, eq\}$	f	1	$\{f, eq\}$
d	2	$\{s, d, eq, f\}$	di	2	$\{si, di, eq, fi\}$
eq	0	$\{eq\}$			

TABLE 2.10 – Dimension et fermeture topologique des relations atomiques de l'algèbre des intervalles

Enfin, Ligozat introduit la notion de *fermeture convexe* concernant les relations de $2^{\mathcal{B}_{int}}$.

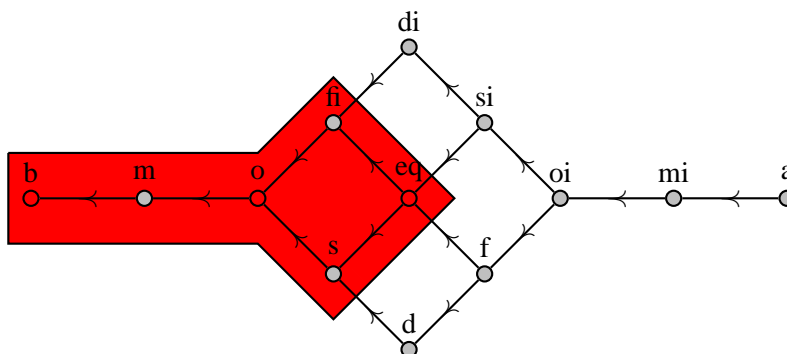
Définition 53 (Fermeture convexe) Soit $R \in 2^{\mathcal{B}}$. La fermeture convexe de R , notée $I(R)$, est la plus petite relation convexe de $2^{\mathcal{B}}$ contenant r .

À travers ces notions, Ligozat définit les critères permettant de déduire la préconvexité d'une relation.

Définition 54 Soit $R \in 2^{\mathcal{B}}$. R est préconvexe si et seulement si elle satisfait l'un des points suivant :

1. $C(R)$ est une relation convexe ;
2. $I(R) \subseteq C(R)$;
3. $\dim(I(R) \setminus R) < \dim(R)$;
4. $I(R) \setminus R$ ne contient que des relations atomiques de dimension strictement inférieure à 2 (c'est-à-dire un sous-ensemble de $\{m, s, fi, eq, f, si, mi\}$).

Prenons l'exemple de la relation $\{b, o, eq\}$. $\dim(\{b, o, eq\}) = 2$ puisque la dimension est définie par $\max\{\dim(b), \dim(o), \dim(eq)\} = \max\{2, 0, 2\} = 2$. La fermeture topologique $C(\{b, o, eq\})$ correspond à $C(b) \cup C(o) \cup C(eq)$. D'après le tableau 2.10, $\{b, o, eq\} = \{b, m\} \cup \{m, o, s, fi, eq\} \cup \{eq\} = \{b, m, o, s, fi, eq\}$. Enfin, la fermeture convexe de $\{b, o, eq\}$ est $I(\{b, o, eq\}) = \{b, m, o, s, fi, eq\}$ (Figure 2.10). La relation $\{b, o, eq\}$ est préconvexe si et seulement si elle satisfait l'un des points décrits Définition 54. $C(\{b, o, eq\}) = [b ; eq]$, donc $C(\{b, o, eq\})$ est bien une relation convexe. Le point $I(\{b, o, eq\}) \subseteq C(\{b, o, eq\})$ est également vérifié. De plus, $I(\{b, o, eq\}) \setminus \{b, o, eq\} = \{m, s, fi\}$ et $\dim(\{m, s, fi\}) = 1$, donc $\dim(I(\{b, o, eq\}) \setminus \{b, o, eq\}) < \dim(\{b, o, eq\})$.

FIGURE 2.10 – Fermeture convexe de la relation $\{b, m, o\}$

Définition 55 (RCQ préconvexe) Soit \mathcal{P} un RCQ construit sur l'algèbre des intervalles. \mathcal{P} est préconvexe si et seulement si $Q_{(v_i, v_j)}$ est une relation préconvexe pour toutes variables $v_i, v_j \in \mathcal{V}$.

Les critères présentés Définition 54 sont tous équivalents, et Ligozat montre que l'ensemble des relations préconvexes coïncide avec la classe \mathcal{H} . Il est donc possible de répondre au problème de la cohérence d'un RCQ préconvexe en temps polynomial dans la taille du RCQ. En comparaison à la théorie ORD développée par Nebel et Bückert, les conditions de préconvexité de la définition de Ligozat présente une certaine facilité de manipulation. En effet, dans le cadre des relations ORD-Horn, il est nécessaire de développer des contraintes linéaires sur les bornes des intervalles par distributivité et de réduire certaines contraintes triviale avant de vérifier si celles-ci sont ORD-Horn. Dans le cadre des relations préconvexes, si les dimensions sont données, alors le calcul d'une relation préconvexe par l'un des points de la définition 54 est immédiate. De plus, à l'aide du treillis de l'algèbre des intervalles et du tableau 2.10 pour la fermeture topologique des relations atomiques, il est également facile de calculer la fermeture convexe et la fermeture topologique d'une relation. Les preuves analytiques en sont également simplifiées et Ligozat montre qu'il est possible de décider de la cohérence d'un RCQ préconvexe en utilisant la fermeture par composition faible.

Propriété 6 Soit \mathcal{P} un RCQ préconvexe. \mathcal{P} est cohérent si et seulement si \mathcal{P} est \diamond -cohérent.

2.5 Conclusion

Dans ce chapitre, nous avons présenté la modélisation de connaissances temporelles et spatiales à l'aide de différents formalismes qualitatifs. La représentation des connaissances sur des domaines non finis peut dépasser les limites des technologies actuelles, et ainsi rendre la modélisation difficile à manipuler dans un cadre quantitatif. Appréhender le problème de manière qualitative présente un avantage de la simplicité et de la compacité de la représentation. Ainsi, les connaissances modélisées à l'aide d'un formalisme qualitatifs permettent de décrire des entités et des relations entre celles-ci. Les RCQ constituent un modèle de représentation idéal pour y greffer les connaissances, afin de répondre aux problèmes de décisions ou d'optimisations très étudiées en intelligence artificielle, tels que les problèmes CSP ou MAXCSP. Le parallèle est immédiat entre RCQ et RCD, cependant la manipulation de domaines non finis nécessite d'adapter des outils algorithmiques spécifiques au cadre qualitatif. Les formalismes qualitatifs présentent un cadre théorique solide, sur lequel il est possible de s'appuyer pour le développement de techniques de recherche ou d'inférence. Ainsi, de nombreux outils ont été développés afin de réduire l'arbre de recherche parcouru : la cohérence locale, les choix heuristiques, l'analyse de conflits, *etc.* Une autre approche a permis de faire évoluer les algorithmes de recherche classique en utilisant la notion

de classes traitables. La définition des classes de relations ORD-Horn et préconvexes offre des outils théoriques intéressants, à la base de nombreuses améliorations pratiques.

Chapitre 3

Formules propositionnelles

Sommaire

3.1	Introduction	64
3.2	La logique propositionnelle	65
	3.2.1 Syntaxe : le langage propositionnel	65
	3.2.2 Sémantique : la satisfiabilité	67
3.3	Techniques algorithmiques	68
	3.3.1 Algorithmes de recherche et propagation	68
	3.3.2 Simplification de formules propositionnelles	69
	3.3.3 Outils algorithmiques modernes	70
	3.3.4 Classes traitables	71
3.4	Conclusion	72

3.1 Introduction

La modélisation des connaissances prend son essor au fil du développement des technologies informatiques. Initialement, dans le but d'exploiter les outils informatiques, le cadre adopté utilise une représentation binaire des informations pour s'approcher du langage machine. Le langage de la logique propositionnelle permet de décrire ces informations. Celui-ci se définit sur un ensemble de propositions pouvant être « vraie » ou « fausse » et de connecteurs logiques entre ces propositions, afin de construire des expressions, appelées formules propositionnelles. Il est possible de décrire les formules propositionnelles sous plusieurs formes, telles que la forme normale conjonctive (CNF), la forme normale disjonctive (DNF), *etc.* D'un point de vue sémantique, une formule propositionnelle est une application mathématiques prenant pour valeur « vrai » ou « faux ». Le problème SAT est le problème de décision consistant à déterminer l'existence d'une valuation, appelée *interprétation*, telle que la réponse « vrai » lorsqu'elle est associée à une formule CNF.

En pratique, l'étude du problème SAT connaît un succès important depuis les années 1960. De nombreux outils algorithmiques très performants ont été développés du fait de sa proximité avec le langage machine et du cadre théorique sur lequel il repose. Il contribue également à améliorer les techniques algorithmiques des problèmes CSP. Très utilisé pour des applications comme la vérification formelle, la planification ou encore la bio-informatique, le problème SAT constitue un domaine majeur de la programmation par contraintes. Le problème SAT est également très utilisé en théorie de la complexité. Dans la hiérarchie des problèmes de décisions, celui-ci est NP-complet. Pour y répondre en temps polynomial dans la taille de la formule, il requiert une machine non-déterministe vérifiant chaque interprétation de

la formule propositionnelle, et tout problème NP-complet peut être réduit au problème SAT en temps polynomial.

Dans ce chapitre, nous décrivons la logique propositionnelle, de manière syntaxique puis sémantique afin de définir le problème SAT. Nous proposons ensuite un aperçu des techniques de recherche et techniques d'inférence pour répondre à ce problème.

3.2 La logique propositionnelle

3.2.1 Syntaxe : le langage propositionnel

Définition générale

La logique propositionnelle est décrite par un langage, appelé *langage de la logique propositionnelle*.

Définition 56 (Langage de la Logique Propositionnelle) *Le langage de la logique propositionnelle, noté LP, est défini par :*

- les valeurs booléennes 0 et 1 ;
- les symboles \top et \perp ;
- un ensemble infini dénombrable $\mathcal{P} = \{p_1, \dots, p_n, \dots\}$ de variables propositionnelles, chacune définie sur l'ensemble des booléens $\mathbb{B} = \{0, 1\}$;
- un ensemble \mathcal{C} de connecteurs logiques binaires entre les variables de \mathcal{P} , tel que pour tout connecteur $\circ \in \mathcal{C}$, \circ est une application de $\mathbb{B} \times \mathbb{B}$ dans \mathbb{B} ;
- les symboles de ponctuations (et).

Une variable propositionnelle décrit une proposition, et les valeurs booléennes 0 et 1 décrivent respectivement *faux* et *vrai*. Ainsi, $p = 1$ si et seulement si la proposition associée à p est *vraie* et $p = 0$ sinon. Dans le cas général, nous supposons que les connecteurs sont binaires, bien que nous puissions en définir d'arité quelconque.

Les éléments de ce langage permettent de définir inductivement des expressions, appelées *formules propositionnelles*.

Définition 57 (Formule propositionnelle) *Soit LP le langage de la logique propositionnelle :*

- 0 et 1 sont des formules de LP ;
- \perp et \top sont des formules de LP ;
- toute variable propositionnelle $p \in \mathcal{P}$ est une formule de LP ;
- si Σ est une formule de LP, alors (Σ) est une formule de LP ;
- si Σ_1 et Σ_2 sont des formules de LP et $\circ \in \mathcal{C}$ un connecteur propositionnel de LP, alors $\circ(\Sigma_1, \Sigma_2)$ est une formule de LP.

Dans le cas général, nous noterons une formule Σ de LP telle que $\Sigma \in \text{LP}$. Nous pouvons associer à chaque formule $\Sigma \in \text{LP}$ un ensemble fini de variables propositionnelles, noté $\text{vars}(\Sigma) = \{p_1, \dots, p_n\}$, tel que $\text{vars}(\Sigma) \subseteq \mathcal{P}$. Puisque chaque connecteur $\circ \in \mathcal{C}$ est une application de \mathbb{B}^2 dans \mathbb{B} et par définition, nous pouvons déduire que toute formule $\Sigma \in \text{LP}$ telle que $|\text{vars}(\Sigma)| = n$ est une application de \mathbb{B}^n dans \mathbb{B} .

Connecteurs propositionnelles

Dans notre cadre, nous restreignons les connecteurs à l'ensemble $\{\neg, \wedge, \vee\} \subseteq \mathcal{C}$.

Définition 58 (Connecteurs propositionnels) *Les connecteurs de $\{\neg, \wedge, \vee\}$ sont décrits par :*

- $\forall \Sigma \in \text{LP}, \forall \alpha \in \mathbb{B}^n, \Sigma(\alpha) = 1$ si et seulement si $\neg \Sigma(\alpha) = 0$;
- $\forall \Sigma_1, \Sigma_2 \in \text{LP}, \forall \alpha_1, \alpha_2 \in \mathbb{B}^n, \Sigma_1(\alpha_1) \vee \Sigma_2(\alpha_2) = 1$ si et seulement si $\Sigma_1(\alpha_1) = 1$ **ou** (non exclusif) $\Sigma_2(\alpha_2) = 1$;
- $\forall \Sigma_1, \Sigma_2 \in \text{LP}, \forall \alpha_1, \alpha_2 \in \mathbb{B}^n, \Sigma_1(\alpha_1) \vee \Sigma_2(\alpha_2) = 1$ si et seulement si $\Sigma_1(\alpha_1) = 1$ **et** $\Sigma_2(\alpha_2) = 1$.

« \neg » est appelé *négation*. C'est un connecteur unaire, et \neg est un connecteur prioritaire dans les formules propositionnelles. Il est idempotent, donc $\forall \Sigma \in \text{LP} : \neg \neg \Sigma = \Sigma$.

« \vee » est appelé *disjonction*. C'est un connecteur binaire, idempotent, associatif et commutatif. Pour $\Sigma_1, \Sigma_2, \Sigma_3 \in \text{LP}, \Sigma_1 \vee \Sigma_1 = \Sigma_1$ (idempotence), $(\Sigma_1 \vee \Sigma_2) \vee \Sigma_3 = \Sigma_1 \vee (\Sigma_2 \vee \Sigma_3) = \Sigma_1 \vee \Sigma_2 \vee \Sigma_3$ (associativité) et $\Sigma_1 \vee \Sigma_2 = \Sigma_2 \vee \Sigma_1$ (commutativité). L'élément absorbant est 1 et l'élément neutre est 0.

« \wedge » est appelé *conjonction*. Il est binaire et il est également associatif, commutatif et idempotent. L'élément absorbant est 0 et l'élément neutre est 1. \vee et \wedge sont distributifs l'un par rapport à l'autre. Ainsi, $(\Sigma_1 \vee \Sigma_2) \wedge \Sigma_3 = (\Sigma_1 \wedge \Sigma_3) \vee (\Sigma_2 \wedge \Sigma_3)$, et $(\Sigma_1 \wedge \Sigma_2) \vee \Sigma_3 = (\Sigma_1 \vee \Sigma_3) \wedge (\Sigma_2 \vee \Sigma_3)$. Enfin, ces connecteurs suivent la loi de dualité de De Morgan : $\neg(\Sigma_1 \vee \Sigma_2) = \neg \Sigma_1 \wedge \neg \Sigma_2$ et $\neg(\Sigma_1 \wedge \Sigma_2) = \neg \Sigma_1 \vee \neg \Sigma_2$.

L'ensemble de connecteurs $\{\neg, \vee, \wedge\}$ permet de définir d'autres connecteurs propositionnels. Posons $\Sigma_1, \Sigma_2 \in \text{LP}$:

- *implication* : $\Sigma_1 \Rightarrow \Sigma_2 = \neg \Sigma_1 \vee \Sigma_2$;
- *équivalence* : $\Sigma_1 \Leftrightarrow \Sigma_2 = (\Sigma_1 \Rightarrow \Sigma_2) \wedge (\Sigma_2 \Rightarrow \Sigma_1)$;
- *Mutuelle exclusivité* : $\Sigma_1 \oplus \Sigma_2 = (\Sigma_1 \vee \Sigma_2) \wedge (\neg \Sigma_1 \vee \neg \Sigma_2)$.

Notons que la loi de dualité de De Morgan nous permet également de réduire l'ensemble $\{\neg, \vee, \wedge\}$ à $\{\neg, \vee\}$ ou $\{\neg, \wedge\}$. Ces ensembles de connecteurs sont dits *complets* car tout connecteur de \mathcal{C} peut se définir par cet ensemble.

Formes normales

Parmi les formules propositionnelles de LP, nous pouvons distinguer certaines formules particulières.

Une formule $\Sigma = p$ est appelé *littéral positif* ou *occurrence positive* de p , et $\Sigma = \neg p$ est appelée un *littéral négatif* ou *occurrence négative* de p . Une disjonction de littéraux est une *clause* et une conjonction de littéraux est un *cube*. Une clause et un cube sont dits *unitaires* si et seulement si ils ne sont composés que d'un seul et unique littéral. À ces différentes formes s'ajoutent des formes dites *normales*.

Définition 59 (Formes normales)

- une formule propositionnelle en forme normale conjonctive, notée CNF, est une conjonction de clauses.
- Une formule propositionnelle en forme normale disjonctive, notée DNF, est une disjonction de cubes.
- Une formule propositionnelle en forme normale négative, notée NNF, est une formule propositionnelle pour laquelle la négation n'apparaît que sur les littéraux négatifs.

Cette définition permet de nous approcher de la définition du problème SAT. Dans la suite, nous noterons par *formule CNF* (resp. *formule DNF*) toute formule sous forme normale *conjonctive* (resp. *disjonctive*).

Notons qu'il existe une représentation ensembliste des formules CNF. Chaque élément est une clause de la formule et l'ensemble représente la conjonction des clauses. Une formule CNF $\Sigma = clause_1 \wedge \dots \wedge clause_e$ peut ainsi être décrite par $\Sigma = \{clause_1, clause_2, \dots, clause_e\}$. Si chaque clause est une clause unitaire, alors la formule est un cube. Un cube est donc une formule CNF. Dans la suite, e dénotera le nombre de clauses d'une formule CNF.

3.2.2 Sémantique : la satisfiabilité

Pour définir le problème de la satisfiabilité, il est nécessaire de définir le sens d'une formule propositionnelle. Comme énoncé précédemment, une formule propositionnelle $\Sigma \in \text{LP}$ est une application de \mathbb{B}^n dans \mathbb{B} , avec $|\text{vars}(\Sigma)| = n$.

Définition 60 (Interprétation) Soit $P \subseteq \mathcal{P}$. Une interprétation \mathcal{I}_P est une application qui, à tout $p \in P$, associe une valeur de \mathbb{B} .

Une interprétation dans le cadre propositionnel peut être rapprochée d'une *instanciation* du cadre des réseaux de contraintes. Une *interprétation complète* pour une formule Σ est une interprétation $\mathcal{I}_{\text{vars}(\Sigma)}$ sur les variables $\text{vars}(\Sigma)$ de la formule Σ . Nous définissons un opérateur $|$ permettant, à une formule $\Sigma \in \text{LP}$, d'appliquer une interprétation \mathcal{I}_P afin de produire une formule $\Sigma' \in \text{LP}$, telle que $\Sigma' = \Sigma|_{\mathcal{I}_P}$. Nous utilisons ce raccourci d'écriture pour dénoter que toute variable $p \in P$ est remplacée par sa valeur $\mathcal{I}_P(p)$ dans l'application Σ . Ainsi, si $\Sigma|_{p=0}$ (resp. $\Sigma|_{p=1}$) signifie que toute variable propositionnelle p est remplacée par la valeur 0 (resp. la valeur 1). Les valeurs peuvent être *propagées* en utilisant les définitions et les propriétés des connecteurs logiques décrits précédemment : $\neg 0 = 1$, $\neg 1 = 0$, 1 est absorbant pour la disjonction et neutre pour la conjonction, 0 est absorbant pour la conjonction et neutre pour la disjonction. Remarquons que si une interprétation complète \mathcal{I} est appliquée à une formule $\Sigma \in \text{LP}$, alors le résultat est 0 ou 1.

Définition 61 (Satisfiabilité d'une formule propositionnelle) Soit $\Sigma \in \text{LP}$:

- Σ est satisfiable, si et seulement si il existe une interprétation complète \mathcal{I} telle que $\Sigma|_{\mathcal{I}} = 1$;
- Σ est insatisfiable si et seulement si pour toute interprétation complète \mathcal{I} : $\Sigma|_{\mathcal{I}} = 0$;
- Σ est une tautologie, notée \top , si et seulement si pour toute interprétation complète \mathcal{I} : $\Sigma|_{\mathcal{I}} = 1$;
- Σ est une contradiction, notée \perp , si et seulement si Σ est insatisfiable.

Définition 62 (Modèle) Soit $\Sigma \in \text{LP}$ une formule propositionnelle. Un modèle de Σ est une instanciation complète \mathcal{I} telle que $\Sigma|_{\mathcal{I}} = 1$.

Le problème SAT est le problème de la satisfiabilité d'une formule CNF. En d'autres termes, il est décrit par la question « Est-ce que la formule CNF est satisfiable ? ». Remarquons, au même titre que le problème CSP, que cela n'implique pas de trouver un modèle de la formule mais qu'il faut en prouver l'existence. Puisqu'une formule CNF est décrite par un ensemble fini de variables propositionnelles, une formule propositionnelle admet un nombre fini d'interprétations. L'ensemble de ces interprétations décrit l'espace de recherche d'une formule propositionnelle. Dans cet espace de recherche, nous pouvons caractériser un sous-ensemble qui constitue l'ensemble des modèles.

Définition 63 (Ensemble des modèles) Soit Σ une formule propositionnelle. $\text{mod}(\Sigma)$ est l'ensemble des modèles de Σ sur l'ensemble des variables propositionnelles \mathcal{P} de LP.

Remarquons que l'ensemble des modèles d'une formule $\Sigma \in \text{LP}$ est défini sur toutes les variables propositionnelles de LP, et pas uniquement sur les variables de la formule $\text{vars}(\Sigma)$. Ceci permet d'exprimer que tout prolongement d'un modèle \mathcal{I} de Σ sur l'ensemble des variables propositionnelles \mathcal{P} est également un modèle de Σ . Ainsi, il est possible de comparer les modèles des formules qui ne sont pas définis sur un même ensemble de variables propositionnelles.

Par définition, une formule $\Sigma \in \text{LP}$ est une tautologie si et seulement si toute instanciation complète de Σ est un modèle. En d'autres termes, l'ensemble des modèles $\text{mod}(\Sigma)$ est le même pour toute tautologie. Par exemple, pour tout $p, p' \in \mathcal{P}$, la formule $p \vee \neg p$ est une tautologie, et $(p \wedge \neg p) \vee p'$ également.

Ces deux formules CNF ont le même ensemble de modèles mais ne sont pourtant pas définis sur les mêmes variables propositionnelles. De façon analogue, toute formule ayant pour ensemble de modèles l'ensemble vide est une contradiction.

La comparaison de l'ensemble des modèles nous permet d'introduire la notion de *conséquence sémantique* et d'*équivalence sémantique*.

Définition 64 (Conséquence sémantique) Soient $\Sigma_1, \Sigma_2 \in \text{LP}$. Σ_2 est une conséquence sémantique de Σ_1 , noté $\Sigma_1 \models \Sigma_2$, si et seulement si $\text{mod}(\Sigma_1) \subseteq \text{mod}(\Sigma_2)$.

Définition 65 (Équivalence sémantique) Soient $\Sigma_1, \Sigma_2 \in \text{LP}$. Σ_1 et Σ_2 sont équivalents sémantiquement, noté $\Sigma_1 \equiv \Sigma_2$, si et seulement si $\text{mod}(\Sigma_1) = \text{mod}(\Sigma_2)$.

\top est donc sémantiquement équivalent à toute formule tautologique de LP et \perp est sémantiquement équivalent à toute formule contradictoire. Lorsqu'une formule $\Sigma \in \text{LP}$ est telle que $\top \models \Sigma$, également noté $\models \Sigma$, on dit également que Σ est *valide*.

Propriété 7 Soient $\Sigma_1, \Sigma_2 \in \text{LP}$. $\Sigma_1 \models \Sigma_2$ si et seulement si $\Sigma_1 \Rightarrow \Sigma_2$.

3.3 Techniques algorithmiques

3.3.1 Algorithmes de recherche et propagation

Pour décider de la satisfiabilité d'une formule propositionnelle sous forme CNF, il est possible d'exploiter les techniques de résolutions décrites dans le cadre des réseaux de contraintes 1. L'approche repose sur un parcours l'ensemble des interprétations complètes d'une formule CNF à l'aide d'un arbre de recherche. Si une interprétation complète est une solution, alors la formule est satisfiable. Cependant, comme nous l'avons mentionné précédemment, le cadre d'un algorithme de type génération d'interprétations complètes et vérification n'est pas le plus pertinent. Il faut maintenir la cohérence locale de la formule propositionnelle après chaque affectation. Maintenir la cohérence locale dans le cadre propositionnelle signifie propager la valeur affectée afin de vérifier si celle-ci n'est pas trivialement incohérente. Pour cela, nous utilisons la *propagation unitaire*.

Définition 66 (Propagation unitaire) La propagation unitaire est une application qui, à toute formule $\Sigma \in \text{LP}$, associe une formule $\Sigma^* \in \text{LP}$ telle que :

- si il n'existe pas de clauses unitaires dans Σ , alors $\Sigma^* = \Sigma$;
- si, quelle que soit $\Sigma' \in \text{LP}$, Σ est de la forme $\Sigma = \Sigma' \wedge p \wedge \neg p$, alors $\Sigma^* = \perp$;
- si, quelle que soit $\Sigma' \in \text{LP}$, Σ est de la forme $\Sigma = \Sigma' \wedge \neg p$ alors $\Sigma^* = \Sigma_{|p=0}^*$;
- si, quelle que soit $\Sigma' \in \text{LP}$, Σ est de la forme $\Sigma = \Sigma' \wedge p$ alors $\Sigma^* = \Sigma_{|p=1}^*$.

La propagation unitaire est un procédé récursif qui se termine lorsque la formule atteint un *point fixe*, c'est-à-dire lorsque la formule est prouvée incohérente ou lorsqu'il n'y a plus de clauses unitaires dans la formule propositionnelle. En considérant $\Sigma \in \text{LP}$ une formule CNF et $p \in \text{vars}(\Sigma)$, appliquer la propagation unitaire sur la formule $\Sigma \wedge p$ (resp. $\Sigma \wedge \neg p$) produit la formule $\Sigma_{|p=1}$ (resp. $\Sigma_{|p=0}$) et la propagation se poursuit jusqu'à atteindre un point fixe. Les affectations peuvent donc être effectuées par la conjonction de clauses unitaires à la formule et la propagation de ceux-ci. Dans la suite, quelle que soit \mathcal{I} une interprétation de Σ , la notation $p \in \mathcal{I}$ (resp. $\neg p \in \mathcal{I}$) sera équivalente à $(p, 1) \in \mathcal{I}$ (resp. $(p, 0) \in \mathcal{I}$).

Propriété 8 Soit $\Sigma \in \text{LP}$. Σ est satisfiable si et seulement si Σ^* est satisfiable.

On utilise parfois une forme de conséquence sémantique affaiblie, notée $\Sigma \models_* \Sigma'$, afin de préciser qu'une formule $\Sigma' \in LP$ est la conséquence sémantique de $\Sigma \in LP$ par propagation unitaire. Par exemple, $\Sigma \not\models \perp$ signifie $mod(\Sigma) \not\subseteq \emptyset$, que l'on peut traduire par Σ admet un modèle. Dans le cas de l'affaiblissement, $\Sigma \not\models_* \perp$ signifie que $\Sigma \in LP$ n'est pas incohérente pas propagation unitaire sur Σ . Dans la suite, nous appellerons $UP(\cdot)$ la procédure permettant d'appliquer la propagation unitaire sur une formule CNF, telle que $UP(\Sigma)$ est équivalent à $\Sigma \leftarrow \Sigma^*$ pour tout $\Sigma \in LP$.

Pour répondre au problème de la satisfiabilité, nous pouvons utiliser une approche par recherche locale [SELMAN *et al.* 1992] ou une approche par recherche exhaustive [DAVIS *et al.* 1962]. La procédure Davis-Putnam-Logemann-Loveland (DPLL) (Algorithme 11) est un algorithme correct et complet pour décider de la satisfiabilité d'une formule CNF. Tout d'abord, la propagation unitaire est appliquée sur la formule propositionnelle afin d'éliminer toutes clauses unitaires. Si la formule résultante est la contradiction \perp alors la formule est incohérente. Si la formule résultante est tautologique, alors la formule est satisfiable. Dans le cas contraire, une variable propositionnelle p est choisie. La formule Σ est alors satisfiable si et seulement si l'une des affectations $p = 0$ ou $p = 1$ est globalement cohérente dans Σ . Nous supposons donc que la branche droite n'est évaluée que si la branche gauche est incohérente.

Algorithme 11: Procédure DPLL

Entrées : Σ : une formule CNF
Sorties : Oui si Σ est satisfiable, Non sinon

```

1  début
2  |   UP( $\Sigma$ )
3  |   si  $\Sigma = \perp$  alors
4  |   |   retourner Non
5  |   si  $\Sigma = \top$  alors
6  |   |   retourner Oui
7  |    $p \leftarrow$  Choisir une variable propositionnelle de  $vars(\Sigma)$ 
8  |   retourner DPLL( $\Sigma \wedge p$ ) ou DPLL( $\Sigma \wedge \neg p$ )

```

L'efficacité de cet algorithme dépend, comme énoncé dans le cadre des réseaux de contraintes, du choix heuristique sur la variable propositionnelle à affecter, et du choix de l'occurrence (positive ou négative) pour le branchement. Outre ces choix heuristiques, de nombreuses améliorations peuvent être apportées afin de réduire l'arbre de recherche évalué. Dans la suite, nous présentons succinctement quelques approches permettant de simplifier la formule propositionnelle, et nous décrivons l'un des algorithmes modernes les plus utilisé pour répondre au problème SAT.

3.3.2 Simplification de formules propositionnelles

Pour illustrer la simplification de formules propositionnelles, nous pouvons tout d'abord mentionner la propagation de littéraux purs. Un littéral est dit *pur* dans une formule propositionnelle si et seulement si sa négation n'apparaît pas dans la formule. En d'autres termes, si le littéral p , resp. $\neg p$, est le seul apparaissant dans Σ , alors $\Sigma \wedge p \models \Sigma$, resp. $\Sigma \wedge \neg p \models \Sigma$. Par conséquent, si $UP(\Sigma \wedge p)$ (resp. $UP(\Sigma \wedge \neg p)$) est satisfiable alors Σ est satisfiable. L'intérêt de cette approche réside essentiellement dans la réduction du nombre de clauses de la formule, pouvant engendrer l'apparition de nouveaux littéraux purs afin de faciliter la preuve de la satisfiabilité.

Une autre forme de simplification sur les clauses est la *subsumption*.

Définition 67 (Subsumption) Soient $c_1, c_2 \in LP$ deux clauses. c_1 subsume c_2 si et seulement si il existe une clause $c' \in LP$ telle que $c_2 = c_1 \vee c'$.

La détection de clauses subsumées réduit la formule propositionnelle en éliminant certaines clauses. Cette règle s'appuie sur la loi d'absorption, c'est-à-dire $c_1 \wedge (c_1 \vee c') = c_1$. Si $\Sigma' \in \text{LP}$ est la simplification de $\Sigma \in \text{LP}$ par la suppression des clauses subsumées, alors $\Sigma' \models \Sigma$. Notons que la propagation unitaire peut être assimilée à une forme de subsumption faible puisque toute clause subsumée par une clause unitaire sera supprimée, ce qui est équivalent à propager la clause unitaire.

Certaines méthodes de simplifications s'appuient sur des règles d'inférence logique, telle que la résolution [ROBINSON 1965].

Définition 68 (Règle de résolution) Soient $c = p \vee c_1$ et $c' = \neg p \vee c_2$ deux clauses de LP. La résolvente de c et c' sur p , notée $\text{resolution}(p, c, c')$, est la clause $c_1 \vee c_2 \in \text{LP}$.

La règle de résolution permet donc l'ajout de nouvelles clauses, jusqu'à la stabilité de la formule, c'est-à-dire jusqu'à ce que toutes nouvelles clauses obtenues par résolution soient subsumées par une clause de la formule. En limitant la taille des résolvantes, en exploitant certaines structures de la formule propositionnelle ou en simplifiant les résolvantes, il est ainsi possible de déterminer certaines incohérences locales et donc de réduire l'arbre de recherche en augmentant le nombre de clauses.

3.3.3 Outils algorithmiques modernes

De la procédure DPLL (Algorithme 11) dérive la plupart des algorithmes de recherche exhaustive actuels, sur lesquels ce sont greffées de nombreuses améliorations. Parmi ces améliorations, nous retrouvons les heuristiques de choix de variables dirigées par les conflits afin d'adopter une stratégie d'*échec d'abord*, les redémarrages afin d'éviter les phénomènes *heavy tail* et donner plus de poids à l'heuristique de choix de variables, l'analyse de conflits permettant d'effectuer des sauts dans l'arbre de recherche, et l'apprentissage par l'utilisation d'une base de *nogoods*.

L'algorithme 12 présente un algorithme de recherche dirigée par les conflits et l'apprentissage de clauses (CDCL pour *Conflict Driven Clause Learning*). Celui-ci est basé sur l'algorithme DPLL qui, à partir d'une formule Σ , développe un arbre de recherche, ce qui en fait un algorithme correct et complet. La première étape consiste à appliquer la propagation unitaire et effectuer une analyse de conflits le cas échéant. Si il existe une clause vide dans Σ^* , alors Σ est incohérente. Si aucune affectation n'a été effectuée, alors la formule est incohérente. Dans le cas contraire, la clause de Σ produisant cette incohérence et l'instanciation courante sont analysées, afin de déterminer la raison γ de l'incohérence. Cela permet de déduire le niveau dans l'arbre de recherche à partir duquel il est nécessaire d'effectuer de nouveaux choix, donc d'effectuer un *saut-arrière*. Mais γ est également une information permettant d'éviter d'obtenir de nouveau l'incohérence détectée, appelée *nogood*. Le *nogood* est donc ajouté à la formule propositionnelle à évaluer. Si un redémarrage doit être effectué, le saut se fait à la racine de l'arbre, cela permet d'éviter de traiter un sous-problème incohérent dans les profondeurs de l'arbre de recherche. Dans le cas contraire, il y a un saut-arrière au niveau spécifié.

Si l'instanciation est complète, le problème est satisfiable. Dans le cas contraire, un littéral l est choisi suivant l'approche VSIDS (*Variable State Independent Decaying Sum*), décrite par :

- chaque littéral possède un poids, initialisé à 0 ;
- lorsqu'un littéral est impliqué dans le processus d'analyse du conflit, ce poids est incrémenté ;
- le littéral avec le poids le plus fort est choisi, ce qui permet de choisir les littéraux potentiellement impliqués dans un sous-problème incohérent ;
- les poids sont divisés périodiquement par une constante, afin d'éviter de ne traiter qu'un sous-problème fortement pondéré récemment.

On cherche alors la cohérence de la formule $\Sigma \wedge l$. Et le procédé se poursuit par l'application de la propagation unitaire.

Algorithme 12: Algorithme CDCL

Entrées : Σ : une formule CNF
Sortie : Oui si Σ est satisfiable, Non sinon

```

1  début
2  |  $\mathcal{I} \leftarrow \emptyset$ 
3  | Simplifier( $\Sigma$ )
4  | tant que (Vrai) faire
5  | |  $c \leftarrow \text{UP}(\Sigma)$ 
6  | | si ( $c$  existe) alors
7  | | | si ( $\mathcal{I}$  est vide) alors retourner Non
8  | | |  $\gamma \leftarrow \text{AnalyseConflit}(\Sigma, \mathcal{I}, c)$ 
9  | | | saut  $\leftarrow \text{CalculSaut}(\gamma, \mathcal{I})$ 
10 | | |  $\Sigma \leftarrow \Sigma \wedge \gamma$ 
11 | | | si (Redémarrage()) alors saut  $\leftarrow 0$ 
12 | | | RetourArrière( $\Sigma, \mathcal{I}, \text{saut}$ )
13 | | sinon
14 | | | si ( $\mathcal{I}$  est complète) alors retourner Oui
15 | | |  $l \leftarrow \text{Choisir un littéral de } \Sigma$ 
16 | | |  $\mathcal{I} \leftarrow \mathcal{I} \cup \{l\}$ 
17 | | |  $\Sigma \leftarrow \Sigma \wedge l$ 

```

Cet algorithme est l'un des algorithmes de recherches complets permettant d'obtenir les meilleures performances en temps afin de répondre au problème SAT.

3.3.4 Classes traitables

Dans le cas général, décider de la satisfiabilité d'une formule CNF est NP-complet. Intuitivement, décider de la satisfiabilité d'une formule propositionnelle est un problème difficile. Cependant, il est possible d'identifier certaines classes de formules, appelées *classes traitables*, afin de simplifier la tâche des algorithmes de recherche. Une classe traitable est une classe de problèmes qu'il est possible de reconnaître en temps polynomial et pour lequel il existe un algorithme polynomial en temps dans la taille de la formule permettant de décider de sa satisfiabilité. Parmi ces classes, nous proposons un aperçu des plus célèbres [PAPADIMITRIOU 1994].

Formule de Horn Une *formule de Horn* est une formule CNF dans laquelle chaque clause possède au plus un littéral positif. De telles clauses sont appelées *clauses de Horn*. Le problème de la satisfiabilité d'une formule de Horn est appelé HORNSAT. Il est P-complet, il est donc possible de décider de la satisfiabilité de la formule avec une machine déterministe en temps polynomial.

Formule 2CNF Une *formule 2CNF* est une formule CNF dans laquelle chaque clause est composée d'au plus 2 littéraux. Le problème permettant de décider de la satisfiabilité d'une formule 2CNF est appelé 2SAT, et il est NL-complet. Il est possible de décider de la satisfiabilité de cette formule avec une machine non-déterministe en utilisant un espace au plus logarithmique. La classe NL est incluse dans P, il existe donc un algorithme en temps polynomial permettant de décider si une formule 2CNF est satisfiable.

Formule Horn-renommable Une *formule Horn-renommable* est une formule CNF qui peut être transformée en formule de Horn par renommage des variables propositionnelles. Déterminer si une formule

CNF est Horn-renommable peut être effectué en temps polynomial, car ce problème se ramène à décider de la satisfiabilité d'une formule 2SAT. Renommer les variables propositionnelles pour obtenir une formule de Horn peut être effectué en temps linéaire dans le nombre de variables. Pour cela, il suffit de remplacer les littéraux d'une formule par leur négation. Par conséquent, il existe un algorithme polynomial permettant de vérifier si la formule peut être renommée en formule de Horn, et le cas échéant de la renommer et de répondre au problème de satisfiabilité. Les formules Horn-renommable peuvent être vues comme une généralisation des formules de Horn, et toute formule de Horn est Horn-renommable. Le problème de la satisfiabilité d'une formule Horn-renommable est P-complet.

Exploiter ces classes particulières durant la recherche passe par une analyse structurale de la formule propositionnelle. Certaines structures, appelées *backdoor* [WILLIAMS *et al.* 2003], caractérisent cette approche.

Définition 69 (Backdoor) Soit $\Sigma \in \text{LP}$ et $P \subseteq \text{vars}(\Sigma)$ un ensemble non vide de variables propositionnelles. P est une *backdoor* si il existe une instantiation \mathcal{I}_P sur Σ telle que $\Sigma|_{\mathcal{I}_P}$ puisse être résolue en temps polynomial.

Définition 70 (Strong Backdoor) Soit $\Sigma \in \text{LP}$ et $P \subseteq \text{vars}(\Sigma)$ un ensemble non vide de variables propositionnelles. P est une *strong backdoor* si, pour toute instantiation \mathcal{I}_P sur Σ , $\Sigma|_{\mathcal{I}_P}$ peut être résolue en temps polynomial.

Appliquer une instantiation des variables d'un backdoor à une formule propositionnelle permet alors d'obtenir une formule assimilée à une classe polynomiale. Extraire un *strong backdoor* de taille $n' \leq n$, où n est le nombre de variables propositionnelles d'une formule CNF, permet de développer un arbre de recherche contenant $2^{n'}$ instantiations de taille n' . Après application d'une instantiation, il suffit alors de résoudre en temps polynomial la sous-formule obtenue. L'approche apparaît intéressante en pratique si la taille du *strong backdoor* est petite par rapport au nombre de variables, ou si la sous-formule traitable est grande. Or, identifier la plus grande sous-formule associée à une classe polynomiale n'est pas toujours facile. Par exemple, décider s'il existe une sous-formule de taille au moins k qui est Horn-renommable est un problème NP-complet dans le cas général. Il est néanmoins possible de faire une approximation du *strong backdoor*, notamment par le développement d'un arbre de recherche sur des sous-ensembles de variables. Ceux-ci peuvent être de taille prédéfinie, ou encore choisis aléatoirement.

3.4 Conclusion

La logique propositionnelle est un cadre théorique sur lequel repose de nombreuses études. Exploiter le langage de la logique propositionnelle, et en particulier les formules CNF, procure un avantage certain pour la modélisation de connaissances. Dans tout formalisme, de nombreuses approches algorithmiques modernes dérivent d'étude du problème SAT, comme les algorithmes de recherche complets reposant sur le principe de la procédure DPLL. Le cadre booléen apporte une proximité avec le langage machine, ce qui facilite le développement d'outils algorithmiques pour répondre au problème de la satisfiabilité. L'analyse de conflits et l'apprentissage sont des techniques algorithmiques fortement exploitées dans le cadre propositionnel, qu'il est plus difficile à mettre en place dans un cadre non booléen. La définition de classes traitables est également très utilisée, pour répondre au problème SAT lui-même, mais également pour caractériser des classes traitables dans d'autres formalismes. Cette notion est liée à la notion de transformation, permettant de passer d'une représentation des connaissances à une autre.

Définir et exploiter les transformations

Chapitre 4

Du problème CSP au problème SAT

Sommaire

4.1	Introduction	75
4.2	De l’instanciation à l’interprétation	76
4.3	Modélisation des contraintes : les conflits	77
4.3.1	Décrire les tuples conflits	77
4.3.2	Vers de nouvelles approches exploitant les transformations	78
4.4	Modélisation des contraintes : les supports	82
4.4.1	Supports et Cohérence d’Arcs	82
4.4.2	Vers de nouvelles modélisations de cohérence locale	83
4.5	Conclusion	84

4.1 Introduction

Exploiter les transformations entre différentes représentations d’un même problème est un aspect important en programmation par contraintes. En l’occurrence, transformer les RCD en formule CNF permet d’exploiter les outils théoriques, fournis par la logique propositionnelle, et algorithmiques, en s’approchant du langage machine. De nombreux travaux ont mis en évidence de telles transformations [KASIF 1990, WALSH 2000, GENT 2002, ROUSSEL 2005]. L’objectif commun est de transformer un RCD en une formule CNF afin de décider de la cohérence du RCD en décidant de la cohérence de la formule CNF associée. En théorie, les problèmes CSP et SAT sont NP-complets. Par conséquent, il existe une *réduction fonctionnelle polynomiale* du problème CSP au problème SAT et *vice versa*, c’est-à-dire un processus polynomial en temps permettant de répondre au problème CSP en exploitant le problème SAT et *vice versa*.

Puisque vérifier la cohérence d’un RCD passe par la construction d’une instanciation complète et la vérification de sa cohérence, plusieurs approches ont été définies dans ce sens. Pour construire l’instanciation, le cadre adopté est d’associer une proposition à chaque couple variable/valeur. Ainsi, le lien est fait entre instanciation du RCD et interprétation du cadre propositionnelle. Pour obtenir la correspondance entre la cohérence d’une interprétation et la cohérence du RCD, il est nécessaire de modéliser les contraintes. Pour cela, la transformation directe (ou *direct encoding*) [WALSH 2000] modélise les tuples conflits sous la forme de formules CNF, afin de décrire ce qui est en conflit avec les résultat attendu. La transformation support (ou *support encoding*) [KASIF 1990] permet de faire correspondre la cohérence d’arcs des RCD avec la propagation unitaire du cadre propositionnel en transformant les tuples supports.

Puisque la cohérence d'arcs permet de vérifier la cohérence d'une instantiation, la propagation unitaire permet de vérifier la cohérence d'une interprétation.

Bien que ces transformations des contraintes soient naturelles, la transformation des contraintes supports en formule CNF ne décrit pas totalement la contrainte. Elle est une approximation, dans le but d'obtenir une transformation rapide, et suffisante pour décider la cohérence de la formule CNF. Cette approche masque un certain potentiel des transformations vers le langage propositionnel et des techniques d'inférences de ce cadre. Elle est réalisée naturellement et implicitement à l'aide de processus de simplification de la logique propositionnelle, tels que la règle de résolution ou la subsumption. En exploitant la même approche, il est possible d'entrevoir d'autres transformations pouvant exprimer une autre vision de la résolution du problème CSP par l'exploitation du problème SAT.

Dans ce chapitre, nous étudions les transformations *directes* et *supports* des RCD vers les formules CNF, qui nous permettront de définir des transformations des RCQ vers les formules CNF. Dans un premier temps, nous décrivons la transformation d'instanciation complète vers les formules CNF en définissant l'ensemble des variables propositionnelles utilisées. Nous étudions ensuite la transformation directe et montrons que cette étude peut amener à d'autres raisonnements pour répondre au problème CSP. Nous proposons ensuite un aperçu de la transformation support en montrant que, dans le cas de RCD d'arité quelconque, il pourrait être intéressant de considérer une approche plus coûteuse pour la transformation, qui améliorerait néanmoins le pouvoir d'inférence de la propagation unitaire.

4.2 De l'instanciation à l'interprétation

L'idée générale de cette transformation des RCD aux formules propositionnelles est d'obtenir une correspondance naturelle entre *instanciation complète* et *interprétation complète*. Considérons $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD et Σ la formule propositionnelle associée à \mathcal{P} . La première étape consiste à caractériser les variables propositionnelles. Puisqu'une instantiation complète est un ensemble d'affectations sur toutes les variables d'un RCD, il faut pouvoir modéliser chaque affectation de cette instantiation. Posons \mathcal{I} une instantiation complète de \mathcal{P} . Nous définissons donc la variable propositionnelle $[x = v] \in \text{vars}(\Sigma)$ afin de décrire la proposition « $(x, v) \in \mathcal{I}$ ». Ainsi, $[x = v] = 1$ si et seulement si $(x, v) \in \mathcal{I}$, et $[x = v] = 0$ si et seulement si $(x, v) \notin \mathcal{I}$. Par conséquent, l'ensemble des variables propositionnelles de Σ est composé de nd éléments, avec n le nombre de variables de \mathcal{P} et d le nombre de valeurs par domaine, décrit par :

$$\text{vars}(\Sigma) = \{[x = v] : x \in \mathcal{X} \wedge v \in D_x\}$$

Cependant, une instantiation complète \mathcal{I} est une application sur l'ensemble des variables du RCD. Par conséquent, il est nécessaire de poser des contraintes sur les variables propositionnelles afin qu'une et une seule valeur ne soit associée à une variable :

- $\forall x \in \mathcal{X}$, il existe *au moins une* valeur $v \in D_x$ telle que $(x, v) \in \mathcal{I}$:
- $\forall x \in \mathcal{X}$, il existe *au plus une* valeur $v \in D_x$ telle que $(x, v) \in \mathcal{I}$:

Ces deux règles contraintes sont décrites par des formules propositionnelles, appelées respectivement *At-Least-One* pour « au moins une », notée *ALO*, et *At-Most-One* pour « au plus une », notée *AMO*. Nous noterons dans la suite $\Sigma_{\mathcal{P}}^{ALO}$ et $\Sigma_{\mathcal{P}}^{AMO}$ les formules *ALO* et *AMO* associées à \mathcal{P} .

Définition 71 (Formule ALO) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD. La formule « Au moins une » associée à \mathcal{P} est :

$$\Sigma_{\mathcal{P}}^{ALO} = \bigwedge_{(x \in \mathcal{X})} \bigvee_{(v \in D_x)} [x = v]$$

Définition 72 (Formule AMO) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD. La formule « Au plus une » associée à \mathcal{P} est :

$$\Sigma_{\mathcal{P}}^{AMO} = \bigwedge_{(x \in \mathcal{X})} \bigvee_{(v, v' \in D_x : v \neq v')} \neg[x = v] \vee \neg[x = v']$$

Il est facile de voir que tout modèle de la formule $\Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{AMO}$ peut donc être associée à une instantiation complète par construction.

Définition 73 (Du modèle à l'instanciation complète) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, et μ un modèle de $\Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{AMO}$. \mathcal{I}_{μ} est l'instanciation complète de \mathcal{P} associée à μ telle que, pour toute variable $x \in \mathcal{X}$, $(x, v) \in \mathcal{I}_{\mu}$ si et seulement si $[x = v] \in \mu$.

La complexité algorithmique en temps (resp. en espace) dans le pire des cas est $\mathcal{O}(nd)$ (resp. $\mathcal{O}(d)$) pour la construction de la formule *ALO* et $\mathcal{O}(nd^2)$ (resp. $\mathcal{O}(1)$) pour la construction de la formule *AMO*. En ce qui concerne les dimensions des formules obtenues, la formule *ALO* contient n clauses de taille d , et la formule *AMO* est composée de $n \frac{d(d-1)}{2}$ clauses de taille 2.

Il est possible d'énumérer toute instantiation complète d'un RCD en énumérant les modèles de $\Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{AMO}$. En évaluant cette formule par un algorithme de recherche avec maintien de la propagation unitaire, il est facile de voir qu'à tout instant de la recherche, une interprétation localement cohérente peut être associée à une instantiation partielle. Cependant, il faut pouvoir vérifier la cohérence locale de cette instantiation et vérifier si une instantiation complète est une solution. Pour cela, deux approches ont été proposées. L'une se concentre sur la représentation des contraintes sous la forme de tuples conflits [WALSH 2000], en énonçant qu'une contrainte est satisfaite si l'instanciation n'est associée à aucun tuple conflit. L'autre se base sur la représentation des tuples supports [GENT 2002], en énonçant qu'une contrainte est satisfaite par une instantiation si et seulement si satisfait elle est associée à un des supports.

4.3 Modélisation des contraintes : les conflits

4.3.1 Décrire les tuples conflits

Considérons dans un premier temps le cas de la transformation des contraintes sous la forme de conflits [WALSH 2000], appelée *encodage direct*, ou *direct encoding* en anglais. Par définition, la contrainte c est satisfaite par \mathcal{I} si et seulement si il n'existe pas de tuple conflit $\tau \in \bar{c}$, tel que $(x = \tau[x]) \in \mathcal{I}$ pour tout $x \in scp(c)$. Supposons, dans un premier temps, que la contrainte \bar{c} ne soit composée que d'un seul tuple conflit (v_1, v_2, \dots, v_r) entre les variables x_1, x_2, \dots, x_r ⁴. c est satisfaite par \mathcal{I} si et seulement si $\{(x_1, v_1), \dots, (x_r, v_r)\} \notin \mathcal{I}$. Par conséquent, c est satisfaite par \mathcal{I} si et seulement si $(x_1, v_1) \notin \mathcal{I}$ ou \dots ou $(x_r, v_r) \notin \mathcal{I}$. Puisque toute proposition $(x_i, v_i) \notin \mathcal{I}$ est décrite par $\neg[x_i = v_i]$, pour tout $i \in [1 ; r]$, nous pouvons affirmer que c est satisfaite par \mathcal{I} si et seulement si $\neg[x_1 = v_1] \vee \dots \vee \neg[x_r = v_r]$ admet un modèle. Si maintenant nous souhaitons vérifier la cohérence d'une instantiation sur l'ensemble des contraintes d'un RCD, chacune composée d'un ensemble de tuples conflits, nous pouvons généraliser le raisonnement précédent et déduire que \mathcal{I} est cohérente si et seulement si il existe un modèle de la formule suivante :

4. Nous rappelons que c et \bar{c} sont deux contraintes équivalentes, représentant respectivement les tuples supports et les tuples conflits.

$$\Sigma_{\mathcal{P}}^{Conflit} = \bigwedge_{(c \in \mathcal{C})} \left(\neg \bigvee_{(\tau \in \bar{c})} \bigwedge_{(x \in scp(c))} [x = \tau[x]] \right)$$

Cette formule peut donc facilement être développée en une formule CNF.

Définition 74 (Formule Conflit) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD. La formule conflit associée à \mathcal{P} est :

$$\Sigma_{\mathcal{P}}^{Conflit} = \bigwedge_{(c \in \mathcal{C})} \left(\bigwedge_{(\tau \in \bar{c})} \bigvee_{(x \in scp(c))} \neg [x = \tau[x]] \right)$$

La complexité algorithmique en temps (resp. en espace) dans le pire des cas pour décrire la formule conflit associé à un RCD est $\mathcal{O}(erd^r)$ (resp. $\mathcal{O}(r)$). La formule $\Sigma_{\mathcal{P}}^{Conflit}$ contient alors ed^r clauses de taille r dans le pire des cas.

Définition 75 (Encodage direct) Soit \mathcal{P} un réseau de contraintes. La formule propositionnelle associée à \mathcal{P} par l'encodage direct est :

$$\Sigma_{\mathcal{P}}^{direct} = \Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{Conflit}$$

Propriété 9 Soient \mathcal{P} un RCD et $\Sigma_{\mathcal{P}}^{direct}$ la formule propositionnelle associée à \mathcal{P} par encodage direct. \mathcal{P} est cohérent si et seulement si $\Sigma_{\mathcal{P}}^{direct}$ est satisfiable.

La preuve est immédiate par construction. Les clauses *AMO* et *ALO* garantissent la construction d'une instantiation et la formule conflit assure la cohérence de celle-ci. Nous avons donc décrit un procédé pour transformer un RCD en formule CNF, en temps polynomial dans la taille du RCD. L'exemple 12 illustre la transformation d'un RCD en formule CNF par l'utilisation des tuples conflits.

Exemple 12 Considérons $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD tel que $\mathcal{X} = \{x, x'\}$. Supposons que $D_x = \{0, 1\}$, $D_{x'} = \{0, 1\}$, et que la seule contrainte décrite sous la forme de conflits est $\bar{c}_{(x, x')} = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. Construisons $\Sigma_{\mathcal{P}}^{direct}$ la formule CNF associée par l'encodage direct.

$$vars(\Sigma_{\mathcal{P}}^{direct}) = \{[x = 0], [x = 1], [x' = 0], [x' = 1]\}$$

$\Sigma_{\mathcal{P}}^{ALO}$	\wedge $[x = 0] \vee [x = 1]$ $[x' = 0] \vee [x' = 1]$
$\Sigma_{\mathcal{P}}^{AMO}$	\wedge $\neg [x = 0] \vee \neg [x = 1]$ $\neg [x' = 0] \vee \neg [x' = 1]$
$\Sigma_{\mathcal{P}}^{Conflit}$	\wedge $\neg [x = 0] \vee \neg [x' = 0]$ $\neg [x = 0] \vee \neg [x' = 1]$ $\neg [x = 1] \vee \neg [x' = 0]$ $\neg [x = 1] \vee \neg [x' = 1]$

4.3.2 Vers de nouvelles approches exploitant les transformations

En pratique, nous bénéficions des avantages, par exemple l'apprentissage ou l'analyse de conflits, de la représentation propositionnelle pour répondre au problème CSP. Remarquons qu'il est difficile de comparer les formes de cohérence locale utilisées dans les deux cadres : la propagation unitaire et la cohérence d'arcs. L'exemple 13 montre un RCD qui n'est pas arcs-cohérent mais qui est cohérent localement par la propagation unitaire. La cohérence d'arcs s'attache à vérifier si chaque valeur du domaine d'une variable possède un tuple dans chaque contrainte qui l'implique. La propagation unitaire propage uniquement les clauses unitaires. Posons \mathcal{P} un RCD et $\Sigma_{\mathcal{P}}^{direct}$ la formule associée. Nous pouvons affirmer que si $UP(\neg[x = v]) \models^* \neg[x = v]$, alors $v \notin D_x$ dans $AC(\mathcal{P})$. Dans le cas général, pour obtenir l'équivalence entre la cohérence d'arcs d'un RCD et la propagation unitaire d'une formule CNF, il est nécessaire d'utiliser d'autres procédés, tels que la résolution, la subsumption, ou encore la propagation de littéraux purs.

Exemple 13 Revenons sur l'exemple 12. Clairement, ce RCD est incohérent. Toute valeur est en conflit avec les autres. La cohérence d'arcs permet donc de supprimer toutes les valeurs. La propagation unitaire est un algorithme de propagation basé sur les clauses unitaires. Cependant, la formule $\Sigma_{\mathcal{P}}^{direct}$ ne possède aucune clause unitaire. La notion de « tuple conflit » des RCD perd son sens dans la formule CNF, la propagation unitaire n'est pas adaptée pour être aussi efficace que la cohérence d'arcs. Dans ce cas, il est nécessaire d'utiliser la résolution afin de déterminer l'incohérence.

Pour rappel, $resolution(p, p \vee c, \neg p \vee c') = c \vee c'$.

p	$p \vee c$	$\neg p \vee c'$	$resolution(p, p \vee c, \neg p \vee c')$
$[x' = 0]$	$[x' = 0] \vee [x' = 1]$	$\neg[x' = 0] \vee \neg[x = 0]$	$[x' = 1] \vee \neg[x = 0]$
$[x' = 1]$	$[x' = 1] \vee \neg[x = 0]$	$\neg[x' = 1] \vee \neg[x = 0]$	$\neg[x = 0]$
$[x = 0]$	$[x = 0] \vee [x = 1]$	$\neg[x = 0]$	$[x = 1]$
$[x = 1]$	$[x = 1]$	$\neg[x = 1] \vee \neg[x' = 0]$	$\neg[x' = 0]$
$[x = 1]$	$[x = 1]$	$\neg[x = 1] \vee \neg[x' = 1]$	$\neg[x' = 1]$
$[x' = 0]$	$\neg[x' = 0]$	$[x' = 0] \vee [x' = 1]$	$[x' = 1]$

Appliquer la règle de résolution nous amène à déduire $[x' = 1]$ et $\neg[x' = 1]$, ce qui montre une incohérence. La formule est donc prouvée insatisfiable par la règle de résolution.

Pour déterminer la satisfiabilité du réseau de contraintes en utilisant cette réduction vers une formule propositionnelle, il n'est pas nécessaire d'imposer de construire une instanciation complète. Il est possible de réduire uniquement le réseau de contraintes, en n'imposant pas une valeur par domaine. En d'autres termes, les clauses *AMO* ne sont pas nécessaires. Ce résultat permet d'affirmer que la cohérence de la formule $\Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{Conflit}$ est suffisante pour déterminer la cohérence du réseau de contraintes \mathcal{P} .

Propriété 10 Soient \mathcal{P} un RCD et $\Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{Conflit}$ une formule CNF associée à \mathcal{P} . \mathcal{P} est cohérent si et seulement si $\Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{Conflit}$ est satisfiable.

Preuve \rightarrow Trivialement, $\Sigma_{\mathcal{P}}^{direct} \models \Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{Conflit}$. Donc, si $\Sigma_{\mathcal{P}}^{direct}$ est satisfiable, alors $\Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{Conflit}$ est satisfiable.

\leftarrow Supposons maintenant que nous ayons un modèle μ de $\Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{Conflit}$, tel qu'il existe une variable $x \in \mathcal{X}$, pour laquelle $[x = v_i] \in \mu$ pour tout $v_i \in D_x$. Clairement, μ n'est pas un modèle de $\Sigma_{\mathcal{P}}^{direct}$ car il ne satisfait pas la formule $\Sigma_{\mathcal{P}}^{AMO}$. Néanmoins, les formules $\Sigma_{\mathcal{P}}^{ALO}$ et $\Sigma_{\mathcal{P}}^{Conflit}$ sont satisfaites. Une instanciation complète de \mathcal{P} est une application de l'ensemble des variables vers les valeurs des domaines. Or, μ ne peut être associé à une instanciation complète car μ correspond à plusieurs

affectations pour x . Posons μ' une interprétation de Σ telle que $\mu'([x' = v]) = \mu([x' = v])$ ssi $x' \neq x$, $\exists v_i \in D_x : [x = v_i] \in \mu'$ et $\forall v_j \in D_x, i \neq j : \neg[x = v_j] \in \mu'$. μ' satisfait $\Sigma_{\mathcal{P}}^{ALO}$ et $\Sigma_{\mathcal{P}}^{AMO}$ puisqu'il existe une et une seule affectation pour toute variable $x \in \mathcal{X}$, telle que $[x = v] \in \mu'$. De plus, $\Sigma_{\mathcal{P}}^{Conflit}$ est également toujours satisfaite, car cette formule est uniquement composée de littéraux négatifs. μ satisfait $\Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{Conflit}$ donc μ satisfait $\Sigma_{\mathcal{P}}^{Conflit}$. Or, par construction de μ' , certains littéraux positifs de μ ont été modifiées pour devenir négatifs dans μ' . μ satisfait donc $\Sigma_{\mathcal{P}}^{Conflit}$ ce qui ne peut pas falsifier la formule. Donc μ' satisfait $\Sigma_{\mathcal{P}}^{direct}$, et \mathcal{P} est cohérent.

Par conséquent, \mathcal{P} est cohérent si et seulement si $\Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{Conflit}$ est satisfiable. \square

Cette propriété et sa preuve nous permettent également de déduire que tout modèle de $\Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{Conflit}$ est associé à un ensemble de solutions de \mathcal{P} . À partir d'un modèle de cette formule, il est possible de déterminer une solution en $\mathcal{O}(nd)$ dans le pire des cas. Pour chaque variable du RCD, il suffit de poser une affectation et d'éliminer toutes les autres. Cette approche semble intéressante puisqu'elle permet de manipuler la formule CNF pour répondre au problème sans avoir à imposer une valeur par variable. Cependant, l'arbre de recherche contient 2^{dn} interprétations complètes. Puisque le problème de la satisfiabilité s'attache uniquement à déterminer la cohérence d'un RCD, voire à la prouver en donnant une seule solution, les clauses *AMO* permettent de contraindre le problème et de réduire le nombre d'interprétations complètes à évaluer. Néanmoins, cette propriété peut être appréciable dans le cadre d'autres approches, telle qu'une méthode de recherche incomplète par recherche locale.

Comme énoncé précédemment, la forme des clauses conflits fournies précédemment est obtenue directement en transformant la contrainte en formule propositionnelle CNF. Cependant, cette représentation n'est pas unique. Puisque la règle de résolution en logique propositionnelle permet de déduire de nouvelles clauses, il est possible de donner une autre forme aux clauses conflits. Par exemple, appliquer la règle de résolution entre les formules $\Sigma_{\mathcal{P}}^{Conflit}$ et $\Sigma_{\mathcal{P}}^{ALO}$ permet de déduire un ensemble des clauses décrit par la formule suivante :

$$\Sigma_{\mathcal{P}}^{Conflit_2} = \bigwedge_{(c \in \mathcal{C})} \left(\bigwedge_{(\tau \in \bar{c})} \bigvee_{(x \in scp(\bar{c}))} \bigvee_{(v \in D_x : v \neq \tau[x])} [x = v] \right)$$

Remarquons que la complexité algorithmique en temps (resp. en espace) dans le pire des cas pour la construction de la formule $\Sigma_{\mathcal{P}}^{Conflit_2}$ est en $\mathcal{O}(erd^{r+1})$ (resp. $\mathcal{O}(rd)$), et la formule contient ed^r clauses de taille rd . De plus, nous pouvons retrouver la formule conflit initiale $\Sigma_{\mathcal{P}}^{Conflit}$ par résolution sur les formules $\Sigma_{\mathcal{P}}^{AMO}$ et $\Sigma_{\mathcal{P}}^{Conflit_2}$.

$$\Sigma_{\mathcal{P}}^{Conflit} \wedge \Sigma_{\mathcal{P}}^{AMO} \models \Sigma_{\mathcal{P}}^{Conflit_2}$$

$$\Sigma_{\mathcal{P}}^{Conflit_2} \wedge \Sigma_{\mathcal{P}}^{AMO} \models \Sigma_{\mathcal{P}}^{Conflit}$$

En utilisant la règle d'affaiblissement sur ces formules par l'introduction d'une formule à droite et à gauche, nous pouvons déduire les résultats suivants :

$$\Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{Conflit} \models \Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{Conflit_2}$$

$$\Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{Conflit} \equiv \Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{Conflit_2}$$

Si $\Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{Conflit_2}$ est insatisfiable alors le RCD est incohérent. De plus, en exploitant l'équivalence $\Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{Conflit} \equiv \Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{Conflit_2}$, il est possible de montrer, par un

raisonnement analogue à la propriété 10 que prouver la satisfiabilité de $\Sigma_{AMO} \wedge \Sigma_{\mathcal{P}}^{Conflit_2}$ suffit à prouver la cohérence du RCD. Si un modèle μ de la formule n'est pas associé à une instantiation du RCD, il existe une variable x qui n'ait pas d'affectation, donc $\forall v \in D_x : \neg[x = v] \in \mu$. Il est alors possible de construire un modèle vérifiant les clauses AMO , qui satisferont les clauses de $\Sigma_{\mathcal{P}}^{Conflit_2}$ puisqu'elles ne sont composées que de littéraux positifs.

Exemple 14 Pour illustrer la construction de la formule $\Sigma_{\mathcal{P}}^{Conflit_2}$ pour le RCD de l'exemple 13, nous présentons tout d'abord la résolution des clauses conflits avec les clauses ALO afin de déduire les clauses $\Sigma_{\mathcal{P}}^{Conflit_2}$. La résolution s'écrit $resolution(p, p \vee c, \neg p \vee c') = c \vee c'$. Les clauses représentées en bleu sont des clauses de la formule $\Sigma_{\mathcal{P}}^{Conflit_2}$.

p	$p \vee c$	$\neg p \vee c'$	$resolution(p, p \vee c, \neg p \vee c')$
$[x = 0]$	$\neg[x = 0] \vee \neg[x' = 0]$	$[x = 0] \vee [x = 1]$	$\neg[x' = 0] \vee [x = 1]$
$[x' = 0]$	$\neg[x' = 0] \vee [x = 1]$	$[x' = 0] \vee [x' = 1]$	$[x' = 1] \vee [x = 1]$
$[x = 0]$	$\neg[x = 0] \vee \neg[x' = 1]$	$[x = 0] \vee [x = 1]$	$\neg[x' = 1] \vee [x = 1]$
$[x' = 1]$	$\neg[x' = 1] \vee [x = 1]$	$[x' = 1] \vee [x' = 0]$	$[x' = 0] \vee [x = 1]$
$[x = 1]$	$\neg[x = 1] \vee \neg[x' = 0]$	$[x = 1] \vee [x = 0]$	$\neg[x' = 0] \vee [x = 0]$
$[x' = 0]$	$\neg[x' = 0] \vee [x = 0]$	$[x' = 0] \vee [x' = 1]$	$[x' = 1] \vee [x = 0]$
$[x = 1]$	$\neg[x = 1] \vee \neg[x' = 1]$	$[x = 0] \vee [x = 1]$	$\neg[x' = 1] \vee [x = 0]$
$[x' = 1]$	$\neg[x' = 1] \vee [x = 0]$	$[x' = 1] \vee [x' = 0]$	$[x' = 0] \vee [x = 0]$

$\Sigma_{\mathcal{P}}^{AMO}$	\wedge	$\neg[x = 0] \vee \neg[x = 1]$
		$\neg[x' = 0] \vee \neg[x' = 1]$
$\Sigma_{\mathcal{P}}^{Conflit_2}$	\wedge	$[x = 1] \vee [x' = 1]$
	\wedge	$[x = 1] \vee [x' = 0]$
	\wedge	$[x = 0] \vee [x' = 1]$
	\wedge	$[x = 0] \vee [x' = 0]$

Puisque $\Sigma_{\mathcal{P}}^{AMO} \vee \Sigma_{\mathcal{P}}^{Conflit_2}$ est insatisfiable, \mathcal{P} est incohérent.

La figure 4.1 illustre les ensembles de modèles pour les formules $\Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{Conflit}$, $\Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{Conflit_2}$ et $\Sigma_{\mathcal{P}}^{direct}$. Il est clair que tous les modèles de $\Sigma_{\mathcal{P}}^{direct}$ peuvent être induits des modèles de $\Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{Conflit}$ et $\Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{Conflit_2}$.

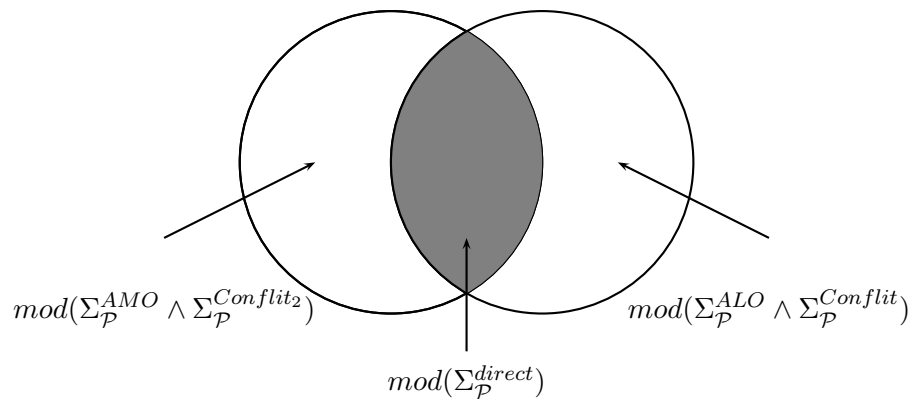


FIGURE 4.1 – Illustration des modèles des formules CNF basées sur les conflits

Il est ainsi possible de définir de nombreuses formes de transformations des contraintes des RCD vers les formules CNF. Cette nouvelle formule pour représenter les conflits s'appuie de la règle de résolution pour la mettre en évidence, mais peut être facilement déduite du RCD. En particulier, dans le cas de RCD binaires, la transformation est polynomiale en temps dans la taille du RCD. Les algorithmes de recherche et de cohérence locale peuvent avoir un impact différent en fonction de la représentation du RCD. Ce qui nous semble intéressant dans cette démarche est de comprendre et d'exploiter ce type de transformation afin de faire apparaître de nouvelles approches pour la résolution des RCD.

4.4 Modélisation des contraintes : les supports

4.4.1 Supports et Cohérence d'Arcs

Pour transformer un RCD en exploitant les tuples supports, il est nécessaire d'utiliser les clauses *ALO* et *AMO* afin de créer une instantiation complète. Comme pour les tuples conflits de la transformation directe, nous pouvons assimiler une contrainte à une formule DNF. Une contrainte c est satisfaite par \mathcal{I} si et seulement si il existe un tuple $\tau \in c$, tel que pour toute variable $x \in \text{scp}(c)$, nous ayons $(x = \tau[x]) \in \mathcal{I}$.

$$\Sigma_{\mathcal{P}}^S = \bigwedge_{(c \in \mathcal{C})} \left(\bigvee_{(\tau \in c)} \bigwedge_{(x \in \text{scp}(c))} [x = \tau[x]] \right)$$

L'inconvénient qui semble apparaître pour cette approche, est qu'elle amène à transformer une formule propositionnelle de DNF en CNF. En considérant, dans le cas général, une contrainte de d^r tuples impliquant r variables, la formule propositionnelle CNF résultante développera r^{d^r} clauses. La formule est donc très grande, la calculer et la simplifier sont des tâches difficiles en pratique.

Kasif [KASIF 1990] propose une transformation différente avec l'*encodage de supports, support encoding* en anglais. Cette transformation naturelle s'appuie sur les tuples supports des contraintes. Cette approche permet d'obtenir l'équivalence entre propagation unitaire dans la formule CNF avec la cohérence d'arcs dans le RCD. La cohérence d'arcs permet de prouver qu'une instantiation partielle est localement cohérente et qu'une instantiation complète du RCD est une solution. Simuler la cohérence d'arcs par la propagation unitaire dans la formule CNF permet alors de montrer si une interprétation complète est un modèle. Gent utilise cette transformation pour appliquer la cohérence d'arcs sur les RCD en passant par une formule CNF. Il l'approche dans le cadre des réseaux de contraintes binaire et évoque la généralisation aux arités supérieures comme identique. Nous présentons ici une généralisation de la transformation des contraintes du RCD en formule CNF.

Définition 76 (Formule des supports par cohérence d'arcs) Soit \mathcal{P} un RCD. La formule propositionnelle associée à \mathcal{P} par encodage de supports est :

$$\Sigma_{\mathcal{P}}^{\text{GAC}} = \bigwedge_{(c \in \mathcal{C})} \bigwedge_{(x \in \text{scp}(c))} \bigwedge_{(v \in D_x)} \bigwedge_{(x' \in \text{scp}(c): x \neq x')} \left([x = v] \Rightarrow \bigvee_{(\tau \in c: \tau[x]=v)} [x' = \tau[x']] \right)$$

Gent propose d'appliquer la cohérence d'arcs sur le RCD avant la transformation, cependant les valeurs non arcs-cohérentes sont détectées naturellement par cette généralisation. Si une valeur v de x est non arcs-cohérentes, alors une clause $[x = v] \Rightarrow \perp$ est créée, ce qui correspond à la clause

unitaire $\neg[x = v]$. Pour que la formule CNF coïncide avec la fermeture par cohérence d'arcs, il est alors nécessaire d'appliquer la propagation unitaire pour propager cette valeur.

La complexité algorithmique en temps (resp. en espace) dans le pire des cas d'une telle transformation est $\mathcal{O}(er^2d^r)$ (resp. $\mathcal{O}(d^2)$). La formule CNF construite possède erd^r clauses de taille $d + 1$ dans le pire des cas. En particulier, dans le cas de RCD binaires, la complexité algorithmique en temps dans le pire des cas est $\mathcal{O}(ed^2)$ [KASIF 1990, GENT 2002].

En ajoutant les formules $\Sigma_{\mathcal{P}}^{ALO}$ et $\Sigma_{\mathcal{P}}^{AMO}$ afin de construire de pouvoir associée une instantiation complète à toute interprétation complète de la formule CNF, nous pouvons définir l'encodage de supports.

Définition 77 (transformation des supports) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD. La formule propositionnelle associée à \mathcal{P} par l'encodage des supports est :

$$\Sigma_{\mathcal{P}}^{Support} = \Sigma_{\mathcal{P}}^{ALO} \wedge \Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^{AC}$$

Propriété 11 Soient \mathcal{P} un RCD et $\Sigma_{\mathcal{P}}^{Support}$ la formule CNF associée à \mathcal{P} . \mathcal{P} est cohérent si et seulement si $\Sigma_{\mathcal{P}}^{Support}$ est satisfiable.

La preuve est également immédiate dans ce cadre. Toute interprétation μ de la formule propositionnelle vérifiant les clauses *ALO* et *AMO* est associée à une instantiation complète \mathcal{I} du réseau de contraintes. Puisque la propagation unitaire est équivalente à la cohérence d'arcs dans ce cadre, on en déduit que \mathcal{I} est arcs-cohérent si et seulement si la propagation unitaire ne détecte pas d'incohérence pour l'interprétation μ dans $\Sigma_{\mathcal{P}}^{AC}$. Donc \mathcal{I} est une solution si et seulement si μ est un modèle. Par cette transformation, Kasif exhibe une nouvelle réduction fonctionnelle polynomiale du problème CSP vers les SAT. La transformation exploitant les supports est illustrée Annexe B.1.

Le tableau 4.1 présente un récapitulatif des dimensions des différentes formules CNF présentées et de la complexité algorithmiques pour les calculer.

Formule	Dimensions		Complexité algorithmique	
	Nombre de clauses	Taille des clauses	Temps	Espace
$\Sigma_{\mathcal{P}}^{ALO}$	n	d	$\mathcal{O}(nd)$	$\mathcal{O}(d)$
$\Sigma_{\mathcal{P}}^{AMO}$	$n \frac{d(d-1)}{2}$	2	$\mathcal{O}(nd^2)$	$\mathcal{O}(1)$
$\Sigma_{\mathcal{P}}^{Conflit}$	ed^r	r	$\mathcal{O}(erd^r)$	$\mathcal{O}(r)$
$\Sigma_{\mathcal{P}}^{Conflit_2}$	ed^r	rd	$\mathcal{O}(erd^{r+1})$	$\mathcal{O}(rd)$
$\Sigma_{\mathcal{P}}^{AC}$	erd^r	$d + 1$	$\mathcal{O}(erd^r)$	$\mathcal{O}(d^2)$

TABLE 4.1 – Dimensions de la formule CNF et Complexité algorithmique dans le pire des cas

4.4.2 Vers de nouvelles modélisations de cohérence locale

Pour aller un peu plus loin, revenons sur la première définition de contraintes sous la forme de supports $\Sigma_{\mathcal{P}}^S$.

$$\Sigma_{\mathcal{P}}^S = \bigwedge_{(c \in \mathcal{C})} \left(\bigvee_{(\tau \in c)} \bigwedge_{(x \in scp(c))} [x = \tau[x]] \right)$$

Trivialement, nous pouvons remarquer que les deux formules $\Sigma_{\mathcal{P}}^S$ et $\Sigma_{\mathcal{P}}^{GAC}$ ne sont pas sémantiquement équivalentes. La formule $\Sigma_{\mathcal{P}}^{AMO} \wedge \Sigma_{\mathcal{P}}^S$ suffit pour décider de la cohérence de \mathcal{P} ⁵. En développant la formule DNF en CNF, nous obtenons systématiquement les clauses *ALO* associée à chaque variable de \mathcal{P} . Intuitivement, transformer les tuples supports en exploitant la cohérence d'arcs $\Sigma_{\mathcal{P}}^{GAC}$ suffit pour répondre à la cohérence d'une interprétation. $\Sigma_{\mathcal{P}}^{GAC}$ peut donc être vue comme une approximation de $\Sigma_{\mathcal{P}}^S$, facile à calculer et suffisante dans notre cadre. Cependant, $\Sigma_{\mathcal{P}}^S$ est bien plus expressive, puisqu'elle décrit parfaitement la contrainte. $\Sigma_{\mathcal{P}}^S$ laisse entrevoir la possibilité, dans le cas de RCD d'arité quelconque, que d'autres formules CNF, basées sur les supports, peuvent décrire des formes de cohérence locale plus fortes que GAC.

Remarquons que nous pouvons déduire, par la règle de résolutions entre les formules $\Sigma_{\mathcal{P}}^{ALO}$, $\Sigma_{\mathcal{P}}^{AMO}$ et $\Sigma_{\mathcal{P}}^{AC}$, une formule que l'on peut assimiler à la cohérence de chemins. Et en poursuivant le raisonnement, la règle de résolution permet de déduire des formes de cohérence locale. L'idée générale est de décrire ces formes de cohérences locales sans avoir à les déduire par la règle de résolution. En nous appuyant sur la définition 22 de la k -cohérence, nous définissons la formule CNF permettant de modéliser la k -cohérence. La modélisation de la chemins-cohérence est présentée Annexe B.2.

$$\Sigma_{\mathcal{P}}^k = \bigwedge_{(X \subseteq \mathcal{X}: |X|=k-1)} \bigwedge_{(x_k \subseteq \mathcal{X} \setminus X)} \left(\left(\bigwedge_{(x \in X, v \in D_x)} [x = v] \right) \Rightarrow \left(\bigwedge_{\substack{v_k \in D_{x_k}: \forall x \in X, v \in D_x, \\ \{(x,v), (x_k, v_k)\} \text{ soit AC}}} [x_k = v_k] \right) \right)$$

Toute formule $\Sigma_{\mathcal{P}}^k$ peut être déduite par résolution sur les formules $\Sigma_{\mathcal{P}}^{ALO}$, $\Sigma_{\mathcal{P}}^{AMO}$ et $\Sigma_{\mathcal{P}}^{k-1}$ et $\Sigma_{\mathcal{P}}^2$. La motivation est de procurer à la recherche et à la propagation unitaire un pouvoir d'inférence plus important qu'avec AC. Cependant, il existe une contre-partie évidente entre le pouvoir d'inférence offerte par la transformation et la complexité algorithmique en temps dans le pire des cas pour la calculer.

4.5 Conclusion

Exploiter les transformations vers les formules CNF pour répondre au problème de la cohérence d'un RCD présente des aspects intéressants en pratique. L'approche par la transformation support a été exploitée en pratique [GENT 2002, TAMURA *et al.* 2009], et peut même s'avérer parfois plus efficace que les algorithmes des RCD pour répondre au problème CSP, comme le *solveur* CSP [TAMURA *et al.* 2008] exploitant les outils développés pour le problème SAT. Au-delà de cette utilisation, les deux transformations peuvent être un cadre d'inspiration pour le développement de nouvelles approches. En effet, en exploitant la transformation *conflict*, nous pouvons répondre à la cohérence d'un RCD sans imposer *au plus une* valeur par variable, mais nous pouvons également imaginer de nouvelles modélisation des contraintes afin de ne pas imposer *au moins une* valeur par variable. La transformation *support* permet de faire la correspondance entre *propagation unitaire* et *cohérence d'arcs*. Cependant, elle ne retranscrit pas entièrement le sens des contraintes, mais elle est suffisante pour répondre à la cohérence du RCD. En utilisant la règle de résolution, nous pouvons voir apparaître des sous-formules représentant des formes de cohérences plus fortes. Puisque la résolution est peu utilisée en cours de recherche dans les algorithmes modernes, il est possible de définir des transformations s'appuyant sur d'autres formes de cohérence locale afin d'apporter un pouvoir d'inférence logique plus fort que AC. Comprendre la transformation des RCD ternaires en formule CNF montre une voie pour la transformation de RCQ en formules CNF, afin d'apporter un support d'études théoriques et pratiques pour répondre à la cohérence de RCQ.

5. Remarquons que si nous ajoutons la mutuelle exclusivité des tuples dans notre formule $\Sigma_{\mathcal{P}}^S$ afin de spécifier qu'un et un seul tuple par contrainte suffit, alors $\Sigma_{\mathcal{P}}^S$ suffit pour décider de la satisfiabilité de \mathcal{P} .

Chapitre 5

Du problème CSP qualitatif au problème SAT

Sommaire

5.1	Introduction	85
5.2	Identification des variables propositionnelles	86
5.2.1	Variables propositionnelles	86
5.2.2	Du scénario à l'interprétation	86
5.3	Description des contraintes en logique propositionnelle	87
5.3.1	Utilisation des conflits	88
5.3.2	Utilisation des supports	90
5.4	Conclusion	91

5.1 Introduction

Dans ce chapitre, nous proposons d'étudier la modélisation de problèmes de nature temporelle ou spatiale en exploitant les diverses ressources, théoriques et pratiques, offertes par le cadre propositionnel. Certaines études ont mis en évidence de telles transformations vers la logique propositionnelle. Afin de mettre en évidence la classe des relations ORD-Horn pour l'algèbre des intervalles, une axiomatisation d'un ordre sur les bornes des intervalles a été défini dans le langage propositionnel [NEBEL & BÜRCKERT 1995]. Sur le même formalisme qualitatif, Pham *et al.* [PHAM *et al.* 2006] utilise l'approche de Walsh et Kasif [KASIF 1990, WALSH 2000] pour modéliser les bornes des intervalles en logique propositionnelle à l'aide de conflits et de supports.

En nous appuyant sur la définition de transformations existantes de RCD vers les formules CNF, nous proposons une transformation générique des RCQ vers les formules CNF. Nous proposons d'associer un scénario d'un RCQ à une interprétation du cadre propositionnel, et de vérifier sa cohérence à l'aide de la table de composition. Pour vérifier la cohérence de celle-ci, nous proposons deux approches. L'une exprimant les tuples qui ne coïncident pas avec la table de composition, l'autre permettant d'obtenir la correspondance entre la méthode de la faible composition des RCQ avec la propagation unitaire pour les formules CNF. Cependant, la faible composition n'est pas complète pour la cohérence d'un scénario dans le cas général. En nous concentrant sur certains formalismes qualitatifs, nous montrons que nous pouvons répondre à la cohérence d'un RCQ en répondant à la cohérence de la formule CNF associée.

Dans ce chapitre, nous présentons et étudions deux transformations permettant de transformer un RCQ en formule CNF, sur le même schéma que les transformations présentées des RCD vers les for-

mules CNF (Chapitre 4). Dans un premier temps, nous définissons les variables propositionnelles et faisons le lien entre scénario et interprétation. Pour modéliser la propriété de faible composition, nous proposons ensuite deux approches, l'une s'appuyant sur les conflits et l'autre s'appuyant sur la propriété de faible composition. Nous pouvons ainsi garantir que notre transformation, polynomiale en temps dans la taille du RCQ, permet d'associer à tout modèle de la formule CNF un scénario cohérent pour la faible composition du RCQ.

5.2 Identification des variables propositionnelles

5.2.1 Variables propositionnelles

Dans un premier temps, nous souhaitons identifier les propositions et définir les variables propositionnelles, afin de caractériser un scénario et vérifier la cohérence de celui-ci. Un RCQ est décrit par des contraintes entre les variables, et leurs relations complexes associées. Considérons $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ construit sur un formalisme qualitatif défini sur les relations atomiques \mathcal{B} et $\Sigma_{\mathcal{N}}$ la formule associée à \mathcal{N} . L'idée générale est de décrire un scénario \mathcal{P} et de vérifier sa cohérence. Nous exprimons naturellement la microstructure par la proposition « $r \in Q_{(v_i, v_j)}$ » pour tout $r \in \mathcal{B}$ et $v_i, v_j \in \mathcal{V}$, par la variable propositionnelle $[r \in Q_{(v_i, v_j)}]$. Ainsi, $[r \in Q_{(v_i, v_j)}] = 1$ caractérise l'affirmation $r \in Q_{(v_i, v_j)}$ et $[r \in Q_{(v_i, v_j)}] = 0$ représente l'affirmation $r \notin Q_{(v_i, v_j)}$. Considérons $\Sigma_{\mathcal{N}}$ la formule associée à \mathcal{N} . L'ensemble des variables propositionnelles de la $\Sigma_{\mathcal{N}}$ est composé de $n|\mathcal{B}|$ éléments, où n est le nombre de variables de \mathcal{N} , tel que :

$$\text{vars}(\Sigma_{\mathcal{N}}) = \{[r \in Q_{(v_i, v_j)}] : v_i, v_j \in \mathcal{V} \wedge r \in Q_{(v_i, v_j)}\}$$

Proposition 2 (Du modèle SAT au RCQ) Soit \mathcal{N} un RCQ et $\Sigma = \top$ la formule propositionnelle associée. À tout modèle μ de Σ peut être associé $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ un RCQ, tel que pour tout $[r \in Q_{(v_i, v_j)}] \in \text{vars}(\Sigma)$, $r \in Q'_{(v_i, v_j)}$ si et seulement si $[r \in Q_{(v_i, v_j)}] \in \mu$ (c'est-à-dire $\mu([r \in Q_{(v_i, v_j)}]) = 1$).

Remarquons, par la proposition 2, que si \mathcal{N}' est construit à partir d'un modèle μ de la formule propositionnelle $\Sigma_{\mathcal{N}}$ associée à \mathcal{N} , alors $\mathcal{N}' \subseteq \mathcal{N}$.

5.2.2 Du scénario à l'interprétation

En nous appuyant sur la proposition 2, nous pouvons restreindre l'ensemble des modèles afin de construire un scénario. Un scénario est un RCQ $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ atomique. Par conséquent :

- pour tout $v_i, v_j \in \mathcal{V}$, il existe *au moins une* relation atomique $r \in \mathcal{B}$ telle que $r \in Q_{(v_i, v_j)}$;
- pour tout $v_i, v_j \in \mathcal{V}$, il existe *au plus une* relation atomique $r \in \mathcal{B}$ telle que $r \in Q_{(v_i, v_j)}$.

Pour imposer cela, nous définissons les formules *au moins une* $\Sigma_{\mathcal{N}}^{ALO}$ et *au plus une* $\Sigma_{\mathcal{N}}^{AMO}$.

Définition 78 (Formule ALO) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ. La formule « au moins une » associée à \mathcal{N} est :

$$\Sigma_{\mathcal{N}}^{ALO} = \bigwedge_{(v_i, v_j \in \mathcal{V})} \bigvee_{(r \in Q_{(v_i, v_j)})} [r \in Q_{(v_i, v_j)}]$$

Définition 79 (Formule AMO) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ. La formule « au plus une » associée à \mathcal{N} est :

$$\Sigma_{\mathcal{N}}^{AMO} = \bigwedge_{(v_i, v_j)} \bigvee_{(r, r' \in Q_{(v_i, v_j)} : r \neq r')} \neg[r \in Q_{(v_i, v_j)}] \vee \neg[r' \in Q_{(v_i, v_j)}]$$

Remarquons qu'il est possible de vérifier en temps polynomial dans la taille d'un RCQ si celui-ci contient une contrainte vide. Nous supposons donc que la transformation traite uniquement les RCQ qui ne sont pas trivialement incohérents.

Proposition 3 Soit \mathcal{N} un RCQ. À tout modèle de la formule $\Sigma_{\mathcal{N}}^{ALO} \wedge \Sigma_{\mathcal{N}}^{AMO}$ est associé un scénario \mathcal{N}' tel que $\mathcal{N}' \subseteq \mathcal{N}$.

Preuve Considérons deux RCQ $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$, et construisons un RCQ $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ à partir d'un modèle μ de $\Sigma = \Sigma_{\mathcal{N}}^{ALO} \wedge \Sigma_{\mathcal{N}}^{AMO}$.

Tout d'abord, puisque μ est un modèle de $\Sigma = \Sigma_{\mathcal{N}}^{ALO} \wedge \Sigma_{\mathcal{N}}^{AMO}$ et d'après la proposition 2, il est possible de construire \mathcal{N}' tel que : pour toute variable propositionnelle $[r \in Q_{(v_i, v_j)}] \in \text{vars}(\Sigma)$, $r \in Q'_{(v_i, v_j)}$ si et seulement si $[r \in Q_{(v_i, v_j)}] \in \mu$. Par cette définition, nous pouvons affirmer que $\mathcal{N}' \sqsubseteq \mathcal{N}$.

Puisque μ est un modèle de Σ , μ est un modèle de $\Sigma_{\mathcal{N}}^{ALO}$ et $\Sigma_{\mathcal{N}}^{AMO}$. μ est un modèle de $\Sigma_{\mathcal{N}}^{ALO}$. Par conséquent, pour tout $v_i, v_j \in \mathcal{V}$, il existe $r \in Q_{(v_i, v_j)}$ telle que $[r \in Q_{(v_i, v_j)}] \in \mu$. Par construction, il existe au moins une relation $r \in Q_{(v_i, v_j)}$ telle que $r \in Q'_{(v_i, v_j)}$. \mathcal{N}' ne possède donc aucune contrainte vide. μ est un modèle de $\Sigma_{\mathcal{N}}^{AMO}$. Pour tout $v_i, v_j \in \mathcal{V}$ et tout $r, r' \in Q_{(v_i, v_j)}$ avec $r \neq r'$, $\neg[r \in Q_{(v_i, v_j)}]$ ou $\neg[r' \in Q_{(v_i, v_j)}]$. Par construction, pour tout $r, r' \in Q_{(v_i, v_j)}$ avec $r \neq r'$, $r \notin Q'_{(v_i, v_j)}$ ou $r' \notin Q'_{(v_i, v_j)}$. Par conséquent, pour tout $v_i, v_j \in \mathcal{V}$, $|Q'_{(v_i, v_j)}| = 1$. \mathcal{N}' est donc un scénario issu du modèle μ de la formule $\Sigma_{\mathcal{N}}^{ALO} \wedge \Sigma_{\mathcal{N}}^{AMO}$. \square

Nous rappelons que nous manipulons uniquement des RCQ normalisés. Puisque nous souhaitons construire un scénario, nous pouvons nous appuyer sur la propriété 1 pour restreindre l'ensemble $\text{vars}(\Sigma)$ à toutes variables $v_i, v_j \in \mathcal{V}$ telles que $i < j$. Cela nous permet de modifier la construction d'un RCQ \mathcal{N}' à partir d'un modèle μ d'une formule $\Sigma_{\mathcal{N}}$ associée à un RCQ \mathcal{N} (Proposition 2), afin d'imposer la normalisation :

- pour tout $v_i \in \mathcal{V}$, $\text{id} \in Q'_{(v_i, v_i)}$;
- pour tout $v_i, v_j \in \mathcal{V}$ avec $i < j$, si $[r \in Q_{(v_i, v_j)}] \in \mu$, alors $r \in Q'_{(v_i, v_j)}$ et $r^{-1} \in Q'_{(v_j, v_i)}$.

La nature de la proposition 3 ne change pas, elle impose uniquement que le scénario soit normalisé et qu'il ne soit donc pas trivialement incohérent. Nous pouvons maintenant vérifier la cohérence d'un scénario obtenu par un modèle de la formule $\Sigma_{\mathcal{N}}^{ALO} \wedge \Sigma_{\mathcal{N}}^{AMO}$.

Pour calculer les dimensions de ces formules et la complexité algorithmique pour les construire, rappelons que le RCQ est un graphe complet de n variables construit sur un formalisme qualitatif défini sur les relations atomiques \mathcal{B} . La formule $\Sigma_{\mathcal{N}}^{ALO}$ porte sur chaque contrainte, et chaque relation atomique qui la compose. Elle est donc composée de $\binom{n}{2}$ clauses de taille $|\mathcal{B}|$ dans le pire des cas. La complexité algorithmique en temps (resp. en espace) dans le pire des cas pour construire cette formule est en $\mathcal{O}(n^2|\mathcal{B}|)$ (resp. $\mathcal{O}(|\mathcal{B}|)$). De plus, pour chaque contrainte, la formule $\Sigma_{\mathcal{N}}^{AMO}$ décrit la mutuelle exclusivité des $\binom{|\mathcal{B}|}{2}$ couples de relations atomiques. Dans le pire des cas, $\Sigma_{\mathcal{N}}^{AMO}$ est composée de $\binom{n}{2} \binom{|\mathcal{B}|}{2}$ clauses de taille 2. La complexité algorithmique temporelle (resp. spatiale) dans le pire des cas est en $\mathcal{O}(n^2|\mathcal{B}|^2)$ et $\mathcal{O}(1)$ respectivement.

5.3 Description des contraintes en logique propositionnelle

Les transformations des RCD en formules CNF présentées Chapitre 4 permettent d'illustrer les mécanismes utilisés afin de décrire les contraintes. Dans le cadre des formalismes qualitatifs, les contraintes sont définies implicitement par l'inverse et la composition faible. Nous avons imposé la manipulation de RCQ normalisés, il reste à définir la composition faible. Pour cela, nous utilisons la table de composition

TC du formalisme qualitatif. Nous rappelons que, pour tout $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ, \mathcal{N} est fermé par composition faible si et seulement si pour tout $r_{ij} \in Q_{(v_i, v_j)}$, il existe $r_{ik} \in Q_{(v_i, v_k)}$ et $r_{kj} \in Q_{(v_k, v_j)}$, tels que $(r_{ij}, r_{ik}, r_{kj}) \in \text{TC}$, pour tout $v_i, v_j, v_k \in \mathcal{V}$, avec $1 \leq i, j, k \leq n$.

5.3.1 Utilisation des conflits

Dans un premier temps, donnons l'idée d'une transformation de la table de composition à l'aide des conflits. Pour cela, posons $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ ainsi que les variables $v_i, v_j, v_k \in \mathcal{V}$, avec $1 \leq i, j, k \leq n$, et supposons que $(r_1, r_2, r_3) \in \mathcal{B}^3$ soient en conflits. $r_1, r_2, r_3 \in \mathcal{B}$ sont en conflits si et seulement si $(r_1, r_2, r_3) \notin \text{TC}$. Nous nous focalisons uniquement sur les contraintes entre les variables v_i, v_j et v_k dans un premier temps. Quel que soit un scénario $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ avec $\mathcal{N}' \subseteq \mathcal{N}$, \mathcal{N}' n'est pas \diamond -cohérent sur v_i, v_j, v_k si et seulement si $r_1 \in Q'_{(v_i, v_j)}$, $r_2 \in Q'_{(v_i, v_k)}$ et $r_3 \in Q'_{(v_k, v_j)}$. Considérons μ un modèle de $\Sigma_{\mathcal{N}}^{ALO} \wedge \Sigma_{\mathcal{N}}^{AMO}$, et supposons que \mathcal{N}' soit issu de μ par la proposition 3. \mathcal{N}' n'est pas \diamond -cohérence sur v_i, v_j, v_k si et seulement si $[r_1 \in Q_{(v_i, v_j)}] \in \mu$, $[r_2 \in Q_{(v_i, v_k)}] \in \mu$ et $[r_3 \in Q_{(v_k, v_j)}] \in \mu$. Nous déduisons que \mathcal{N}' vérifie la \diamond -cohérence sur v_i, v_j, v_k si et seulement si $\neg[r_1 \in Q_{(v_i, v_j)}] \in \mu$, $\neg[r_2 \in Q_{(v_i, v_k)}] \in \mu$ ou $\neg[r_3 \in Q_{(v_k, v_j)}] \in \mu$. Par conséquent, \mathcal{N}' vérifie la \diamond -cohérence sur v_i, v_j, v_k si et seulement si μ est un modèle de $\neg[r_1 \in Q_{(v_i, v_j)}] \vee \neg[r_2 \in Q_{(v_i, v_k)}] \vee \neg[r_3 \in Q_{(v_k, v_j)}]$. En imposant cette contrainte pour tout $v_i, v_j, v_k \in \mathcal{V}$ avec $1 \leq i, j, k \leq n$, nous pouvons déduire un scénario vérifiant la \diamond -cohérence pour tout triplet de variables.

Définition 80 (Formule $\Sigma_{\mathcal{N}}^{\text{Conflict}}$) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ. La formule « Conflict » associée à \mathcal{N} est définie par :

$$\Sigma_{\mathcal{N}}^{\text{Conflict}} = \bigwedge_{\substack{v_i, v_j, v_k \in \mathcal{V} : 1 \leq i < k < j \leq n, \\ r_1 \in Q_{(v_i, v_j)}, r_2 \in Q_{(v_i, v_k)}, r_3 \in Q_{(v_k, v_j)} : \\ (r_1, r_2, r_3) \notin \text{TC}}} \neg[r_1 \in Q_{(v_i, v_j)}] \vee \neg[r_2 \in Q_{(v_i, v_k)}] \vee \neg[r_3 \in Q_{(v_k, v_j)}]$$

Nous pouvons ainsi définir la transformation d'un RCQ vers une formule propositionnelle, en prenant en compte les clauses *ALO*, *AMO* et *Conflicts*.

Définition 81 (Transformation directe) Soit \mathcal{N} un RCQ. La transformation directe associée à \mathcal{N} est définie par :

$$\Sigma_{\mathcal{N}}^{\text{direct}} = \Sigma_{\mathcal{N}}^{\text{ALO}} \wedge \Sigma_{\mathcal{N}}^{\text{AMO}} \wedge \Sigma_{\mathcal{N}}^{\text{Conflict}}$$

La définition de la transformation *directe* nous permet alors de vérifier certaines propriétés liées à la cohérence du RCQ et de la formule associée. Dans un premier temps, nous pouvons affirmer que la transformation est *correcte*.

Proposition 4 Soit \mathcal{N} un RCQ. Si \mathcal{N} est cohérent, alors $\Sigma_{\mathcal{N}}^{\text{direct}}$ est satisfiable.

Preuve Pour montrer cette proposition, posons $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ. Supposons qu'il existe un scénario cohérent $\mathcal{N}' \subseteq \mathcal{N}$ tel que $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$.

À partir du scénario \mathcal{N}' , posons une interprétation complète μ de $\Sigma_{\mathcal{N}}^{\text{direct}}$ telle que si $r \in Q'_{(v_i, v_j)}$, alors $[r \in Q_{(v_i, v_j)}] \in \mu$, sinon $\neg[r \in Q_{(v_i, v_j)}] \in \mu$. Puisque \mathcal{N}' est cohérent, alors il est fermé par composition faible. Nous déduisons donc, pour tout $v_i, v_j, v_k \in \mathcal{V}$ tels que $1 \leq i, j, k \leq n$, qu'il existe $r_1, r_2, r_3 \in \mathcal{B}$ tels que $Q'_{(v_i, v_j)} = \{r_1\}$, $Q'_{(v_i, v_k)} = \{r_2\}$ et $Q'_{(v_k, v_j)} = \{r_3\}$, avec $r_1 \in r_2 \diamond r_3$, c'est-à-dire que $(r_1, r_2, r_3) \in \text{TC}$. Pour tout $r_1, r_2, r_3 \in \mathcal{B}$ tels que $(r_1, r_2, r_3) \notin \text{TC}$: $r_1 \notin Q'_{(v_i, v_j)}$ ou $r_2 \notin Q'_{(v_i, v_k)}$ ou $r_3 \notin Q'_{(v_k, v_j)}$. Or, puisque $\mathcal{N}' \subseteq \mathcal{N}$, nous savons que $Q'_{(v_i, v_j)} \subseteq Q_{(v_i, v_j)}$ pour tout $v_i, v_j \in \mathcal{V}$ avec

$1 \leq i, j, k \leq n$. Nous pouvons donc garantir que $\neg[r_1 \in Q_{(v_i, v_j)}] \in \mu$ ou $\neg[r_2 \in Q_{(v_i, v_k)}] \in \mu$ ou $\neg[r_3 \in Q_{(v_k, v_j)}] \in \mu$. μ satisfait la clause $\neg[r_1 \in Q_{(v_i, v_j)}] \vee \neg[r_2 \in Q_{(v_i, v_k)}] \vee \neg[r_3 \in Q_{(v_k, v_j)}]$ pour tout $r_1, r_2, r_3 \in \mathcal{B}$ tels que $(r_1, r_2, r_3) \notin \text{TC}$. Le modèle μ satisfait donc la formule $\Sigma_{\mathcal{N}}^{\text{Conflit}}$. Par construction, \mathcal{N}' est un scénario de \mathcal{N} , donc μ est un modèle de $\Sigma_{\mathcal{N}}^{\text{ALO}} \wedge \Sigma_{\mathcal{N}}^{\text{AMO}}$. Par conséquent, μ satisfait $\Sigma_{\mathcal{N}'}^{\text{direct}}$. Si \mathcal{N} est satisfiable, alors $\Sigma_{\mathcal{N}'}^{\text{direct}}$ est satisfiable. \square

Nous ne pouvons pas affirmer que la transformation *directe* soit complète. $\Sigma_{\mathcal{N}'}^{\text{direct}} \models \Sigma_{\mathcal{N}}^{\text{ALO}} \wedge \Sigma_{\mathcal{N}}^{\text{AMO}}$ donc tout modèle de $\Sigma_{\mathcal{N}'}^{\text{direct}}$ peut être associé à un scénario par la proposition 3. Cependant, tout modèle de $\Sigma_{\mathcal{N}'}^{\text{direct}}$ n'est pas nécessairement associé à un scénario cohérent. La formule $\Sigma_{\mathcal{N}}^{\text{Conflit}}$ modélise la propriété de composition faible, qui n'est pas suffisante pour décider de la cohérence d'un scénario dans le cas général.

Exemple 15 *Considérons pour exemple l'algèbre des intervalles cycliques, et particulièrement le RCQ représenté Figure 2.5 présentant un scénario \mathcal{N} \diamond -fermé mais incohérent. Puisque \mathcal{N}' est \diamond -fermé, $\Sigma_{\mathcal{N}'}^{\text{Conflit}}$ ne contient aucune clause, car il n'existe pas $v_i, v_j, v_k \in \mathcal{V}$, tels que $Q_{(v_i, v_j)} = \{r_1\}$, $Q_{(v_i, v_k)} = \{r_2\}$, $Q_{(v_k, v_j)} = \{r_3\}$ avec $(r_1, r_2, r_3) \notin \text{TC}$.*

$\Sigma_{\mathcal{N}'}^{\text{direct}} = [ppi \in Q_{(A, X)}] \wedge [ppi \in Q_{(B, X)}] \wedge [ppi \in Q_{(C, X)}] \wedge [m \in Q_{(A, B)}] \wedge [m \in Q_{(B, C)}] \wedge [mi \in Q_{(A, C)}]$. $\Sigma_{\mathcal{N}'}^{\text{direct}}$ admet bien un modèle et, par conséquent, cette formule est satisfiable. Pourtant, le scénario est incohérent.

Proposition 5 *Soit \mathcal{N} un RCQ. Si $\Sigma_{\mathcal{N}'}^{\text{direct}}$ est satisfiable, alors il existe un scénario $\mathcal{N}' \subseteq \mathcal{N}$ \diamond -cohérent.*

Preuve *Considérons $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ et $\Sigma_{\mathcal{N}'}^{\text{direct}}$ la formule propositionnelle obtenue par transformation directe de \mathcal{N} . Supposons que $\Sigma_{\mathcal{N}'}^{\text{direct}}$ soit satisfiable et qu'elle admet μ pour modèle.*

Tout d'abord, puisque μ est un modèle de $\Sigma_{\mathcal{N}'}^{\text{direct}}$, alors μ est également modèle de $\Sigma_{\mathcal{N}}^{\text{ALO}} \wedge \Sigma_{\mathcal{N}}^{\text{AMO}}$. Par la proposition 3, nous pouvons décrire un scénario normalisé $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ associé à μ tel que $\mathcal{N}' \subseteq \mathcal{N}$ et, pour tout $r \in \mathcal{B}$ et $v_i, v_j \in \mathcal{V}$ avec $i < j$: $Q'_{(v_i, v_j)} = \{r\}$ si et seulement si $\mu([r \in Q_{(v_i, v_j)}]) = 1$.

Vérifions maintenant que \mathcal{N}' est \diamond -cohérent. μ satisfait $\Sigma_{\mathcal{N}'}^{\text{Conflit}}$, par conséquent, pour tout $v_i, v_j, v_k \in \mathcal{V}$ avec $1 \leq i < k < j \leq n$, $r_1 \in Q_{(v_i, v_j)}$, $r_2 \in Q_{(v_i, v_k)}$, $r_3 \in Q_{(v_k, v_j)}$ tel que $(r_1, r_2, r_3) \notin \text{TC}$, nous savons que μ satisfait toute clause $\neg[r_1 \in Q_{(v_i, v_j)}] \vee \neg[r_2 \in Q_{(v_i, v_k)}] \vee \neg[r_3 \in Q_{(v_k, v_j)}]$. Dans le scénario associé à μ , si $(r_1, r_2, r_3) \notin \text{TC}$ alors $r_1 \notin Q'_{(v_i, v_j)}$ ou $r_2 \notin Q'_{(v_i, v_k)}$ ou $r_3 \notin Q'_{(v_k, v_j)}$. Par conséquent, quels que soient $Q'_{(v_i, v_j)} = \{r_1\}$, $Q'_{(v_i, v_k)} = \{r_2\}$ et $Q'_{(v_k, v_j)} = \{r_3\}$, nous pouvons affirmer que $(r_1, r_2, r_3) \in \text{TC}$. Nous avons donc construit un scénario \mathcal{N}' , tel que $Q'_{(v_i, v_j)} \subseteq Q'_{(v_i, v_k)} \diamond Q'_{(v_k, v_j)}$ pour tout $v_i, v_j, v_k \in \mathcal{V}$ avec $1 \leq i < k < j \leq n$. Par la proposition 1, nous pouvons étendre ce résultat à $Q'_{(v_i, v_j)} \subseteq Q'_{(v_i, v_k)} \diamond Q'_{(v_k, v_j)}$ pour tout $v_i, v_j, v_k \in \mathcal{V}$ avec $1 \leq i, j, k \leq n$. \mathcal{N}' est donc \diamond -cohérent. \square

Corollaire 2 *Soit \mathcal{N} un RCQ construit sur un formalisme qualitatif \mathcal{F} . Si la \diamond -cohérence est complète pour décider de la cohérence d'un scénario dans \mathcal{F} , alors $\Sigma_{\mathcal{N}'}^{\text{direct}}$ est satisfiable si et seulement si \mathcal{N} est cohérent.*

La preuve est immédiate. S'il existe un scénario \diamond -cohérent de \mathcal{N} , il est également cohérent, et \mathcal{N} est satisfiable. Dans le cas des formalismes qualitatifs pour lesquels la \diamond -cohérence est complète pour décider de la cohérence d'un scénario, la transformation *directe* est correcte et complète pour décider de la cohérence d'un RCQ. Décider de la satisfiabilité de la formule propositionnelle permet donc de décider de la satisfiabilité du réseau de contraintes qualitatives.

La formule $\Sigma_{\mathcal{N}'}^{\text{Conflit}}$ est composée de $\binom{n}{3} |\mathcal{B}|^3$ clauses ternaires. La complexité algorithmique en temps (resp. en espace) dans le pire des cas pour construire la formule propositionnelle $\Sigma_{\mathcal{N}'}^{\text{Conflit}}$ est donc $\mathcal{O}(n^3 |\mathcal{B}|^3)$ (resp. $\mathcal{O}(1)$). La transformation $\Sigma_{\mathcal{N}'}^{\text{direct}}$ est donc polynomiale en temps dans la taille du RCQ.

5.3.2 Utilisation des supports

Dans le même cadre d'étude, définissons les contraintes à partir des tuples supports, et démontrons les mêmes propriétés. Pour cela, nous utilisons également la table de composition telle que $(r_1, r_2, r_3) \in \text{TC}$ si et seulement si $r_1 \in r_2 \diamond r_3$. Posons $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un scénario et les variables $v_i, v_j, v_k \in \mathcal{V}$, avec $1 \leq i, j, k \leq n$. \mathcal{N} est \diamond -cohérent sur v_i, v_j, v_k si et seulement si il existe un $r_1 \in Q_{(v_i, v_j)}$, $r_2 \in Q_{(v_i, v_k)}$ et $r_3 \in Q_{(v_k, v_j)}$ telle que $(r_1, r_2, r_3) \in \text{TC}$. Comme énoncé dans le chapitre 4, de nombreuses formes de cohérence locale peuvent être modélisée. Nous faisons le choix de composition faible afin de décrire cette contrainte par : si $Q_{(v_i, v_k)} = \{r_2\}$ et $Q_{(v_k, v_j)} = \{r_3\}$, alors $\exists r_1 \in Q_{(v_i, v_j)} : r_1 \in r_2 \diamond r_3$.

Définition 82 (Formule $\Sigma_{\mathcal{N}}^{\text{Support}}$) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ. La formule « Support » associée à \mathcal{N} est décrite par :

$$\Sigma_{\mathcal{N}}^{\text{Support}} = \bigwedge_{\substack{v_i, v_j, v_k \in \mathcal{V} : 1 \leq i < k < j \leq n, \\ r_2 \in Q_{(v_i, v_k)}, r_3 \in Q_{(v_k, v_j)}} \left(\neg[r_2 \in Q_{(v_i, v_k)}] \vee \neg[r_3 \in Q_{(v_k, v_j)}] \vee \bigvee_{\substack{r_1 \in Q_{(v_i, v_j)} : \\ (r_1, r_2, r_3) \in \text{TC}}} [r_1 \in Q_{(v_i, v_j)}] \right)$$

La formule *Supports* permet de compléter les formules *ALO* et *AMO* et de définir la transformation par les supports.

Définition 83 (Formule $\Sigma_{\mathcal{N}}^{\diamond}$) Soit \mathcal{N} un RCQ. La transformation *support* est définie par :

$$\Sigma_{\mathcal{N}}^{\diamond} = \Sigma_{\mathcal{N}}^{\text{ALO}} \wedge \Sigma_{\mathcal{N}}^{\text{AMO}} \wedge \Sigma_{\mathcal{N}}^{\text{Support}}$$

Comme pour la transformation utilisant les conflits, T_{SATS} est correcte mais n'est pas complète dans le cas général.

Proposition 6 Soit \mathcal{N} un RCQ. Si \mathcal{N} est satisfiable, alors $\Sigma_{\mathcal{N}}^{\diamond}$ est satisfiable.

Preuve Soient $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ et $\mathcal{N}' \subseteq \mathcal{N}$ un scénario cohérent de \mathcal{N} tel que $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$. Posons μ une interprétation de $\Sigma_{\mathcal{N}}^{\diamond}$, telle que pour tout $r \in \mathcal{B}$, si $r \in Q_{(v_i, v_j)}$ alors $[r \in Q_{(v_i, v_j)}] \in \mu$, sinon $[r \in Q_{(v_i, v_j)}] \notin \mu$.

Puisque le scénario \mathcal{N}' est cohérent, alors il est \diamond -cohérent. Nous déduisons donc, pour tout $v_i, v_j, v_k \in \mathcal{V}$ tels que $1 \leq i, j, k \leq n$, qu'il existe $r_1, r_2, r_3 \in \mathcal{B}$ tels que $[r_1 \in Q_{(v_i, v_j)}], [r_2 \in Q_{(v_i, v_k)}]$ et $[r_3 \in Q_{(v_k, v_j)}]$ et $r_1 \in r_2 \diamond r_3$. Considérons la formule $\Sigma_{\mathcal{N}}^{\text{Conflict}}$. Pour toutes relations atomiques $r \in Q_{(v_i, v_k)}$ et $r' \in Q_{(v_k, v_j)}$:

- si $r \neq r_2$ ou $r \neq r_3$, alors toute clause $\neg[r \in Q_{(v_i, v_k)}] \vee \neg[r' \in Q_{(v_k, v_j)}] \vee \bigvee_{r'' \in Q_{(v_i, v_j)} \cap (r_2 \diamond r_3)} [r'' \in Q_{(v_i, v_j)}]$ est satisfaite par μ puisque $\neg[r \in Q_{(v_i, v_k)}] \in \mu$ ou $\neg[r' \in Q_{(v_k, v_j)}] \in \mu$;
- si $r = r_2$ et $r = r_3$, alors toute clause $\neg[r \in Q_{(v_i, v_k)}] \vee \neg[r' \in Q_{(v_k, v_j)}] \vee \bigvee_{r'' \in Q_{(v_i, v_j)} \cap (r_2 \diamond r_3)} [r'' \in Q_{(v_i, v_j)}]$ est satisfaite par μ si et seulement si il existe $r'' \in Q_{(v_i, v_j)}$ tel que $r'' \in Q_{(v_i, v_j)} \cap (r_2 \diamond r_3)$. Or, $[r_1 \in Q_{(v_i, v_j)}] \in \mu$.

Par conséquent, μ est un modèle de $\Sigma_{\mathcal{N}}^{\text{Conflict}}$. De plus, trivialement, puisque $\mathcal{N}' \subseteq \mathcal{N}$ et par construction de μ , la formule $\Sigma_{\mathcal{N}}^{\text{ALO}} \wedge \Sigma_{\mathcal{N}}^{\text{AMO}}$ est satisfaite par μ . La formule $\Sigma_{\mathcal{N}}^{\diamond}$ est donc satisfaite par μ . Par conséquent, si \mathcal{N} est satisfiable, alors $\Sigma_{\mathcal{N}}^{\diamond}$ est satisfiable. \square

Proposition 7 Soit \mathcal{N} un RCQ. Si $\Sigma_{\mathcal{N}}^{\diamond}$ est satisfiable, alors il existe un scénario $\mathcal{N}' \subseteq \mathcal{N}$ \diamond -cohérent.

Preuve Soient $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ et $\Sigma_{\mathcal{N}}^{\diamond}$ la formule propositionnelle associée à \mathcal{N} . Considérons que $\Sigma_{\mathcal{N}}^{\diamond}$ soit satisfiable et qu'elle admet μ pour modèle.

La proposition 3 permet de définir un RCQ atomique normalisé $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ associé à μ , tel que $\mathcal{N}' \subseteq \mathcal{N}$, et pour tout $r \in \mathcal{B}$, $v_i, v_j \in \mathcal{V}$ avec $i < j$, $Q'_{(v_i, v_j)} = \{r\}$ si et seulement si $[r \in Q_{(v_i, v_j)}] \in \mu$.

Vérifions maintenant que \mathcal{N}' soit \diamond -cohérent. Posons $v_i, v_j, v_k \in \mathcal{V}$ avec $1 \leq i < k < j \leq n$ et $Q'_{(v_i, v_j)} = \{r_1\}$, $Q'_{(v_i, v_k)} = \{r_2\}$, $Q'_{(v_k, v_j)} = \{r_3\}$. Par hypothèse μ satisfait $\Sigma_{\mathcal{N}}^{\diamond}$. Puisque $[r_2 \in Q_{(v_i, v_k)}] \in \mu$ et $[r_3 \in Q_{(v_k, v_j)}] \in \mu$, la clause $\neg[r_2 \in Q_{(v_i, v_k)}] \vee \neg[r_3 \in Q_{(v_k, v_j)}] \vee \bigvee_{r \in Q_{(v_i, v_j)} \cap (r_2 \diamond r_3)} [r \in Q_{(v_i, v_j)}]$ est satisfaite par μ si et seulement si $\bigvee_{r \in Q_{(v_i, v_j)} \cap (r_2 \diamond r_3)} [r \in Q_{(v_i, v_j)}]$ est satisfaite. C'est-à-dire satisfaite par μ si et seulement si il existe $r \in Q'_{(v_i, v_j)}$ tel que $r \in Q'_{(v_i, v_j)} \cap (r_2 \diamond r_3)$. Par conséquent, $r_1 \in r_2 \diamond r_3$ et $Q'_{(v_i, v_j)} \subseteq Q'_{(v_i, v_k)} \diamond Q'_{(v_k, v_j)}$ pour tout $v_i, v_j, v_k \in \mathcal{V}$ avec $1 \leq i < k < j \leq n$. La proposition 1 nous permet de généraliser ce résultat à tout $v_i, v_j, v_k \in \mathcal{V}$ avec $1 \leq i, j, k \leq n$. Nous pouvons conclure que si $\Sigma_{\mathcal{N}}^{\diamond}$ est satisfiable, alors \mathcal{N}' est \diamond -cohérent. \square

Afin de compléter la transformation *support*, nous pouvons également spécifier cette proposition à certains formalismes qualitatifs.

Corollaire 3 Soit \mathcal{N} un RCQ construit sur un formalisme qualitatif. Si la \diamond -cohérence est complète pour décider de la cohérence d'un scénario, alors $\Sigma_{\mathcal{N}}^{\diamond}$ est satisfiable si et seulement si \mathcal{N} est cohérent.

La transformation *support* est donc correcte et complète pour décider de la cohérence des RCQ sur les formalismes qualitatifs pour lesquels la \diamond -cohérence est complète pour décider de la cohérence d'un scénario.

La formule propositionnelle $\Sigma_{\mathcal{N}}^{\diamond}$ est composée de $\binom{n}{3} |\mathcal{B}|^2$ clauses de taille $2 + |\mathcal{B}|$ dans le pire des cas. La complexité algorithmique en temps (resp. en espace) dans le pire des cas, pour construire la formule $\Sigma_{\mathcal{N}}^{\text{Support}}$ est donc $\mathcal{O}(n^3 |\mathcal{B}|^2)$ (resp. $\mathcal{O}(|\mathcal{B}|)$). La transformation $\Sigma_{\mathcal{N}}^{\diamond}$ est donc polynomiale en temps dans la taille du RCQ.

Le tableau 5.1 illustre les dimensions des différentes formules CNF définies et la complexité algorithmique en temps et en espace pour les calculer dans le pire des cas. Remarquons la formule résultante est très imposante. Supposons un RCQ construit sur l'algèbre des intervalles, composé de 50 variables et de 1.225 contraintes *orientées*. La table de composition, contenant 169 tuples, est transformée pour 39.200 triplets de variables. Ainsi, plus de six millions de clauses modélisent la propriété de faible composition pour le RCQ.

Formule	Dimensions		Complexité algorithmique	
	Nombre de clauses	Taille des clauses	Temps	Espace
$\Sigma_{\mathcal{P}}^{\text{ALO}}$	$\binom{n}{2}$	$ \mathcal{B} $	$\mathcal{O}(n^2 \mathcal{B})$	$\mathcal{O}(\mathcal{B})$
$\Sigma_{\mathcal{P}}^{\text{AMO}}$	$\binom{n}{2} \binom{ \mathcal{B} }{2}$	2	$\mathcal{O}(n^2 \mathcal{B} ^2)$	$\mathcal{O}(1)$
$\Sigma_{\mathcal{P}}^{\text{Conflict}}$	$\binom{n}{3} \mathcal{B} ^3$	3	$n^3 \mathcal{B} ^3$	1
$\Sigma_{\mathcal{P}}^{\diamond}$	$\binom{n}{3} \mathcal{B} ^2$	$2 + \mathcal{B} $	$\mathcal{O}(n^3 \mathcal{B} ^2)$	$\mathcal{O}(\mathcal{B})$

TABLE 5.1 – Dimensions de la formule CNF et Complexité algorithmique dans le pire des cas

5.4 Conclusion

Dans ce chapitre, nous avons présenté deux transformations afin de modéliser un RCQ, quel que soit le formalisme qualitatif binaire utilisé, en formule CNF. Par une description naturelle de la microstructure du RCQ, nous nous appuyons sur les approches développées par Walsh et Kasif, afin de modéliser

les tuples interdits de la table de composition et en simulant la propriété de composition faible par la propagation unitaire. Les deux transformations sont correctes, mais ne sont pas complètes dans le cas général. Elles permettent de prouver l'existence d'un scénario \diamond -cohérent, ce qui n'est pas un critère suffisant pour décider de la cohérence. Cependant, pour de nombreux formalismes qualitatifs, tels que l'algèbre des intervalles ou le formalisme RCC8, les transformations sont complètes. Nous pouvons donc décider de la cohérence des RCQ en exploitant les outils algorithmiques développés pour le problème SAT.

La transformation de RCQ en logique propositionnelle a fait l'objet de récentes études. On peut citer la décomposition de formules CNF issues de RCQ pour améliorer la résolution [LI *et al.* 2009]. D'autres approches visent à réduire la taille imposante de la table de composition de certaines algèbres qualitatives [LI & RENZ 2010]. La transformation en logique propositionnelle a également été utilisée pour la fusion de RCQ [CONDOTTA *et al.* 2008]. Une étude comparative des algorithmes de résolution des RCQ a été menée dans [WESTPHAL & WÖLFL 2009]. Les auteurs montrent que la transformation en logique propositionnelle présente un intérêt lorsque le calcul de la \diamond -cohérence est difficile en RCQ.

Les perspectives à nos travaux sont évidemment d'étudier et d'adapter les techniques algorithmiques du cadre propositionnel, telles que l'analyse de conflits ou l'apprentissage. Au même titre que pour les transformations des RCD vers les formules propositionnelles, nous pouvons également exploiter des procédés de simplification afin de déduire de nouvelles transformations. De nombreuses approches peuvent également être envisagées pour répondre au problème de la cohérence des RCQ sans exhiber de scénario cohérent ou pour définir de nouvelles formes de cohérences locales. En particulier, des formes de cohérence locale pour les RCQ plus fortes que la \diamond -cohérence, qui sont difficiles à manipuler dans le cadre qualitatif. Nous souhaitons également diriger nos études vers la « réduction » de la formule propositionnelle obtenue. Pour un RCQ relativement petit, le nombre de clauses est conséquent, notamment à cause de la modélisation de la table de composition pour chaque triplet de variables. Or, ces clauses ont toujours la même structure, la même forme et la même sémantique, seules les variables impliquées changent. Nous ne pouvons affirmer, dans le cas général, que la taille de cette formule est un frein à la résolution, mais elle peut contribuer à rendre la tâche plus difficile. Ainsi, nous pensons qu'il peut être intéressant de se diriger vers la compilation de connaissances, ou encore vers le domaine émergent SMT (pour SAT *Modulo Theory*).

Chapitre 6

Du problème CSP qualitatif au problème SAT en exploitant les classes traitables

Sommaire

6.1	Introduction	93
6.2	Transformation	94
6.2.1	Exploiter le treillis conceptuel	94
6.2.2	Définition des variables propositionnelles	95
6.2.3	Des propriétés du treillis conceptuel aux formules CNF	96
6.2.4	Découpage et transformation d'un RCQ	97
6.2.5	Dimension et complexité	99
6.3	Propriétés et complétude	100
6.3.1	Transformation inverse	100
6.3.2	Transformation correcte et complète	101
6.4	Conclusion	103

6.1 Introduction

De nombreuses transformations vers le cadre propositionnel s'appuient sur un schéma commun : établir une analogie entre interprétation, scénario et instanciation, ainsi que modéliser certaines formes de cohérence locale pour décider de la cohérence d'un RCQ ou d'un RCD. L'approche que nous proposons dans ce chapitre ne se concentre pas uniquement sur un scénario, mais sur des RCQ définies sur des classes traitables. Nous proposons une transformation issue de la définition de la classe des relations convexes. Pour cela, nous exploitons la structure de treillis de certains formalismes qualitatifs, en modélisant les intervalles du treillis et représentant les propriétés de celui-ci à l'aide d'une formule CNF. En nous concentrant sur un cadre où la méthode de fermeture par composition faible est complète pour le problème de la cohérence des RCQ convexes, nous montrons que notre transformation est correcte et complète pour le problème de la cohérence des RCQ.

Dans ce chapitre, nous proposons tout d'abord un aperçu des propriétés du treillis conceptuel qui constitueront le cadre d'étude de notre transformation. À partir de ces propriétés, nous dirigeons notre étude vers la logique propositionnelle en définissant les variables propositionnelles et les propositions qu'elles décrivent, puis nous décrivons notre transformation par la représentation des propriétés du treillis conceptuel en tant que formules CNF. Nous nous intéressons ensuite aux propriétés de complétude de

la transformation. Dans un premier temps, nous montrons que la transformation proposée est correcte et complète pour le problème de la cohérence de RCQ convexes. Cette approche nous permet de généraliser ce résultat au cadre RCQ en général lorsque la méthode de fermeture par composition faible est complète pour le problème de la cohérence des RCQ convexes.

6.2 Transformation

6.2.1 Exploiter le treillis conceptuel

La transformation proposée permet la représentation d'un RCQ en une formule CNF dans le but de décider de la cohérence d'un RCQ. Pour cela, nous exploitons l'ordre partiel \preceq sur \mathcal{B} de l'algèbre considéré. Un pré-requis à cette transformation est l'existence de cet ordre partiel, pour lequel certaines propriétés doivent être vérifiées. Tout d'abord, le couple (\mathcal{B}, \preceq) doit être un treillis. Par conséquent, deux éléments $inf, sup \in \mathcal{B}$ doivent être, respectivement, la borne inférieure et la borne supérieure de \mathcal{B} , donc $inf \preceq r$ et $r \preceq sup$ pour tout $r \in \mathcal{B}$. (\mathcal{B}, \preceq) doit également vérifier, pour tout $r_1, r_2, r_3 \in \mathcal{B}$, les propriétés suivantes :

- la réflexivité : $r_1 \preceq r_1$;
- la transitivité : si $r_1 \preceq r_3$ et $r_3 \preceq r_2$ alors $r_1 \preceq r_2$;
- l'antisymétrie : si $r_1 \preceq r_2$ et $r_2 \preceq r_1$ alors $r_1 = r_2$.

Afin d'exploiter une telle structure, nous allons utiliser les intervalles de celle-ci. Pour tout $r_1, r_2 \in \mathcal{B}$, l'intervalle $[r_1 ; r_2]$ représente la relation de $2^{\mathcal{B}}$ définie par $[r_1 ; r_2] = \{r \in \mathcal{B} : r_1 \preceq r \wedge r \preceq r_2\}$. Pour de nombreux formalismes qualitatifs [NÖKEL 1991, NEBEL & BÜRCKERT 1995, LIGOZAT 1998a, BALBIANI *et al.* 1999, BALBIANI & CONDOTTA 2002] un tel treillis existe. Il est appelé treillis qualitatif ou conceptuel et l'ensemble des intervalles $\{[r_1 ; r_2] : r_1, r_2 \in \mathcal{B}\}$ n'est autre que la classe de relations convexes \mathcal{C} ne contenant pas \emptyset . $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ est un RCQ convexe si et seulement si $Q_{(v_i, v_j)} = [r^- ; r^+]$ pour tout $v_i, v_j \in \mathcal{V}$. Dans la suite, un RCQ convexe sera dénoté $RCQ_{\mathcal{C}}$ pour « RCQ Convex ».

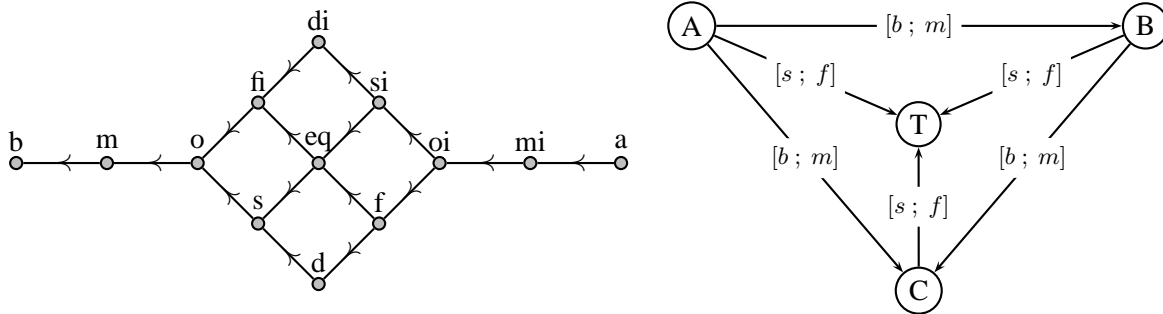


FIGURE 6.1 – Le treillis conceptuel (\mathcal{B}, \preceq) de AI (à gauche) et un $RCQ_{\mathcal{C}}$ cohérent

Nous rappelons le treillis conceptuel de l'algèbre des intervalles Figure 6.1 et nous présentons également un RCQ convexe. Ce $RCQ_{\mathcal{C}}$ est une simplification de l'exemple 2.3 du chapitre 2, dans lequel trois étudiants se partagent le temps de cours pour les corrections. Remarquons que ce $RCQ_{\mathcal{C}}$ est cohérent puisqu'il est \diamond -cohérent.

Dans la suite, nous admettons également deux nouvelles propriétés sur (\mathcal{B}, \preceq) concernant les opérations d'inverse et de composition faible. Celles-ci ont été démontrées par Ligozat [LIGOZAT 1991] en étudiant

une généralisation de l’algèbre des intervalles, et en définissant les propriétés des treillis quelle que soit la généralisation. Associées aux propriétés élémentaires du treillis, celles-ci constituent notre cadre d’étude pour la transformation d’un RCQ en formule CNF.

1. pour tout $r_1, r_2 \in \mathcal{B}$, si $r_1 \preceq r_2$ alors $r_2^{-1} \preceq r_1^{-1}$;
2. pour tout $r_1, r_2 \in \mathcal{B}$, $r_1 \diamond r_2$ est un intervalle de (\mathcal{B}, \preceq) ;
3. pour tout $r_1, r_2, r_3, r_4 \in \mathcal{B}$, $[r_1 ; r_2] \diamond [r_3 ; r_4] = [\text{Inf}(r_1 \diamond r_3) ; \text{Sup}(r_2 \diamond r_4)]$.

6.2.2 Définition des variables propositionnelles

Pour transformer un RCQ en formule propositionnelle en utilisant la classe des relations convexes, nous définissons une transformation T_C qui, à tout RCQ, associe une formule CNF. Pour cela, nous allons utiliser le même procédé que Nebel et Bürckert pour définir les relations ORD-Horn, c’est-à-dire que nous allons modéliser une théorie - les propriétés du treillis - en une formule CNF. Avant cela, il nous faut définir les propositions et variables propositionnelles.

Nous souhaitons modéliser des intervalles de relations atomiques à l’aide de l’ordre \preceq . Pour cela, nous allons décrire une relation atomique « minorante » et « majorante » afin de décrire la relation d’une contrainte qualitative. Pour $v_i, v_j \in \mathcal{V}$ et $r \in \mathcal{B}$:

- la variable propositionnelle $[Q_{(v_i, v_j)} \preceq r]$ modélise la proposition « Il existe une relation $r' \in Q_{(v_i, v_j)}$ telle que $r' \preceq r$ » ;
- une variable propositionnelle $[r \preceq Q_{(v_i, v_j)}]$ modélise la proposition « Il existe une relation $r' \in Q_{(v_i, v_j)}$ telle que $r \preceq r'$ ».

Posons $r \in \mathcal{B}$ et $v_i, v_j \in \mathcal{V}$. Par la propriété d’antisymétrie, s’il existe une relation $r' \in Q_{(v_i, v_j)}$ telle que $r' \preceq r$ et $r \preceq r'$, alors $r \in Q_{(v_i, v_j)}$.

La figure 6.2 illustre le treillis conceptuel de l’algèbre des intervalles pour une contrainte qualitative quelconque par la relation complexe $Q_{(v_i, v_j)} = [s ; si] = \{s, eq, si\}$. Sur celui-ci, les relations atomiques de l’ensemble $\{r \in \mathcal{B} : \exists r' \in Q_{(v_i, v_j)}, r' \preceq r\}$ sont encadrées en vert. Ainsi, il existe une relation $r' \in Q_{(v_i, v_j)}$ telle que pour tout $r \in [b ; si]$, nous ayons $r \preceq r'$. De la même façon, les relations atomiques de l’ensemble $\{r \in \mathcal{B} : \exists r' \in Q_{(v_i, v_j)}, r \preceq r'\}$ sont encadrées en rouge. Par conséquent, il existe une relation $r' \in Q_{(v_i, v_j)}$ telle que pour tout $r \in [s ; oi]$, nous ayons $r' \preceq r$. Par la propriété d’antisymétrie, l’ensemble $\{r \in \mathcal{B} : \exists r' \in Q_{(v_i, v_j)}, r' \preceq r \wedge r' \preceq r\}$ correspond à la relation complexe $[b ; si] \cap [s ; oi] = [s ; si]$ et coïncide avec $Q_{(v_i, v_j)}$.

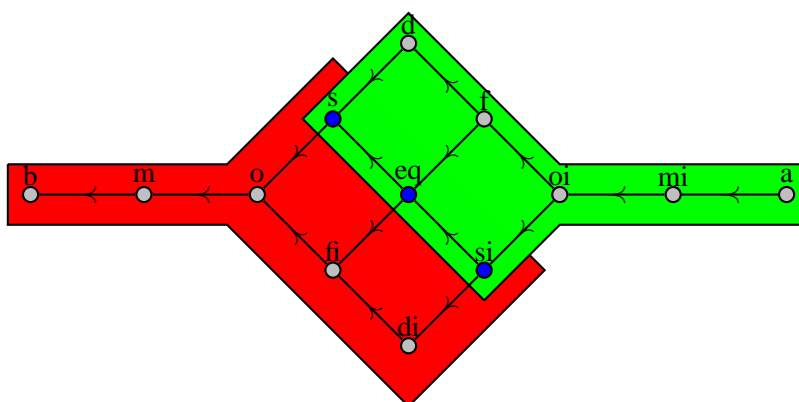


FIGURE 6.2 – Illustration du treillis conceptuel et des propositions de la transformation

Posons $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ et $T_{\mathcal{C}}(\mathcal{N})$ la formule associée par notre transformation. L'ensemble des variables propositionnelles de la formule éléments, est :

$$\text{vars}(T_{\mathcal{C}}(\mathcal{N})) = \{[r \preceq Q_{(v_i, v_j)}] : v_i, v_j \in \mathcal{V} \wedge r \in \mathcal{B}\} \cup \{[Q_{(v_i, v_j)} \preceq r] : v_i, v_j \in \mathcal{V} \wedge r \in \mathcal{B}\}$$

6.2.3 Des propriétés du treillis conceptuel aux formules CNF

Sur les variables propositionnelles, nous devons décrire les propriétés du treillis. Pour cela, nous définissons, dans un premier temps, deux applications Inf et Sup qui, à tout ensemble de relations atomiques, associe respectivement la borne inférieure et la borne supérieure de cet ensemble.

Définition 84 (Infimum et Supremum) Soit $R \subseteq \mathcal{B}$ et $I(R) = [r^- ; r^+]$ la fermeture convexe de R . L'infimum de R , noté $\text{Inf}(R)$, est la borne inférieure de l'intervalle $[r^- ; r^+]$ telle que $\text{Inf}(R) = r^-$. Le supremum de R , noté $\text{Sup}(R)$, est la borne supérieure de l'intervalle $[r^- ; r^+]$ telle que $\text{Sup}(R) = r^+$.

Pour illustrer la définition 84, nous pouvons prendre l'exemple de relations sur le treillis des intervalles : $\{fi, d\}$ et $\{o, eq, oi\}$. La fermeture convexe est $I(\{fi, d\}) = [o ; f]$. Donc $\text{Inf}(\{fi, d\}) = o$ et $\text{Sup}(\{fi, d\}) = f$. De même, $I(\{o, eq, oi\}) = [o ; oi]$. Par conséquent, $\text{Inf}(\{o, eq, oi\}) = o$ et $\text{Sup}(\{o, eq, oi\}) = oi$.

Dans la suite, nous considérons $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ normalisé construit sur un formalisme qualitatif de relations atomiques \mathcal{B} et de domaine \mathbb{D} .

Infimum et Supremum La définition de supremum permet d'affirmer, pour tout $r, r' \in \mathcal{B}$ et pour tout $v_i, v_j \in \mathcal{V}$, que $r \preceq Q_{(v_i, v_j)}$ et $r' \preceq Q_{(v_i, v_j)}$ permettent de déduire $\text{Sup}(\{r, r'\}) \preceq Q_{(v_i, v_j)}$. Par la définition d'infimum, nous avons également, si $Q_{(v_i, v_j)} \preceq r$ et $Q_{(v_i, v_j)} \preceq r'$, alors $Q_{(v_i, v_j)} \preceq \text{Inf}(\{r, r'\})$. Cette propriété est nécessaire puisque nous exploitons des relations convexes, il faut donc traiter uniquement des intervalles du treillis, et pour cela nous manipulons les bornes de la fermeture convexe des relations.

$$T_{\text{SAT}}^I(\mathcal{N}) = \bigwedge_{(v_i, v_j \in \mathcal{V})} \bigwedge_{(r, r' \in \mathcal{B}: r \neq r')} \neg[Q_{(v_i, v_j)} \preceq r] \vee \neg[Q_{(v_i, v_j)} \preceq r'] \vee [Q_{(v_i, v_j)} \preceq \text{Inf}(\{r, r'\})] \quad (\text{Infimum})$$

$$\bigwedge_{(v_i, v_j \in \mathcal{V})} \bigwedge_{(r, r' \in \mathcal{B}: r \neq r')} \neg[r \preceq Q_{(v_i, v_j)}] \vee \neg[r' \preceq Q_{(v_i, v_j)}] \vee [\text{Sup}(\{r, r'\}) \preceq Q_{(v_i, v_j)}] \quad (\text{Supremum})$$

Remarquons que, si les clauses d'infimum et de supremum sont calculées pour une relation convexe $[r, r']$, alors elles sont tautologiques. La clause d'infimum devient $\neg[Q \preceq r] \vee \neg[Q \preceq r'] \vee [Q \preceq r]$ et le supremum est $\neg[r \preceq Q] \vee \neg[r' \preceq Q] \vee [r' \preceq Q]$. Les seules relations atomiques concernées sont donc les relations atomiques « incomparables », c'est-à-dire $r \not\preceq r'$ et $r' \not\preceq r$ quels que soient $r, r' \in \mathcal{B}$.

Transitivité Nous pouvons également définir la propriété sur la transitivité des relations. Considérons $v_i, v_j \in \mathcal{V}$ et $r, r' \in \mathcal{B}$. Trivialement, si $r' \prec r$ et $r \preceq Q_{(v_i, v_j)}$, alors $r' \prec Q_{(v_i, v_j)}$. Ainsi, si $r \not\preceq r'$ et $r \preceq Q_{(v_i, v_j)}$ alors $Q_{(v_i, v_j)} \not\preceq r'$.

$$T_{\text{SAT}}^{II}(\mathcal{N}) = \bigwedge_{(v_i, v_j \in \mathcal{V})} \bigwedge_{(r, r' \in \mathcal{B}: r \not\preceq r')} \neg[r \preceq Q_{(v_i, v_j)}] \vee \neg[Q_{(v_i, v_j)} \preceq r'] \quad (\text{Transitivité})$$

Au regard de la définition des variables propositionnelles, la propriété de transitivité permet de garantir, pour tout $Q_{(v_i, v_j)} = [r^- ; r^+]$, que le supremum des minorants $r \in \mathcal{B}$, resp. l'infimum de majorants $r \in \mathcal{B}$, est tel que $r \preceq r^-$, resp. $r^+ \preceq r$.

Inverse S'ensuit la modélisation de la propriété d'inverse d'une relation. Considérons $r \in \mathcal{B}$ et $v_i, v_j \in \mathcal{V}$. Si $r \preceq Q_{(v_i, v_j)}$ alors $Q_{(v_i, v_j)}^{-1} \preceq r^{-1}$. De même, si $Q_{(v_i, v_j)} \preceq r$ alors $r^{-1} \preceq Q_{(v_i, v_j)}^{-1}$.

$$T_{\text{SAT}}^{\text{III}}(\mathcal{N}) = \wedge \left. \begin{array}{l} \bigwedge_{(v_i, v_j \in \mathcal{V})} \bigwedge_{(r \in \mathcal{B})} \neg[r \preceq Q_{(v_i, v_j)}] \vee [Q_{(v_i, v_j)}^{-1} \preceq r^{-1}] \\ \bigwedge_{(v_i, v_j \in \mathcal{V})} \bigwedge_{(r \in \mathcal{B})} \neg[Q_{(v_i, v_j)} \preceq r] \vee [r^{-1} \preceq Q_{(v_i, v_j)}^{-1}] \end{array} \right\} \text{(Inverse)}$$

Composition faible Enfin, nous modélisons également la propriété de composition faible sur les relations atomiques du treillis. Pour deux relations convexes $[r_1^- ; r_1^+], [r_2^- ; r_2^+] \in \mathcal{C}$, $[r_1^- ; r_1^+] \diamond [r_2^- ; r_2^+] = [\text{Inf}(r_1^- \diamond r_2^-) ; \text{Sup}(r_1^+ \diamond r_2^+)]$. Posons $r, r' \in \mathcal{B}$ et $v_i, v_j, v_k \in \mathcal{V}$. Nous en déduisons que si $r \preceq Q_{(v_i, v_k)}$ et $r' \preceq Q_{(v_k, v_j)}$ alors $\text{Inf}(r \diamond r') \preceq Q_{(v_i, v_j)}$. Mais également, si $Q_{(v_i, v_k)} \preceq r$ et $Q_{(v_k, v_j)} \preceq r'$ alors $Q_{(v_i, v_j)} \preceq \text{Sup}(r \diamond r')$.

$$T_{\text{SAT}}^{\text{IV}}(\mathcal{N}) = \wedge \left. \begin{array}{l} \bigwedge_{(v_i, v_j, v_k \in \mathcal{V})} \bigwedge_{(r, r' \in \mathcal{B})} \left(\neg[r \preceq Q_{(v_i, v_k)}] \vee \neg[r' \preceq Q_{(v_k, v_j)}] \vee [\text{Inf}(r \diamond r') \preceq Q_{(v_i, v_j)}] \right) \\ \bigwedge_{(v_i, v_j, v_k \in \mathcal{V})} \bigwedge_{(r, r' \in \mathcal{B})} \left(\neg[Q_{(v_i, v_k)} \preceq r] \vee \neg[Q_{(v_k, v_j)} \preceq r'] \vee [Q_{(v_i, v_j)} \preceq \text{Sup}(r \diamond r')] \right) \end{array} \right)$$

6.2.4 Découpage et transformation d'un RCQ

Nous pouvons remarquer qu'il est difficile de restreindre l'ensemble des variables propositionnelles dans le cas général. Contrairement à des transformations plus « classiques », s'attachant à modéliser des scénarios, il n'est pas possible de restreindre notre étude au cadre d'un graphe orienté acyclique, c'est-à-dire à tout $v_i, v_j \in \mathcal{V}$ avec $i < j$. Un modèle n'est pas associé à un scénario, donc la proposition 1 ne tient pas dans ce cadre. De plus, la description des variables propositionnelles démontre que nous ne pouvons pas restreindre les variables propositionnelles à la microstructure du RCQ, c'est-à-dire aux relations atomiques uniquement présentes dans les relations complexes des contraintes qualitatives. Néanmoins, nous concentrons notre étude sur les RCQ normalisés, et nous pouvons donc supposer, pour toute variable $v_i \in \mathcal{V}$, $Q_{(v_i, v_i)} = \{\text{id}\}$.

Pour définir notre transformation, nous supposons, dans un premier temps, que nous traitons des $\text{RCQ}_{\mathcal{C}}$. En d'autres termes, pour tout $v_i, v_j \in \mathcal{V}$, nous savons que $Q_{(v_i, v_j)} \in \mathcal{C}$, et donc $Q_{(v_i, v_j)} = [r^- ; r^+]$. Puisque tout modèle μ doit satisfaire $[r^- \preceq Q_{(v_i, v_j)}] \in \mu$ et $[Q_{(v_i, v_j)} \preceq r^+] \in \mu$, il nous faut imposer la contrainte « au moins une » *ALO*.

$$T_{\text{SAT}}^{\text{ALO}}(\mathcal{N}) = \bigwedge_{(v_i, v_j \in \mathcal{V}: Q_{(v_i, v_j)} = [r^- ; r^+])} [r^- \preceq Q_{(v_i, v_j)}] \wedge [Q_{(v_i, v_j)} \preceq r^+]$$

Définition 85 ($T_{\mathcal{C}}(\mathcal{N})$) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un $\text{RCQ}_{\mathcal{C}}$ construit sur un formalisme qualitatif de relations atomiques \mathcal{B} .

$$T_{\mathcal{C}}(\mathcal{N}) = T_{\text{SAT}}^{\text{ALO}}(\mathcal{N}) \wedge T_{\text{SAT}}^{\text{I}}(\mathcal{N}) \wedge T_{\text{SAT}}^{\text{II}}(\mathcal{N}) \wedge T_{\text{SAT}}^{\text{III}}(\mathcal{N}) \wedge T_{\text{SAT}}^{\text{IV}}(\mathcal{N})$$

En nous focalisant sur le cas de la transformation d'un $\text{RCQ}_{\mathcal{C}}$ \mathcal{N} , nous pouvons remarquer que la définition de chaque formule : $T_{\text{SAT}}^{\text{ALO}}(\mathcal{N})$, $T_{\text{SAT}}^{\text{I}}(\mathcal{N})$, $T_{\text{SAT}}^{\text{II}}(\mathcal{N})$, $T_{\text{SAT}}^{\text{III}}(\mathcal{N})$, $T_{\text{SAT}}^{\text{IV}}(\mathcal{N})$ sont uniquement

des clauses de Horn. Tout $\text{RCQ}_C \mathcal{N}$ peut donc être transformé en une formule de Horn $\text{T}_C(\mathcal{N})$. Il est donc possible de décider de la satisfiabilité de $\text{T}_C(\mathcal{N})$ en temps polynomial lorsque \mathcal{N} , nous verrons ensuite la complexité algorithmique pour construire cette formule.

L'objectif est, bien évidemment, de pouvoir utiliser la transformation sur tout RCQ. Les propriétés se focalisent sur le treillis conceptuel du formalisme qualitatif utilisé. Par conséquent, les formules CNF obtenues par T_C pour des RCQ distincts construits sur un même formalisme qualitatif ne se différencient que par les clauses $\text{T}_{\text{SAT}}^{\text{ALO}}$. Pour généraliser la transformation, nous pouvons remarquer que toute contrainte qualitative, décrivant une relation $Q_{(v_i, v_j)} \in 2^{\mathcal{B}}$, peut toujours être découpée en une partition de k relations $\{Q_{(v_i, v_j)}^1, \dots, Q_{(v_i, v_j)}^k\}$ avec $k \leq |\mathcal{B}|$, telle que $Q_{(v_i, v_j)} = \bigcup_{l \in [1, \dots, k]} Q_{(v_i, v_j)}^l$ et $Q_{(v_i, v_j)}^l$ est une sous-relation convexe, pour tout $l \in [1 ; k]$. Ainsi, $Q_{(v_i, v_j)} = [r_1^- ; r_1^+] \cup \dots \cup [r_k^- ; r_k^+]$ avec $r_l^-, r_l^+ \in \mathcal{B}$ pour tout $l \in [1 ; k]$.

La figure 6.3 illustre le découpage de la relation complexe $\{b, m, d, eq, f, si, oi, mi\}$ en sous-relations convexes. Notre choix de découpage se porte sur $[b ; m]$, $[d ; d]$ et $[eq ; mi]$. Notons, dans un premier temps, que le découpage n'est pas unique. Notre choix aurait pu se porter sur $[b ; m]$, $[d ; mi]$ et $[eq ; si]$. De plus, toute relation $\{r\} \in 2^{\mathcal{B}}$ peut être décrite par une relation convexe $[r ; r]$. Dans le pire des cas, une relation $R \in 2^{\mathcal{B}}$ peut donc être découpée en $|R|$ relations convexes. Dans le cas des relations de l'algèbre d'Allen, le nombre de relations convexes nécessaires pour un découpage minimal est de 3.55 en moyenne [NEBEL 1997]. Nous supposons, dans la suite, que le découpage est effectué par critères afin d'obtenir un découpage unique pour chaque relation de $2^{\mathcal{B}}$.

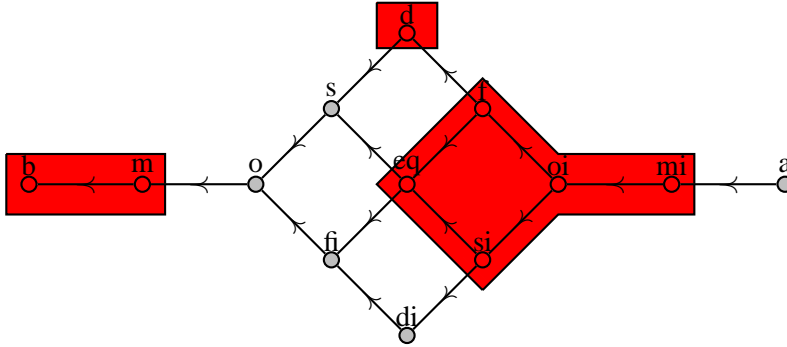


FIGURE 6.3 – Découpage de la relation $\{b, m, d, eq, f, si, oi, mi\}$ de l'algèbre des intervalles en sous-relations convexes

Pour tout $v_i, v_j \in \mathcal{V}$, la modélisation de $Q_{(v_i, v_j)}$ sera une formule DNF qu'il sera nécessaire de transformer en formule CNF. La conjonction de ces formules pour chaque $v_i, v_j \in \mathcal{V}$ décrit la transformation $\text{T}_{\text{SAT}}^{\text{ALO}}$ dans le cas d'un RCQ.

$$\text{T}_{\text{SAT}}^{\text{ALO}}(\mathcal{N}) = \bigwedge_{v_i, v_j \in \mathcal{V} : \left(Q_{(v_i, v_j)} = [r_1^- ; r_1^+] \cup \dots \cup [r_k^- ; r_k^+] \right)} \left(\begin{array}{l} \vee [r_1^- \preceq Q_{(v_i, v_j)}] \wedge [Q_{(v_i, v_j)} \preceq r_1^+] \\ \dots \\ \vee [r_k^- \preceq Q_{(v_i, v_j)}] \wedge [Q_{(v_i, v_j)} \preceq r_k^+] \end{array} \right)$$

La formule $\text{T}_{\text{SAT}}^{\text{ALO}}(\mathcal{N})$ n'est pas une formule de Horn dans le cas général.

Exemple 16 Pour illustrer le découpage et la transformation $\text{T}_{\text{SAT}}^{\text{ALO}}$, reprenons l'exemple 10 du chapitre 2 modélisé en utilisant l'algèbre des intervalles. Pour rappel, pendant le temps d'un cours T , trois corrections A, B et C doivent être réalisées séparément. Nous ajoutons une contrainte triviale lors de la modélisation : A, B et C ne peuvent occuper l'intégralité du temps du cours T . Ainsi, les contraintes entre

corrections sont représentées par $Q_{(A,B)} = Q_{(A,C)} = Q_{(B,C)} = \{b, m, mi, a\} = [b ; m] \cup [mi ; a]$, et les contraintes liées au temps de cours sont modélisées par $Q_{(A,T)} = Q_{(B,T)} = Q_{(C,T)} = \{s, d, f\} = [s ; d] \cup [f ; f]$. La figure 6.4 illustre le découpage en relations convexes appliqué.

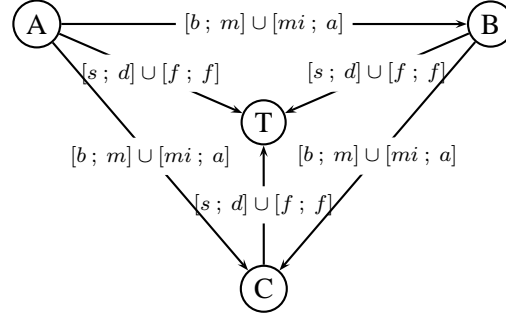


FIGURE 6.4 – Illustration du découpage des relations sur le problème de l'exemple 10

Considérons la contrainte $Q_{(A,B)} = \{b, m, mi, a\} = [b, m] \cup [mi, a]$. Par la transformation T_{SAT}^{ALO} , nous obtenons la formule DNF :

$$\vee \begin{array}{l} [b \preceq Q_{(A,B)}] \wedge [Q_{(A,B)} \preceq m] \\ [mi \preceq Q_{(A,B)}] \wedge [Q_{(A,B)} \preceq a] \end{array}$$

Par la propriété de distributivité, nous obtenons la formule CNF suivante :

$$\wedge \begin{array}{l} [b \preceq Q_{(A,B)}] \vee [mi \preceq Q_{(A,B)}] \\ [b \preceq Q_{(A,B)}] \vee [Q_{(A,B)} \preceq a] \\ [Q_{(A,B)} \preceq m] \vee [b \preceq Q_{(A,B)}] \\ [Q_{(A,B)} \preceq m] \vee [Q_{(A,B)} \preceq a] \end{array}$$

Le procédé est similaire pour $Q_{(A,C)}$, $Q_{(B,C)}$, $Q_{(A,T)}$, $Q_{(B,T)}$, $Q_{(C,T)}$.

6.2.5 Dimension et complexité

Considérons un RCQ $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$, avec $|\mathcal{V}| = n$, construit sur un formalisme qualitatif de relations atomiques \mathcal{B} . Chaque proposition porte sur chaque relation atomique de \mathcal{B} et sur tout couple de variables $v_i, v_j \in \mathcal{V}$ tel que $i \neq j$. Par conséquent, $2n(n-1)|\mathcal{B}|$ variables propositionnelles sont modélisées.

Les formules associées aux propriétés du treillis formalisme qualitatif sont invariantes quel que soit le RCQ. Les propriétés d'*infimum* et de *supremum* sont appliquées à chaque contrainte qualitative, hormis les contraintes d'identité, et sur tout couple distinct de relations atomiques dans le pire des cas. $2n(n-1)\binom{|\mathcal{B}|}{2}$ clauses de taille 3 modélisent ces propriétés. Dans le pire des cas, la complexité algorithmique en temps (resp. en espace) dans le pire des cas pour les calculer est donc $\mathcal{O}(n^2|\mathcal{B}|^2)$ (resp. $\mathcal{O}(1)$). La propriété de *transitivité* est appliquée pour chaque contrainte qualitative et sur tout couple de relations atomiques $r, r' \in \mathcal{B}$ tel que $r \not\preceq r'$. Ce qui correspond à $n(n-1)\binom{|\mathcal{B}|}{2}$ clauses de taille 2, calculées en $\mathcal{O}(n^2|\mathcal{B}|^2)$ en temps et en $\mathcal{O}(1)$ en espace. La propriété d'*inverse* des relations s'applique également pour toute contrainte qualitative sur relation atomique, et impose donc la création de $2n(n-1)|\mathcal{B}|$ clauses de taille 2, qui sont calculées en temps (resp. en espace) en $\mathcal{O}(n^2|\mathcal{B}|)$ et $\mathcal{O}(1)$ respectivement. La modélisation des formules de composition faible est définie sur tout triplet de contraintes qualitatives, et sur tout couple de relations atomiques. Le nombre de clauses résultant est donc $2n(n-1)(n-2)|\mathcal{B}|^2$ de taille 3. La complexité algorithmique en temps (resp. en espace) dans le pire des cas pour réaliser cette opération est donc $\mathcal{O}(n^3)|\mathcal{B}|^2$ (resp. $\mathcal{O}(1)$).

Le cas des clauses *ALO* est particulier puisqu'il dépend avant tout du découpage effectué. En posant k le nombre maximal de relations convexes utilisées pour découper une relation complexe, et puisque le découpage est effectué sur chaque contrainte qualitative, la formule *ALO* contient $2^k n(n-1)$ clauses de taille k . Notons que le découpage des relations de $2^{\mathcal{B}}$ est réalisé *a priori*, afin de garantir un découpage optimal. Pour des formalismes qualitatifs, tels que l'algèbre des intervalles [NEBEL 1997] ou RCC – 8 [RENTZ & NEBEL 2001], le découpage moyen est faible par rapport à $|\mathcal{B}|$. Dans le cas de l'algèbre des intervalles, le nombre de clauses de T_{SAT}^{ALO} peut être estimé à $2^{3.55} n(n-1)$, soit approximativement $|\mathcal{B}|n(n-1)$.

Le tableau 6.1 montre un récapitulatif des dimensions de la formule propositionnelle et la complexité algorithmique dans le pire des cas pour calculer T_C . La transformation T_C est donc polynomiale en temps en fonction du nombre de variables, de relations atomiques si et seulement si 2^k est de l'ordre d'une fonction polynomiale, ce qui est le cas pour la plupart des algèbres qualitatives sur lesquels un treillis conceptuel peut être défini. Notons, dans ce cas, que tout RCQ $_C$ peut être transformé en temps polynomial, reconnu en tant que formule de Horn en temps polynomial et la satisfiabilité de la formule propositionnelle peut être décidée en temps polynomial.

Propriété	Formule CNF		Complexité algorithmique	
	Nombre de clauses	Taille des clauses	Temps	Espace
Infimum et Supremum	$2n(n-1) \binom{ \mathcal{B} }{2}$	3	$\mathcal{O}(n^2 \mathcal{B} ^2)$	$\mathcal{O}(1)$
Transitivité	$n(n-1) \binom{ \mathcal{B} }{2}$	2	$\mathcal{O}(n^2 \mathcal{B} ^2)$	$\mathcal{O}(1)$
Inverse	$2n(n-1) \mathcal{B} $	2	$\mathcal{O}(n^2 \mathcal{B})$	$\mathcal{O}(1)$
Faible composition	$2n(n-1)(n-2) \mathcal{B} ^2$	3	$\mathcal{O}(n^3 \mathcal{B} ^2)$	$\mathcal{O}(1)$
ALO	$2^k n(n-1)$	k	$\mathcal{O}(2^k n^2)$	$\mathcal{O}(k)$

TABLE 6.1 – Dimension de la formule et complexité algorithmique dans le pire des cas

6.3 Propriétés et complétude

6.3.1 Transformation inverse

Comme dans le cadre d'autres transformations, la première étape consiste à définir le « sens » d'un modèle d'une formule propositionnelle obtenue par T_C . Puisque les propriétés modélisées garantissent de pouvoir extraire du modèle une borne inférieure et une borne supérieure pour chaque contrainte qualitative. Cela permet, quel que soit \mathcal{N} un RCQ, de décrire un sous-RCQ de \mathcal{N} .

Définition 86 (T_C^{-1}) Soient $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ et μ un modèle de $T_C(\mathcal{N})$. $T_C^{-1}(\mu)$ est le RCQ $_C$ $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ tel que, pour tout $v_i, v_j \in \mathcal{V}$, $Q'_{(v_i, v_i)} = \{\text{id}\}$ et la contrainte qualitative $Q'_{(v_i, v_j)} = [r^- ; r^+]$ est définie par :

$$\begin{aligned} r^- &= \text{Sup}(\{r \in \mathcal{B} : [r \preceq Q_{(v_i, v_j)}] \in \mu\}); \\ r^+ &= \text{Inf}(\{r \in \mathcal{B} : [Q_{(v_i, v_j)} \preceq r] \in \mu\}). \end{aligned}$$

Dans un premier temps, concentrons-nous sur la transformation d'un RCQ $_C$. Nous pouvons montrer que le RCQ obtenu par T_C^{-1} vérifie deux propositions majeures.

Proposition 8 Soient \mathcal{N} un RCQ $_C$ et μ un modèle de $T_C(\mathcal{N})$. $T_C^{-1}(\mu) \subseteq \mathcal{N}$.

Preuve Soient $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ $_C$ et μ un modèle de $T_C(\mathcal{N})$. Posons $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ un RCQ tel que $\mathcal{N}' = T_C^{-1}(\mu)$ et montrons que $\mathcal{N}' \subseteq \mathcal{N}$. Pour cela, montrons que pour tout $v_i, v_j \in \mathcal{V}$, $Q_{(v_i, v_j)} = [r^- ; r^+]$ et $Q'_{(v_i, v_j)} = [r_{ij}^- ; r_{ij}^+]$ tels que $r^- \preceq r_{ij}^- \preceq r_{ij}^+ \preceq r^+$.

Du fait de la formule $\top_{\text{SAT}}^{\text{ALO}}$, nous savons qu'il existe r^- et r^+ tels que le cube $[r^- \preceq Q_{(v_i, v_j)}] \wedge [Q_{(v_i, v_j)} \preceq r^+]$ soit satisfait. Par définition de $\top_{\mathcal{C}}$ et puisque la propriété de transitivité est vérifiée, $r^- \preceq r_{ij}^-$ et $r_{ij}^+ \preceq r^+$. Montrons maintenant que $r_{ij}^- \preceq r_{ij}^+$. μ satisfait la propriété d'infimum et de supremum, nous pouvons en déduire que μ satisfait $r_{ij}^- \preceq Q_{(v_i, v_j)}$ et $Q_{(v_i, v_j)} \preceq r_{ij}^+$. Or, si $r_{ij}^- \prec r_{ij}^+$, alors μ ne satisfait pas la propriété de transitivité. Par contraposée, il s'ensuit que $r_{ij}^- \preceq r_{ij}^+$. Par conséquent, pour tout $v_i, v_j \in \mathcal{V}$, $Q'_{(v_i, v_j)} \subseteq Q_{(v_i, v_j)}$, et donc $\mathcal{N}' \subseteq \mathcal{N}$. \square

6.3.2 Transformation correcte et complète

Jusqu'ici, nous avons défini une transformation $\top_{\mathcal{C}}$ des RCQ vers les formules CNF, polynomiale en temps dans la taille du RCQ. Il convient maintenant de vérifier si celle-ci est correcte et complète pour le problème de la cohérence des RCQ. Avant cela, prouvons une propriété qui nous sera utile dans la suite concernant le treillis (\mathcal{B}, \preceq) :

Proposition 9 Soient $r_1, r_2, r_3, r_4 \in \mathcal{B}$. Si $r_1 \preceq r_3$ et $r_2 \preceq r_4$ alors nous avons $\text{Inf}(r_1 \diamond r_2) \preceq \text{Inf}(r_3 \diamond r_4)$ et $\text{Sup}(r_1 \diamond r_2) \preceq \text{Sup}(r_3 \diamond r_4)$.

Preuve Posons $r_1, r_2, r_3, r_4 \in \mathcal{B}$ tels que $r_1 \preceq r_3$ et $r_2 \preceq r_4$. Considérons les relations convexes $[r_1, r_3]$ et $[r_2, r_4]$. Celles-ci sont bien définies. Nous savons que $[r_1, r_3] \diamond [r_2, r_4] = [\text{Inf}(r_1 \diamond r_2), \text{Sup}(r_3 \diamond r_4)]$. De même, $[r_3, r_3] \diamond [r_4, r_4] = [\text{Inf}(r_3 \diamond r_4), \text{Sup}(r_3 \diamond r_4)]$. Par définition de la composition faible, puisque $[r_3, r_3] \subseteq [r_1, r_3]$ et $[r_4, r_4] \subseteq [r_2, r_4]$, nous savons que $[r_3, r_3] \diamond [r_4, r_4] \subseteq [r_1, r_3] \diamond [r_2, r_4]$, et donc $[\text{Inf}(r_3 \diamond r_4), \text{Sup}(r_3 \diamond r_4)] \subseteq [\text{Inf}(r_1 \diamond r_2), \text{Sup}(r_3 \diamond r_4)]$. Nous déduisons trivialement $\text{Inf}(r_1 \diamond r_2) \preceq \text{Inf}(r_3 \diamond r_4)$. Par un raisonnement similaire, $[r_1, r_1] \diamond [r_2, r_2] = [\text{Inf}(r_1 \diamond r_2), \text{Sup}(r_1 \diamond r_2)]$. $[r_1, r_1] \subseteq [r_1, r_3]$ et $[r_2, r_2] \subseteq [r_2, r_4]$, donc $[r_1, r_1] \diamond [r_2, r_2] \subseteq [r_1, r_3] \diamond [r_2, r_4]$. Ainsi, $[\text{Inf}(r_1 \diamond r_2), \text{Sup}(r_1 \diamond r_2)] \subseteq [\text{Inf}(r_1 \diamond r_2), \text{Sup}(r_3 \diamond r_4)]$. Par conséquent, $\text{Sup}(r_1 \diamond r_2) \preceq \text{Sup}(r_3 \diamond r_4)$. Nous pouvons donc conclure que si $r_1 \preceq r_3$ et $r_2 \preceq r_4$, alors $\text{Inf}(r_1 \diamond r_2) \preceq \text{Inf}(r_3 \diamond r_4)$ et $\text{Sup}(r_1 \diamond r_2) \preceq \text{Sup}(r_3 \diamond r_4)$. \square

Montrons tout d'abord que la transformation est correcte pour les $\text{RCQ}_{\mathcal{C}}$.

Proposition 10 Soit \mathcal{N} un $\text{RCQ}_{\mathcal{C}}$ défini sur (\mathcal{B}, \preceq) . Si \mathcal{N} est cohérent alors $\top_{\mathcal{C}}(\mathcal{N})$ est satisfiable.

Preuve Posons $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un $\text{RCQ}_{\mathcal{C}}$ cohérent défini sur (\mathcal{B}, \preceq) . Si \mathcal{N} est cohérent, alors il admet un scénario cohérent $\mathcal{S} = (\mathcal{V}, \mathcal{Q}')$ tel que $\mathcal{S} \subseteq \mathcal{N}$ et pour tout $v_i, v_j \in \mathcal{V}$, avec $i \neq j$, $Q'_{(v_i, v_j)} = \{s_{ij}\}$. Nous définissons une interprétation μ de $\top_{\mathcal{C}}(\mathcal{N})$ telle que, pour tout $v_i, v_j \in \mathcal{V}$, avec $i \neq j$, et pour chaque relation atomique $r \in \mathcal{B}$:

- si $r \preceq Q'_{(v_i, v_j)}$ alors $\mu([r \preceq Q_{(v_i, v_j)}]) = 1$, sinon $\mu([r \preceq Q_{(v_i, v_j)}]) = 0$;
- si $Q'_{(v_i, v_j)} \preceq r$ alors $\mu([Q_{(v_i, v_j)} \preceq r]) = 1$, sinon $\mu([Q_{(v_i, v_j)} \preceq r]) = 0$.

Puisque \mathcal{S} est un scénario de \mathcal{N} , nous avons $r_{ij}^- \preceq s_{ij} \preceq r_{ij}^+$, avec $Q_{(v_i, v_j)} = [r_{ij}^-, r_{ij}^+]$. Par conséquent, μ satisfait la formule $\top_{\text{SAT}}^{\text{ALO}}(\mathcal{N})$. De plus, $s_{ij} \in \mathcal{B}$ et (\mathcal{B}, \preceq) est un treillis, il s'ensuit que les propriétés d'infimum, supremum et transitivité sont vérifiées. Donc les formules $\top_{\text{SAT}}^{\text{I}}(\mathcal{N})$ et $\top_{\text{SAT}}^{\text{II}}(\mathcal{N})$ sont satisfaites par μ . Aussi, $s_{ij} = s_{ji}^{-1}$ et nous savons que si $r_1 \preceq r_2$ alors $r_2^{-1} \preceq r_1^{-1}$. Par conséquent, la propriété d'inverse est vérifiée et la formule $\top_{\text{SAT}}^{\text{III}}(\mathcal{N})$ est satisfaite par μ .

Montrons que la propriété de composition faible est vérifiée. Puisque \mathcal{S} est cohérent, il est \diamond -cohérent. Nous avons donc $s_{ij} \in [s_{ik} ; s_{ik}] \diamond [s_{kj} ; s_{kj}]$ pour tout $i, j, k \in [1 ; n]$. Ainsi, $\text{Inf}(s_{ik} \diamond s_{kj}) \preceq s_{ij} \preceq \text{Sup}(s_{ik} \diamond s_{kj})$. Pour tout $r_1, r_2 \in \mathcal{B}$, si $r_1 \preceq s_{ik}$ et $r_2 \preceq s_{kj}$ alors $\text{Inf}(r_1 \diamond r_2) \preceq \text{Inf}(s_{ik} \diamond s_{kj})$ par la proposition 9. Il s'ensuit que $\text{Inf}(r_1 \diamond r_2) \preceq s_{ij}$. De même, pour tout $r_1, r_2 \in \mathcal{B}$, si $s_{ik} \preceq r_1$ et $s_{kj} \preceq r_2$ alors $\text{Sup}(s_{ik} \diamond s_{kj}) \preceq \text{Sup}(r_1 \diamond r_2)$ par la proposition 9. Nous avons donc $s_{ij} \preceq \text{Sup}(r_1 \diamond r_2)$. La formule

$\mathsf{T}_{\text{SAT}}^{\text{IV}}(\mathcal{N})$ est donc satisfaite par μ . Par conséquent, μ est un modèle de la formule $\mathsf{T}_C(\mathcal{N})$. Ainsi, $\mathsf{T}_C(\mathcal{N})$ est satisfiable. \square

Pour montrer que la transformation est *complète*, nous exploitons la définition de la transformation inverse.

Proposition 11 Soient \mathcal{N} un RCQ_C et μ un modèle de $\mathsf{T}_C(\mathcal{N})$. $\mathsf{T}_C^{-1}(\mu)$ est un $\text{RCQ}_C \diamond$ -fermé.

Preuve Soient $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ_C et μ un modèle satisfaisant $\mathsf{T}_C(\mathcal{N})$. Posons $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ le RCQ_C vérifiant $\mathcal{N}' = \mathsf{T}_C^{-1}(\mu)$ et montrons que \mathcal{N}' est \diamond -fermé.

Considérons $v_i, v_j, v_k \in \mathcal{V}$, et posons $Q'_{(v_i, v_j)} = [r_{ij}^-; r_{ij}^+]$, $Q'_{(v_i, v_k)} = [r_{ik}^-; r_{ik}^+]$ et $Q'_{(v_k, v_j)} = [r_{kj}^-; r_{kj}^+]$. Montrons que $[r_{ij}^-; r_{ij}^+] \subseteq [r_{ik}^-; r_{ik}^+] \diamond [r_{kj}^-; r_{kj}^+]$. Par définition de \mathcal{N}' , nous savons $[r_{ik}^- \preceq Q_{(v_i, v_k)}] \in \mu$ et $[r_{kj}^- \preceq Q_{(v_k, v_j)}] \in \mu$, ainsi que $[Q_{(v_i, v_k)} \preceq r_{ik}^+] \in \mu$ et $[Q_{(v_k, v_j)} \preceq r_{kj}^+] \in \mu$. Puisque μ satisfait la propriété de composition faible, nous pouvons affirmer que $[\text{Inf}(r_{ik}^- \diamond r_{kj}^-) \preceq Q_{(v_i, v_j)}] \in \mu$ et $[Q_{(v_i, v_j)} \preceq \text{Sup}(r_{ik}^+ \diamond r_{kj}^+)] \in \mu$. Par la définition de la formule $\mathsf{T}_{\text{SAT}}^{\text{ALO}}$, $[r_{ij}^- \preceq Q_{(v_i, v_j)}] \in \mu$ et $[Q_{(v_i, v_j)} \preceq r_{ij}^+] \in \mu$. Puisque μ vérifie la propriété de transitivité, $[r_{ij}^- \preceq Q_{(v_i, v_j)}]$ et $[\text{Inf}(r_{ik}^- \diamond r_{kj}^-) \preceq Q_{(v_i, v_j)}]$, respectivement $[Q_{(v_i, v_j)} \in r_{ij}^+]$ et $[Q_{(v_i, v_j)} \preceq \text{Sup}(r_{ik}^+ \diamond r_{kj}^+)]$, sont mutuellement satisfaits si et seulement si $[\text{Inf}(r_{ik}^- \diamond r_{kj}^-) \preceq r_{ij}^-]$, respectivement $[r_{ij}^+ \preceq \text{Sup}(r_{ik}^+ \diamond r_{kj}^+)]$. Ainsi, $[\text{Inf}(r_{ik}^- \diamond r_{kj}^-) \preceq r_{ij}^- \preceq r_{ij}^+ \preceq \text{Sup}(r_{ik}^+ \diamond r_{kj}^+)]$, donc $[r_{ij}^-; r_{ij}^+] \subseteq [\text{Inf}(r_{ik}^- \diamond r_{kj}^-); \text{Sup}(r_{ik}^+ \diamond r_{kj}^+)]$. Or, $[r_{ik}^-; r_{ik}^+] \diamond [r_{kj}^-; r_{kj}^+] = [\text{Inf}(r_{ik}^- \diamond r_{kj}^-); \text{Sup}(r_{ik}^+ \diamond r_{kj}^+)]$, nous pouvons déduire que nous avons $[r_{ij}^-; r_{ij}^+] \subseteq [r_{ik}^-; r_{ik}^+] \diamond [r_{kj}^-; r_{kj}^+]$. Pour tout $v_i, v_j, v_k \in \mathcal{V}$, $Q'_{(v_i, v_j)} \subseteq Q'_{(v_i, v_k)} \diamond Q'_{(v_k, v_j)}$. \mathcal{N}' est donc un $\text{RCQ} \diamond$ -fermé. \square

Cependant, il est nécessaire de spécifier celle-ci à certains formalismes qualitatifs, pour lesquels la méthode de fermeture par composition faible est complète pour le problème de la cohérence des RCQ_C . Comme énoncé précédemment, l'algèbre des intervalles ou le formalisme qualitatif RCC font parties des formalismes qualitatifs vérifiant cette propriété. De la proposition 11, nous énonçons le corollaire :

Corollaire 4 Soit \mathcal{N} un RCQ_C pour lequel la méthode de fermeture par composition faible est complète pour le problème de la cohérence. Si $\mathsf{T}_C(\mathcal{N})$ est satisfiable, alors \mathcal{N} est cohérent.

Preuve La preuve est immédiate. Posons \mathcal{N} un RCQ_C pour lequel la méthode de fermeture par composition faible est complète pour le problème de la cohérence. Si $\mathsf{T}_C(\mathcal{N})$ est satisfiable, alors il est possible de construire un $\text{RCQ}_C \mathcal{N}' \diamond$ -fermé. Puisque \mathcal{N}' est \diamond -fermé, il est également cohérent par hypothèse. Or, $\mathcal{N}' \subseteq \mathcal{N}$ par la proposition 8. Par conséquent, \mathcal{N} est cohérent. \square

Pour ces formalismes spécifiques, la transformation T_C est donc correcte et complète pour le problème de la cohérence des RCQ_C .

Proposition 12 Soit \mathcal{N} un RCQ_C . Si la méthode de fermeture par composition faible est complète pour la cohérence des RCQ_C , alors \mathcal{N} est cohérent si et seulement si $\mathsf{T}_C(\mathcal{N})$ est satisfiable.

Preuve Dans le sens « si », supposons que \mathcal{N} soit cohérent. D'après la proposition 10, nous pouvons affirmer que $\mathsf{T}_C(\mathcal{N})$ est satisfiable. Dans le sens « seulement si », supposons que $\mathsf{T}_C(\mathcal{N})$ admet un modèle μ . D'après le corollaire 4, nous pouvons affirmer que \mathcal{N} est cohérent. \square

Remarquons que la transformation T_C sur un RCQ_C peut être effectuée en temps polynomiale dans la taille du RCQ_C et que la formule obtenue est une formule de Horn. La transformation est correcte et complète pour le problème de la cohérence des RCQ_C , il est possible de répondre en temps polynomiale à la cohérence d'un RCQ_C en passant par la transformation définie. Ce résultat rejoint évidemment les nombreuses études ayant démontré la traitabilité de l'ensemble des relations convexes.

Nous pouvons maintenant étendre les résultats obtenus pour les RCQ_C au cadre général des RCQ . Pour cela, nous posons une proposition permettant de faire le lien entre RCQ_C et RCQ .

Proposition 13 Soient \mathcal{N} un RCQ et \mathcal{N}' un RCQC tel que $\mathcal{N}' \subseteq \mathcal{N}$, et pour tout $v_i, v_j \in \mathcal{V}$, $Q_{(v_i, v_j)} = [r_1^- ; r_1^+] \cup \dots \cup [r_k^- ; r_k^+]$ et $Q'_{(v_i, v_j)} = [r_l^- ; r_l^+]$ avec $l \in [1 ; k]$. Si $\mathsf{T}_C(\mathcal{N}')$ est satisfiable alors $\mathsf{T}_C(\mathcal{N})$ est satisfiable.

Preuve Tout d'abord, rappelons que $\text{vars}(\mathsf{T}_C(\mathcal{N}')) = \text{vars}(\mathsf{T}_C(\mathcal{N}))$. Posons μ' un modèle de $\mathsf{T}_C(\mathcal{N}')$, et $v_i, v_j \in \mathcal{V}$, avec $i \neq j$. Par définition de T_C , puisque \mathcal{N} et \mathcal{N}' sont construits sur un même formalisme qualitatif, les formules $\mathsf{T}_C(\mathcal{N}')$ et $\mathsf{T}_C(\mathcal{N})$ ne diffèrent que des clauses obtenues par $\mathsf{T}_{\text{SAT}}^{\text{ALO}}$. Par conséquent, μ' est un modèle de $\mathsf{T}_{\text{SAT}}^{\text{I}}(\mathcal{N}') \wedge \mathsf{T}_{\text{SAT}}^{\text{II}}(\mathcal{N}') \wedge \mathsf{T}_{\text{SAT}}^{\text{III}}(\mathcal{N}') \wedge \mathsf{T}_{\text{SAT}}^{\text{IV}}(\mathcal{N}')$ et il existe un modèle μ de $\mathsf{T}_{\text{SAT}}^{\text{I}}(\mathcal{N}) \wedge \mathsf{T}_{\text{SAT}}^{\text{II}}(\mathcal{N}) \wedge \mathsf{T}_{\text{SAT}}^{\text{III}}(\mathcal{N}) \wedge \mathsf{T}_{\text{SAT}}^{\text{IV}}(\mathcal{N})$ tel que, pour tout $v_i, v_j \in \mathcal{V}$ et pour tout $r \in \mathcal{B}$, $[r \preceq Q_{(v_i, v_j)}] \in \mu$ si et seulement si $[r \preceq Q'_{(v_i, v_j)}] \in \mu'$. De plus, par définition de $\mathsf{T}_{\text{SAT}}^{\text{ALO}}$ sur un RCQC, $\mathsf{T}_{\text{SAT}}^{\text{ALO}}(\mathcal{N}')$ est une formule CNF ne contenant que des clauses unitaires. En particulier, μ' satisfait $[r_i^- \preceq Q'_{(v_i, v_j)}] \wedge [Q'_{(v_i, v_j)} \preceq r_i^+]$. Or, $Q_{(v_i, v_j)} = [r_1^- ; r_1^+] \cup \dots \cup [r_k^- ; r_k^+]$ et $Q'_{(v_i, v_j)} = [r_l^- ; r_l^+]$ avec $l \in [1 ; k]$. $\mathsf{T}_{\text{SAT}}^{\text{ALO}}(\mathcal{N})$ est une formule DNF pour laquelle il existe un cube tel $[r_l^- \preceq Q_{(v_i, v_j)}] \wedge [Q_{(v_i, v_j)} \preceq r_l^+]$. Puisque $\mathcal{N}' \subseteq \mathcal{N}$, $[r_l^- \preceq Q'_{(v_i, v_j)}] \in \mu'$ et $[Q'_{(v_i, v_j)} \preceq r_l^+] \in \mu'$, nous pouvons affirmer que si $[r_l^- \preceq Q_{(v_i, v_j)}] \in \mu$ et $[Q_{(v_i, v_j)} \preceq r_l^+] \in \mu$, alors μ satisfait $[r_l^- \preceq Q_{(v_i, v_j)}] \wedge [Q_{(v_i, v_j)} \preceq r_l^+]$. Donc, pour tout $v_i, v_j \in \mathcal{V}$ et $r \in \mathcal{B}$, μ' satisfait $\mathsf{T}_{\text{SAT}}^{\text{ALO}}(\mathcal{N}')$ et il existe un modèle μ satisfaisant $\mathsf{T}_{\text{SAT}}^{\text{ALO}}(\mathcal{N})$. En conséquence, si $\mathsf{T}_C(\mathcal{N}')$ est satisfiable alors $\mathsf{T}_C(\mathcal{N})$ est satisfiable. \square

La transformation d'un RCQ \mathcal{N} est basée sur un découpage des relations complexes en sous-relations convexes. Nous pouvons donc étendre ces propositions au cas général des RCQ en utilisant le découpage.

Proposition 14 Soit \mathcal{N} un RCQ. Si la méthode de la fermeture par faible composition est complète pour les RCQC alors \mathcal{N} est cohérent si et seulement si $\mathsf{T}_C(\mathcal{N})$ est satisfiable.

Preuve Posons pour tout $v_i, v_j \in \mathcal{V}$, avec $i \neq j$, la contrainte $Q_{(v_i, v_j)} = [r_1^- ; r_1^+] \cup \dots \cup [r_k^- ; r_k^+]$ avec k un entier et pour tout $l \in [1 ; k]$, $r_l^-, r_l^+ \in \mathcal{B}$. Dans le sens « si ». Si \mathcal{N} est cohérent, alors il admet un sous-RCQ $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ cohérent tel que, pour tout $v_i, v_j \in \mathcal{V}$, $Q'_{(v_i, v_j)}$ est définie par une relation $[r_l^- ; r_l^+]$ avec $l \in [1 ; k]$ issue du découpage de $Q_{(v_i, v_j)}$. De la proposition 10, puisque \mathcal{N}' est cohérent, nous savons que $\mathsf{T}_C(\mathcal{N}')$ est satisfiable, et par la proposition 13, $\mathsf{T}_C(\mathcal{N})$ est également satisfiable. Dans le sens « seulement si ». Posons un modèle μ satisfaisant $\mathsf{T}_C(\mathcal{N})$. Nous savons, pour chaque contrainte $Q_{(v_i, v_j)} \in \mathcal{Q}$, qu'il existe un entier $l \in [1 ; k]$ tel que μ satisfait la conjonction $[r_l^- \preceq Q_{(v_i, v_j)}] \wedge [Q_{(v_i, v_j)} \preceq r_l^+]$. En définissant $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ par $Q'_{(v_i, v_j)} = [r_l^- ; r_l^+]$, nous obtenons un RCQC tel que $\mathcal{N}' \subseteq \mathcal{N}$, dont la transformation $\mathsf{T}_C(\mathcal{N}')$ est satisfaite par μ . Par conséquent, d'après la proposition 12, \mathcal{N}' est cohérent. Et donc \mathcal{N} est également cohérent. Nous pouvons donc conclure que \mathcal{N} est cohérent si et seulement si $\mathsf{T}_C(\mathcal{N})$ est satisfiable. \square

Dans le cas de RCQ, la transformation T_C est correcte et complète pour décider de la cohérence d'un RCQ.

6.4 Conclusion

Dans ce chapitre, nous avons défini une transformation des RCQ en logique propositionnelle en exploitant un ordre partiel sur les relations atomiques du formalisme qualitatif utilisé, permettant de décrire une structure de treillis. Cette structure apparaît pour de nombreux formalismes, tels que l'algèbre des intervalles ou le formalisme RCC parmi les plus connus, et permet la caractérisation de classes traitables, telle que la classe convexe. En axiomatisant les propriétés du treillis, nous proposons une transformation polynomiale en temps, basée sur le découpage des relations du RCQ en relation convexe.

Dans le cadre des RCQ convexes, nous montrons que la formule obtenue est une formule de Horn, que nous pouvons construire, reconnaître et décider en temps polynomial dans la taille du RCQ. Pour les RCQ en général, exhiber un modèle de la formule résultant de notre transformation permet de caractériser un sous-RCQ convexe. En restreignant notre étude au cadre des formalismes qualitatifs pour lesquels la méthode de fermeture par composition faible est correcte et complète pour le problème de la cohérence des RCQ convexes, nous prouvons que la transformation T_C est correcte et complète pour décider de la cohérence d'un RCQ.

Pour poursuivre cette étude, nous souhaitons étendre ce cadre à d'autres classes traitables, notamment la classe des préconvexes qui est un sur-ensemble de la classe des convexes, afin d'obtenir un découpage plus faible des relations complexes du RCQ. Cette approche présente un inconvénient pratique qui résulte de la taille de la formule propositionnelle obtenue. Les mêmes propriétés sont définies sur tout couple ou triplet de variables. Comme pour la transformation proposée précédemment (Chapitre 6), il nous semble intéressant d'orienter les recherches vers la compilation des connaissances ou encore vers le cadre SMT (*SAT modulo theory*), permettant une représentation plus compacte. Nous entrevoyons également la possibilité d'adapter cette approche aux transformations des RCQ vers les RCD, pour bénéficier de la compacité des RCD et des outils algorithmiques développés dans ce cadre.

Chapitre 7

Du problème CSP qualitatif au problème CSP

Sommaire

7.1	Introduction	105
7.2	Définition de la transformation	106
7.3	Propriétés liées à la transformation	107
	7.3.1 Transformation inverse	107
	7.3.2 Complétude	107
	7.3.3 Cohérence locale	108
7.4	Conclusion	110

7.1 Introduction

De nombreuses transformations vers le cadre propositionnelle ont été présentées. L'un des inconvénients majeurs réside dans la taille de la formule propositionnelle résultante. Les RCD offrent un cadre intéressant, avec des outils algorithmiques performants et une compacité de la représentation des informations. Le cadre que nous introduisons ici vise à faire l'analogie entre la microstructure d'un RCQ et la microstructure du RCD, et de modéliser la propriété de composition faible du cadre qualitatif. Pour réaliser cela, chaque contrainte qualitative du RCQ est modélisée par une variable du RCD, et chaque relation complexe décrivant la contrainte est modélisée par le domaine de la variable associée. Ainsi, à un RCQ est associé un RCD unique et, en particulier, à un scénario est associé un RCD atomique. Pour vérifier la cohérence locale du RCQ, nous considérons la composition faible comme une contrainte entre les relations atomiques, et nous modélisons la table de composition du formalisme qualitatif étudié en une contrainte entre les valeurs. Nous montrons ainsi le lien entre la méthode de la fermeture par composition faible et cohérence d'arcs généralisée. Cependant, nous devons nous restreindre à certains formalismes qualitatifs, pour lesquels la méthode de fermeture par composition faible permet de vérifier la cohérence d'un scénario.

Dans ce chapitre, nous proposons et étudions une transformation des RCQ vers les RCD. Dans un premier temps, nous définissons le lien entre les microstructures, dont nous vérifions la cohérence en décrivant la table de composition du formalisme qualitatif étudié par une contrainte du RCD. Cette approche nous permet ensuite de faire le lien entre différentes formes de cohérence locale.

7.2 Définition de la transformation

Pour répondre au problème de la cohérence d'un RCQ, l'approche par recherche exhaustive s'attache à énumérer l'ensemble des scénarios et à vérifier leur cohérence. Nous souhaitons pouvoir modéliser l'ensemble de ces scénarios à l'aide d'un RCD. Pour cela, l'approche la plus naturelle consiste à caractériser le lien entre RCQ atomique et RCD atomique. Cependant, il faut garantir la cohérence locale de ce scénario, c'est-à-dire que celui-ci soit \diamond -cohérent. Par conséquent, la propriété de faible composition doit être modélisée dans le RCD et, pour cela, nous utilisons la table de composition du formalisme qualitatif utilisé.

L'approche la plus naturelle consiste à associer une variable $x_{ij} \in \mathcal{X}$ du RCD à la relation de chaque contrainte qualitative Q_{v_i, v_j} du RCQ pour tout $v_i, v_j \in \mathcal{V}$. Ainsi, à chaque valeur $r \in D_{x_{ij}}$ du RCD est associée une relation atomique $r \in Q_{v_i, v_j}$ ⁶. Par conséquent, il est nécessaire d'imposer la contrainte d'identité sur toute variable $x_{ii} \in \mathcal{X}$, la contrainte d'inverse sur tout couple de variables $(x_{ij}, x_{ji}) \in \mathcal{X} \times \mathcal{X}$ et la composition faible sur tout triplet de variables $(x_{ij}, x_{ik}, x_{kj}) \in \mathcal{X}^3$. Cependant, nous souhaitons manipuler des RCQ normalisés et caractériser un scénario cohérent. Nous imposons donc les contraintes d'identité et d'inverse implicitement, en nous restreignant à l'ensemble des variables x_{ij} telle que $i < j$. Ainsi, x_{ii} représente la relation identité, et puisque pour tout $v_i, v_j \in \mathcal{V}$, $Q_{(v_j, v_i)} = Q_{(v_i, v_j)}^{-1}$, nous pouvons déduire les valeurs du domaine de x_{ji} à partir des valeurs du domaine de x_{ij} .

Définition 87 ($T_{\text{DCN}}(\mathcal{N})$) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ construit sur un formalisme \mathcal{F} de relations atomiques \mathcal{B} , avec $|\mathcal{V}| = n$. $T_{\text{DCN}}(\mathcal{N})$ est la transformation de \mathcal{N} permettant de définir un RCD $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ tel que :

- pour tout $v_i, v_j \in \mathcal{V}$, avec $1 \leq i < j \leq n$, est introduite la variable $x_{ij} \in \mathcal{X}$ telle que $D_{x_{ij}} = Q_{(v_i, v_j)}$;
- pour tout $v_i, v_j, v_k \in \mathcal{V}$, avec $1 \leq i < j < k \leq n$, nous introduisons une contrainte ternaire $c_{(x_{ij}, x_{ik}, x_{kj})}$ définie par $c_{(x_{ij}, x_{ik}, x_{kj})} = \{(c, a, b) \in \mathcal{B}^3 : c \in a \diamond b\}$.

Ainsi, le RCD possède $\binom{n}{2}$ variables, et la taille du plus grand domaine est $|\mathcal{B}|$, dans le pire des cas. Les contraintes du RCQ sont liées par la table de composition faible. Chaque contrainte du RCD correspond donc à un triplet de contraintes qualitatives du RCQ, afin de modéliser la propriété de faible composition. Le RCD est donc composé de $\binom{n}{3}$ contraintes ternaires. Le nombre de tuples de chaque contrainte du RCD est fonction du formalisme qualitatif et de la table de composition. Dans le pire des cas, il est de $3^{|\mathcal{B}|}$. La complexité algorithmique en temps (resp. en espace) dans le pire des cas pour appliquer la transformation T_{DCN} est donc $\mathcal{O}(3^{|\mathcal{B}|}n^3)$ (resp. $\mathcal{O}(|\mathcal{B}|)$). T_{DCN} est donc une transformation polynomiale en temps dans la taille du RCQ. Le tableau 7.1 récapitule les dimensions du RCD résultant de la transformation T_{DCN} dans le pire des cas.

		Dimension		Complexité	
		\mathcal{N} un RCQ	$T_{\text{DCN}}(\mathcal{N})$ un RCD	Temps	Espace
Variables	Nombre	n	$\binom{n}{2}$	$\mathcal{O}(n^2 \mathcal{B})$	$ \mathcal{B} $
	Taille des domaines	∞	$ \mathcal{B} $		
Contraintes	Nombre	$\binom{n}{2}$	$\binom{n}{3}$	$\mathcal{O}(n^3 \mathcal{B} ^3)$	$ 1 $
	Arité	2	3		
	Nombre de tuples	\mathcal{B}	$3^{ \mathcal{B} }$		

TABLE 7.1 – Dimensions du RCD et complexité algorithmique pour appliquer la transformation T_{DCN}

6. Remarquons que nous pouvons nous abstraire du sens mathématiques des relations atomiques de \mathcal{B} , puisque celles-ci sont associées à une valeur discrète dans le RCD, représentée par le symbole de la relation atomique.

7.3 Propriétés liées à la transformation

7.3.1 Transformation inverse

Notre transformation définie, nous pouvons faire la corrélation entre RCQ atomique et RCD atomique. Plus spécifiquement, entre scénario et instanciation complète. Pour cela, définissons dans un premier temps la transformation *inverse*, que nous dénoterons T_{DCN}^{-1} .

Définition 88 (T_{DCN}^{-1}) Soient $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ construit sur un formalisme \mathcal{F} , tel que $|\mathcal{V}| = n$, et $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD tel que $\mathcal{P} \subseteq T_{\text{DCN}}(\mathcal{N})$. $T_{\text{DCN}}^{-1}(\mathcal{P})$ est le RCQ $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ construit sur \mathcal{F} défini par :

- pour tout $v_i \in \mathcal{V} : Q'_{(v_i, v_i)} = \{\text{id}\}$;
- pour tout $v_i, v_j \in \mathcal{V}$ avec $i < j : Q'_{(v_i, v_j)} = D_{x_{ij}}$;
- pour tout $v_i, v_j \in \mathcal{V}$ avec $i < j : Q'_{(v_j, v_i)} = Q'_{(v_i, v_j)}^{-1}$.

Cette transformation n'est évidemment pas définie sur tout RCD, mais sur l'ensemble des RCD qui peuvent être obtenus par la transformation T_{DCN} . Cela permet de conserver un formalisme qualitatif correctement défini. La transformation inverse permet d'énoncer quelques propositions qui peuvent être prouvées trivialement par construction.

Proposition 15 Soit \mathcal{N} un RCQ.

- a) $\mathcal{N} = T_{\text{DCN}}^{-1}(T_{\text{DCN}}(\mathcal{N}))$;
- b) si \mathcal{N} est atomique alors $T_{\text{DCN}}(\mathcal{N})$ l'est aussi ;
- c) si $\mathcal{P} \subseteq T_{\text{DCN}}(\mathcal{N})$ alors $T_{\text{DCN}}^{-1}(\mathcal{P}) \subseteq \mathcal{N}$;
- d) si $\mathcal{N} \subseteq \mathcal{N}'$ alors $T_{\text{DCN}}(\mathcal{N}) \subseteq T_{\text{DCN}}(\mathcal{N}')$.

7.3.2 Complétude

L'objectif principal de la transformation est de permettre de répondre au problème de la cohérence d'un RCQ à l'aide d'un RCD. Dans un premier temps, nous pouvons affirmer que la transformation est *correcte*, c'est-à-dire que toute solution du RCQ est bien associée à une solution du RCD.

Proposition 16 Soient \mathcal{N} un RCQ et $\mathcal{P} = T_{\text{DCN}}(\mathcal{N})$. Si \mathcal{N} est cohérent alors \mathcal{P} est cohérent.

Preuve Soient $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ et $T_{\text{DCN}}(\mathcal{N})$ un RCD $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$.

Si \mathcal{N} est cohérent alors il existe un scénario cohérent $\mathcal{S} = (\mathcal{V}, \mathcal{Q}')$ de \mathcal{N} . Pour tout $v_i, v_j, v_k \in \mathcal{V}$ avec $1 \leq i < k < j \leq n$, posons $Q'_{(v_i, v_j)} = \{s_{ij}\}$, $Q'_{(v_i, v_k)} = \{s_{ik}\}$ et $Q'_{(v_k, v_j)} = \{s_{kj}\}$. Puisque \mathcal{S} est cohérent, alors \mathcal{S} est \diamond -fermé. Donc pour tout $v_i, v_j, v_k \in \mathcal{V}$ avec $1 \leq i < k < j \leq n$: $(s_{ij}, s_{ik}, s_{kj}) \in \text{TC}$. Par construction, puisque $\mathcal{S} \subseteq \mathcal{N}$, $s_{ij} \in D_{x_{ij}}$ pour tout $1 \leq i < j \leq n$. Par conséquent, il est possible de construire une instanciation complète \mathcal{I} de \mathcal{P} telle que $\mathcal{I}(x_{ij}) = s_{ij}$. Ainsi, $(\mathcal{I}(x_{ij}), \mathcal{I}(x_{ik}), \mathcal{I}(x_{kj})) = (s_{ij}, s_{ik}, s_{kj})$ est un tuple de la table de composition. Pour tout $1 \leq i < k < j \leq n$: $(\mathcal{I}(x_{ij}), \mathcal{I}(x_{ik}), \mathcal{I}(x_{kj})) \in c_{(x_{ij}, x_{ik}, x_{kj})}$. \mathcal{I} satisfait chaque contrainte de \mathcal{P} , elle est donc une solution de \mathcal{P} . Par conséquent, \mathcal{P} est cohérent. \square

Cependant, bien que la transformation soit correcte, celle-ci n'est pas *complète* dans le cas général.

Propriété 12 Soient \mathcal{N} un RCQ et $\mathcal{P} = T_{\text{DCN}}(\mathcal{N})$. Si \mathcal{P} est cohérent alors il existe un scénario $\mathcal{S} \subseteq \mathcal{N}$ \diamond -cohérent.

Preuve Soient $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ et $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C}) = \mathsf{T}_{\text{DCN}}(\mathcal{N})$ un RCD.

Si \mathcal{P} est cohérent alors il existe une solution \mathcal{I} telle que, pour tout $1 \leq i < k < j \leq n$, $(\mathcal{I}(x_{ij}), \mathcal{I}(x_{ik}), \mathcal{I}(x_{kj})) \in c_{(x_{ij}, x_{ik}, x_{kj})}$. À partir de \mathcal{I} , il est possible de construire un scénario $\mathcal{S} = (\mathcal{V}, \mathcal{Q}')$, avec $\mathcal{S} \subseteq \mathcal{N}$, tel que $Q'_{(v_i, v_j)} = \{\mathcal{I}(x_{ij})\}$, mais également $Q'_{(v_j, v_i)} = Q'_{(v_i, v_j)}^{-1}$ et $Q'_{(v_i, v_i)} = \{\text{id}\}$, pour tout $v_i, v_j \in \mathcal{V}$. Pour démontrer la \diamond -cohérence du scénario \mathcal{S} , procédons par cas. Dans un premier temps, posons trois variables distinctes $v_i, v_j, v_k \in \mathcal{V}$ telle que $1 \leq i < j < k \leq n$. Puisque $(\mathcal{I}(x_{ij}), \mathcal{I}(x_{ik}), \mathcal{I}(x_{kj}))$ est un tuple de la table de composition, nous savons qu'il existe $d_i, d_j, d_k \in \mathbb{D}$ tels que $(d_i, d_j) \in Q'_{(v_i, v_j)}$, $(d_i, d_k) \in Q'_{(v_i, v_k)}$ et $(d_k, d_j) \in Q'_{(v_k, v_j)}$, et $(d_i, d_i) \in Q'_{(v_i, v_i)}$, $(d_j, d_j) \in Q'_{(v_j, v_j)}$, $(d_k, d_k) \in Q'_{(v_k, v_k)}$. \mathcal{S} est donc cohérent pour tout triplet de variables distinctes. Donc $Q'_{(v_i, v_j)} \subseteq Q'_{(v_i, v_k)} \diamond Q'_{(v_k, v_j)}$, pour tout $v_i, v_j, v_k \in \mathcal{V}$ telle que $1 \leq i < k < j \leq n$. Par la proposition 1, puisque le scénario est normalisé, nous pouvons affirmer qu'il est \diamond -cohérent pour tout $v_i, v_j, v_k \in \mathcal{V}$. \square

Remarquons que la preuve illustre la transformation inverse $\mathsf{T}_{\text{DCN}}^{-1}$ permettant à une instanciation d'un RCD obtenu par T_{DCN} d'associer un scénario unique. De plus, cette preuve exprime la raison pour laquelle il n'est pas nécessaire de transformer l'ensemble des contraintes du RCQ en RCD, c'est-à-dire pourquoi seules les contraintes orientées sur les couples de variables distinctes sont nécessaires. À partir des contraintes orientées et en faisant l'hypothèse que le RCQ est normalisé, le résultat peut être généralisé à tout couple de variables par la proposition 1.

Nous pouvons donc formuler une hypothèse sur le formalisme qualitatif utilisé, assurant à chaque scénario \diamond -cohérent d'être cohérent.

Corollaire 5 Soit \mathcal{N} un RCQ construit dans un formalisme qualitatif \mathcal{F} . Si la \diamond -cohérence est complète pour décider de la cohérence d'un scénario, alors \mathcal{N} est cohérent si et seulement si $\mathsf{T}_{\text{DCN}}(\mathcal{N})$ est cohérent.

Pour de nombreux formalismes qualitatifs, ce corollaire peut être exploité. Ainsi, nous avons une transformation correcte et complète pour décider de la cohérence d'un RCQ. Nous pouvons utiliser les algorithmes de résolution et les techniques d'inférences du cadre des RCD pour décider de la cohérence d'un RCQ.

7.3.3 Cohérence locale

Le second objectif de la transformation est de mettre en évidence les liens entre les techniques d'inférence du cadre discret et du cadre qualitatif. En particulier, nous faisons le rapprochement entre la composition faible des RCQ et les formes de cohérences locales des RCD.

Proposition 17 Soit \mathcal{N} un RCQ \diamond -fermé. \mathcal{N} et $\mathsf{T}_{\text{DCN}}(\mathcal{N})$ sont $(0, 3)$ -cohérent.

Preuve Posons $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ fermé par composition faible. Nous savons que pour tout $1 \leq i, j, k \leq n$, $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$. Notamment, il existe $r_{ij} \in Q_{(v_i, v_j)}$, $r_{ik} \in Q_{(v_i, v_k)}$ et $r_{kj} \in Q_{(v_k, v_j)}$ telles que $r_{ij} \in r_{ik} \diamond r_{kj}$.

Dans un premier temps, montrons que \mathcal{N} est $(0, 3)$ -cohérent. Par définition de la composition faible, il existe des valeurs $d_i, d_j, d_k \in \mathbb{D}$ telles que $(d_i, d_j) \in r_{ij}$, $(d_i, d_k) \in r_{ik}$ et $(d_k, d_j) \in r_{kj}$. Par la définition d'inverse d'une relation, $r_{ji} \in Q_{(v_j, v_i)}$, $r_{ki} \in Q_{(v_k, v_i)}$ et $r_{jk} \in Q_{(v_j, v_k)}$ et donc $(d_j, d_i) \in r_{ji}$, $(d_k, d_i) \in r_{ki}$ et $(d_j, d_k) \in r_{jk}$. Enfin, $(d_i, d_i) \in Q_{(v_i, v_i)}$, $(d_k, d_k) \in Q_{(v_k, v_k)}$ et $(d_j, d_j) \in Q_{(v_j, v_j)}$ puisque $Q_{(v_i, v_i)} = Q_{(v_k, v_k)} = Q_{(v_j, v_j)} = \text{id}$. Par conséquent, si \mathcal{N} est \diamond -fermé, alors il existe une instanciation cohérente de \mathcal{N} de taille 3. Donc \mathcal{N} est $(0, 3)$ -cohérent.

Pour montrer que $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C}) = \mathsf{T}_{\text{DCN}}(\mathcal{N})$ est $(0, 3)$ -cohérent, considérons trois variables $x_{ij}, x_{ik}, x_{kj} \in \mathcal{X}$ tel que $1 \leq i < k < j \leq n$. Puisque \mathcal{N} est \diamond -fermé, nous avons $r_{ij} \in r_{ik} \diamond r_{kj}$ avec $r_{ij} \in D_{x_{ij}}, r_{ik} \in D_{x_{ik}}$ et $r_{kj} \in D_{x_{kj}}$, et donc $(r_{ij}, r_{ik}, r_{kj}) \in c_{(x_{ij}, x_{ik}, x_{kj})}$. Si \mathcal{N} est \diamond -fermé, alors il existe une instanciation cohérente de $\mathsf{T}_{\text{DCN}}(\mathcal{N})$ de taille 3 pour tout triplet de variables $x_{ij}, x_{ik}, x_{kj} \in \mathcal{X}$. $\mathsf{T}_{\text{DCN}}(\mathcal{N})$ est donc $(0, 3)$ -cohérent. \square

Proposition 18 Soit \mathcal{N} un RCQ \diamond -fermé construit sur un formalisme qualitatif. $\mathsf{T}_{\text{DCN}}(\mathcal{N})$ est fermé par 3-cohérence forte.

Preuve Soient $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ \diamond -fermé et $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C}) = \mathsf{T}_{\text{DCN}}(\mathcal{N})$ un RCD. $\mathsf{T}_{\text{DCN}}(\mathcal{N})$ est fermé par 3-cohérence forte si et seulement si il est $(0, 1)$ -fermé, $(1, 1)$ -fermé et $(2, 1)$ -fermé :

$(0, 1)$ -fermeture : \mathcal{N} est \diamond -fermé, il est \diamond -cohérent. Pour tout $v_i, v_j \in \mathcal{V}$, $Q_{(v_i, v_j)} \neq \emptyset$. Par conséquent, $D_{x_{ij}} \neq \emptyset$ pour tout $x_{ij} \in \mathcal{X}$. $\mathsf{T}_{\text{DCN}}(\mathcal{N})$ est donc fermé par cohérence de nœuds.

$(1, 1)$ -fermeture : Considérons $v_i, v_j, v_k \in \mathcal{V}$ avec $1 \leq i < k < j \leq n$. Pour tout $r_{ij} \in Q_{(v_i, v_j)}$, il existe $r_{ik} \in Q_{(v_i, v_k)}$ (resp. $r_{kj} \in Q_{(v_k, v_j)}$) tel que (r_{ij}, r_{ik}, r) (resp. (r_{ij}, r, r_{kj})) $\in \text{TC}$ (resp. $(r_{ij}, r, r_{kj}) \in \text{TC}$) quelle que soit $r \in \mathcal{B}$. De même, pour tout $r_{ik} \in Q_{(v_i, v_k)}$, il existe $r_{ij} \in Q_{(v_i, v_j)}$ (resp. $r_{kj} \in Q_{(v_k, v_j)}$) tel que (r_{ij}, r_{ik}, r) (resp. (r, r_{ik}, r_{kj})) $\in \text{TC}$ (resp. $(r, r_{ik}, r_{kj}) \in \text{TC}$) quelle que soit $r \in \mathcal{B}$. Enfin, pour tout $r_{kj} \in Q_{(v_k, v_j)}$, il existe $r_{ij} \in Q_{(v_i, v_j)}$ (resp. $r_{ik} \in Q_{(v_i, v_k)}$) tel que (r_{ij}, r, r_{kj}) (resp. (r, r_{ik}, r_{kj})) $\in \text{TC}$ (resp. $(r, r_{ik}, r_{kj}) \in \text{TC}$) quelle que soit $r \in \mathcal{B}$. Pour tout $r_{ij} \in Q_{(v_i, v_j)}$, $r_{ik} \in Q_{(v_i, v_k)}$ et $r_{kj} \in Q_{(v_k, v_j)}$ avec $1 \leq i < k < j \leq n$, la transformation T_{DCN} définit $r_{ij} \in D_{x_{ij}}$, $r_{ik} \in D_{x_{ik}}$ et $r_{kj} \in D_{x_{kj}}$ respectivement. Par conséquent, $x_{ij}, x_{ik}, x_{kj} \in \mathcal{X}$ sont fermés par GAC pour tout $1 \leq i < k < j \leq n$. Et donc $\mathsf{T}_{\text{DCN}}(\mathcal{N})$ est fermé par GAC.

$(2, 1)$ -fermeture : Considérons $v_i, v_j, v_k \in \mathcal{V}$ pour tout $1 \leq i < k < j \leq n$. Considérons $v_i, v_j, v_k \in \mathcal{V}$ avec $1 \leq i < k < j \leq n$. \mathcal{N} est \diamond -fermé et \mathcal{N} est un graphe complet. Par conséquent, pour tout $r_{ij} \in Q_{(v_i, v_j)}$, $r_{ik} \in Q_{(v_i, v_k)}$, il existe $r \in \mathcal{B}$ tel que $(r_{ij}, r_{ik}, r) \in \text{TC}$. Par la transformation T_{DCN} , pour tout $r_{ij} \in D_{x_{ij}}$ et $r_{ik} \in D_{x_{ik}}$ tels que $(r_{ij}, r_{ik}, r) \in c_{(x_{ij}, x_{ik}, x_{kj})}$. $x_{ij}, x_{ik} \in \mathcal{X}$ avec $1 \leq i < k < j \leq n$ sont fermés par cohérence de chemins. De même, pour tout $r_{ij} \in Q_{(v_i, v_j)}$, $r_{kj} \in Q_{(v_k, v_j)}$, il existe $r \in \mathcal{B}$ tel que $(r_{ij}, r, r_{kj}) \in \text{TC}$. Par T_{DCN} , pour tout $r_{ij} \in D_{x_{ij}}$ et $r_{kj} \in D_{x_{kj}}$ tels que $(r_{ij}, r, r_{kj}) \in c_{(x_{ij}, x_{ik}, x_{kj})}$. $x_{ij}, x_{kj} \in \mathcal{X}$ avec $1 \leq i < k < j \leq n$ sont fermés par cohérence de chemins. Enfin, pour tout $r_{ik} \in Q_{(v_i, v_k)}$, $r_{kj} \in Q_{(v_k, v_j)}$, il existe $r \in \mathcal{B}$ tel que $(r, r_{ik}, r_{kj}) \in \text{TC}$. Donc, par T_{DCN} , pour tout $r_{ik} \in D_{x_{ik}}$ et $r_{kj} \in D_{x_{kj}}$ tels que $(r_{ik}, r, r_{kj}) \in c_{(x_{ij}, x_{ik}, x_{kj})}$. $x_{ik}, x_{kj} \in \mathcal{X}$ avec $1 \leq i < k < j \leq n$ sont fermés par cohérence de chemins. Par conséquent, $x_{ij}, x_{ik}, x_{kj} \in \mathcal{X}$ sont fermés par cohérence de chemins pour tout $1 \leq i < k < j \leq n$. $\mathsf{T}_{\text{DCN}}(\mathcal{N})$ est donc fermé par cohérence de chemins.

$\mathsf{T}_{\text{DCN}}(\mathcal{N})$ est donc $(0, 1)$ -fermé, $(1, 1)$ -fermé et $(2, 1)$ -fermé. Nous concluons que $\mathsf{T}_{\text{DCN}}(\mathcal{N})$ est fermé par 3-cohérence forte. \square

En particulier, nous mettons en avant la cohérence d'arcs qui est la forme de cohérence locale des RCD qui nous intéresse particulièrement, puisqu'elle est la plus utilisée en pratique.

Corollaire 6 Soit \mathcal{N} un RCQ \diamond -fermé. $\mathsf{T}_{\text{DCN}}(\mathcal{N})$ est GAC.

Dans le cadre des RCD, nous savons que toute instanciation complète est une solution si et seulement si elle vérifie la cohérence d'arcs généralisée. Du corollaire 6, nous pouvons donc déduire que si \mathcal{N} est un scénario \diamond -fermé alors $\mathsf{T}_{\text{DCN}}(\mathcal{N})$ est un RCD atomique cohérent. Cependant, notre approche consiste à transformer le RCQ en RCD afin d'appliquer les techniques d'inférences des RCD. Il est donc plus intéressant de montrer la propriété dans ce sens.

Proposition 19 Soit \mathcal{N} un RCQ. Si $\mathsf{T}_{\text{DCN}}(\mathcal{N})$ est fermé par GAC alors \mathcal{N} est \diamond -fermé.

Preuve Soient $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ et $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C}) = \text{T}_{\text{DCN}}(\mathcal{N})$ un RCD fermé par GAC. Posons $v_i, v_j, v_k \in \mathcal{V}$ avec $1 \leq i, j, k \leq n$.

Dans un premier temps, posons $i < k < j$. Puisque $\text{T}_{\text{DCN}}(\mathcal{N})$ est fermé par GAC, pour toute relation $r_{ij} \in D_{x_{ij}}$, $\exists \tau \in c_{(x_{ij}, x_{ik}, x_{kj})} \cap \pi(\{x_{ij}, x_{ik}, x_{kj}\})$ telle que $\tau[x_{ij}] = r_{ij}$. Par conséquent, il existe un tuple (r_{ij}, r_{ik}, r_{kj}) tel que $r_{ik} \in D_{x_{ik}}$ et $r_{kj} \in D_{x_{kj}}$. Par la transformation, puisque $r_{ik} \in Q_{(v_i, v_k)}$ et $r_{kj} \in Q_{(v_k, v_j)}$, alors $r_{ij} \in Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$. Puisque cela est vrai pour tout $r_{ij} \in D_{x_{ij}}$, alors $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$, pour tout $1 \leq i < k < j \leq n$. De même, puisque $Q_{(v_i, v_j)}^{-1} \subseteq (Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)})^{-1}$, alors $Q_{(v_j, v_i)} \subseteq Q_{(v_j, v_k)} \diamond Q_{(v_k, v_i)}$. Par conséquent, $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$, pour tout $1 \leq j < k < i \leq n$.

Par un raisonnement similaire, posons maintenant $r_{ik} \in D_{x_{ik}}$ avec $i < j < k$. $\text{T}_{\text{DCN}}(\mathcal{N})$ est fermé par GAC, il existe donc $r_{ij} \in D_{x_{ij}}$ et $r_{kj} \in D_{x_{kj}}$, tel que $(r_{ij}, r_{ik}, r_{kj}) \in c_{(x_{ij}, x_{ik}, x_{kj})}$. Par la transformation, nous savons que $(r_{ij}, r_{ik}, r_{kj}) \in \text{TC}$ et par l'inverse de la composition faible nous pouvons déduire $(r_{ik}, r_{ij}, r_{kj}^{-1})$. Puisque $r_{ij} \in Q_{(v_i, v_j)}$ et $r_{kj}^{-1} \in Q_{(v_j, v_k)}$, nous avons $r_{ik} \in Q_{(v_i, v_j)} \diamond Q_{(v_j, v_k)}$. Ce raisonnement étant vrai pour tout $r_{ik} \in D_{x_{ik}}$, donc $Q_{(x_i, x_k)} \in Q_{(x_i, x_j)} \diamond Q_{(x_j, x_k)}$. Par la définition de l'inverse d'une relation, $Q_{(x_i, x_k)}^{-1} \in Q_{(x_i, x_j)} \diamond Q_{(x_j, x_k)}^{-1}$, donc $Q_{(x_k, x_i)} \in Q_{(x_k, x_j)} \diamond Q_{(x_j, x_i)}$. Par conséquent, $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$, pour tout $1 \leq i < j < k \leq n$ et pour tout $1 \leq k < j < i \leq n$.

Toujours par le même procédé, en nous appuyant sur $r_{kj} \in D_{x_{kj}}$ avec $i < j < k$, nous pouvons montrer que $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$, pour tout $1 \leq j < i < k \leq n$ et pour tout $1 \leq k < i < j \leq n$.

Supposons maintenant que $i = j < k$. Les relations sont donc $Q_{(v_i, v_j)} = Q_{(v_j, v_i)} = \{\text{id}\}$. De plus, nous savons que pour tout $r \in \mathcal{B}$, $\text{id} \subseteq r \diamond r^{-1}$. Donc $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$ pour tout $1 \leq i = j < k \leq n$ et, par l'inverse, nous pouvons déduire pour tout $1 \leq k < i = j \leq n$. Il est facile de voir que cela est également vrai pour $i = k, j = k$ et $i = j = k$.

Par conséquent, $Q_{(v_i, v_j)} \subseteq Q_{(v_i, v_k)} \diamond Q_{(v_k, v_j)}$, pour tout $1 \leq i \leq j \leq k \leq n$. \mathcal{N} est donc fermé par faible composition. \square

La figure 7.1 illustre les différents liens entre les RCD et les RCQ en exploitant la propriété 19. Notamment la possibilité d'appliquer un algorithme de fermeture par GAC sur $\text{T}_{\text{DCN}}(\mathcal{N})$, afin d'obtenir un RCQ \diamond -fermé par la transformation inverse. Nous pouvons ainsi imaginer une adaptation des algorithmes de fermeture par cohérence d'arcs utilisé pour les RCD pour développer de nouveaux algorithmes de fermeture par faible composition des RCQ. Il est intéressant de voir que les algorithmes de recherche de type MAC dans le cadre des RCD permettent de maintenir la \diamond -fermeture du RCQ correspondant.

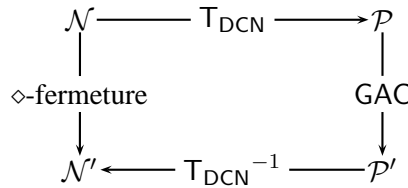


FIGURE 7.1 – La \diamond -fermeture en passant par les RCD.

7.4 Conclusion

Dans ce chapitre, nous avons présenté une transformation permettant d'exploiter la puissance des outils algorithmiques développés dans les RCD, afin d'apporter une nouvelle vision de la cohérence

locale dans les RCQ. Les domaines des RCQ étant infinis, nous nous tournons vers l'ensemble des relations atomiques, qui lui est fini. L'approche consiste à associer un RCD unique à chaque RCQ, et en particulier à associer une instanciation complète à chaque scénario. La transformation est relativement peu coûteuse en temps et en espace, d'autant que la représentation des RCD permettent de partager une même relation pour toutes les contraintes, pour la majorité des algorithmes de recherche modernes, pour lesquels il n'y a pas de modifications des contraintes en cours de recherche. Puisque la seule contrainte des RCQ est définie par la table de composition, nous avons la possibilité de ne manipuler qu'une seule relation. Les propriétés de la transformation sont intéressantes. Lorsque la \diamond -cohérence est complète pour décider de la cohérence d'un scénario, la transformation proposée est correcte et complète pour décider de la cohérence des RCQ. Cette transformation naturelle associe parfaitement la microstructure de chaque représentation, ce qui nous permet de comparer les techniques de cohérence locale. Ainsi, la fermeture par cohérence d'arcs généralisée appliquée à un RCD obtenu par notre transformation est équivalente à la fermeture par composition faible appliquée sur le RCQ initial.

Dans [WESTPHAL & WÖLFL 2009], les auteurs proposent une étude comparative de la performance des algorithmes de diverses approches pour la résolution des RCQ. Les algorithmes pour les RCQ [GANTNER *et al.* 2008], du cadre propositionnel [PHAM *et al.* 2008], et du cadre discret par notre transformation [CONDOTTA *et al.* 2006a] sont comparés sur plusieurs algèbres qualitatives (RCC8, algèbre des intervalles, *etc*). Les résultats montrent que la transformation en RCD peut s'avérer être plus performante, notamment pour des formalismes qualitatifs où la table de composition est grande. Ce qui laisse entrevoir l'intérêt d'adapter les algorithmes des RCD vers les RCQ.

Nos perspectives s'orientent évidemment vers la possibilité d'adapter de nombreuses formes de cohérences locales des RCD dans le cadre des RCQ. De nombreuses approches s'appuient sur la cohérence d'arcs, telles que les algorithmes de recherches ou la cohérence d'arcs de singletons. Des techniques d'inférences telle que l'analyse de conflits s'appuient sur le maintien de la cohérence d'arcs pour diriger la recherche ou faire de l'apprentissage. Outre l'adaptation de techniques d'inférences, nous pouvons également imaginer d'autres types de transformations, basés sur la définition des classes traitables dans les RCQ. En effet, nous voyons une facilité de manipulation d'un RCQ au travers de la transformation vers les RCD. Nous avons mentionné la possibilité de ne maintenir qu'une seule table de composition dans le RCD, cependant il peut être intéressant de pouvoir manipuler la table de compositions. Cette approche n'est pas réalisable dans le cadre des RCQ, puisque le cadre se concentre sur un formalisme qualitatif bien défini. Cependant, nous pourrions mettre en évidence des sous-ensembles de la table de composition qui ne sont pas utiles ou des sous-ensembles traitables. Nous pouvons également entrevoir une nouvelle forme de relaxation des contraintes, notamment par la relaxation de la table de composition.

Chapitre 8

Relaxation de contraintes qualitatives

Sommaire

8.1	Introduction	113
8.2	Relaxation	114
8.2.1	Transformation incomplète	114
8.2.2	Transformation inverse et complétion de l'approche	116
8.2.3	Exploiter les relations traitables	118
8.3	Expérimentations	119
8.4	Conclusion	122

8.1 Introduction

La représentation d'un problème permet d'exploiter les outils mathématiques et informatiques afin de répondre à la satisfiabilité de celui-ci. Pour de nombreux formalismes, montrer l'incohérence d'un problème conduit à des domaines connexes au problème de la satisfiabilité, tels que la détection de la source de l'incohérence et la restauration de la cohérence. La source de l'incohérence d'un problème est liée aux contraintes qui le décrivent. En nous appuyant sur la transformation des RCQ vers les RCD, nous proposons un cadre facilitant la preuve de l'incohérence de problèmes de nature temporelle ou spatiale, par la relaxation de contraintes qualitatives. Pour cela, nous relaxons localement la table de composition du formalisme qualitatif sur certaines contraintes du RCQ. Une telle manipulation n'est pas réalisable dans le cadre des RCQ. Cette approche par relaxation est incomplète pour décider de la cohérence d'un RCQ dans le cas général. Si la relaxation d'un RCQ est incohérente, alors le RCQ est incohérent. Dans le cadre qualitatif, la relation totale est la relation de cardinalité la plus grande, et elle vérifie des propriétés particulières. La relation totale étant absorbante pour l'opération de composition faible, nous faisons l'hypothèse que celle-ci peut être relaxée sans trop nous éloigner des résultats d'un algorithme de recherche complet. Nous proposons une forme de relaxation de la contrainte la plus permissive du formalisme qualitatif utilisé. La relation totale est également une relation faisant partie de nombreuses classe traitables. En nous appuyant sur ce fait, nous proposons un algorithme de recherche complet en exploitant la relaxation.

Dans ce chapitre, nous définissons une méthode de relaxation originale en exploitant le cadre des RCD. Dans un premier temps, nous définissons la transformation incomplète des RCQ vers les RCD pour décider de la cohérence d'un RCQ. Nous complétons ensuite la méthode en exploitant les classes traitables, puis nous présentons quelques résultats expérimentaux.

8.2 Relaxation

8.2.1 Transformation incomplète

La relaxation d'un réseau de contraintes est un processus consistant à modifier les contraintes afin de leur autoriser plus qu'elles ne le permettent initialement.

Définition 89 (Relaxation) Soit \mathcal{P} un réseau de contraintes. La relaxation de \mathcal{P} est un réseau de contraintes \mathcal{P}' tel que $\mathcal{P} \sqsubseteq \mathcal{P}'$.

Nous donnons ici une définition globale, valable pour les RCD ou les RCQ. La relaxation peut être exploitée dans plusieurs cadres, tels que la détection de sources d'incohérence d'un problème, la restauration de sa cohérence ou simplement pour faciliter la preuve de l'incohérence. Dans cette étude, nous utilisons la relaxation pour faciliter la preuve de l'incohérence de certains problèmes difficiles.

Propriété 13 Soient \mathcal{P} un réseau de contraintes et \mathcal{P}' une relaxation de \mathcal{P} . Si \mathcal{P}' est incohérent, alors \mathcal{P} est incohérent.

La preuve est évidente puisque, par définition de la relaxation, nous avons $\mathcal{P} \sqsubseteq \mathcal{P}'$. Par conséquent, prouver l'incohérence de \mathcal{P}' permet de prouver l'incohérence de \mathcal{P} . Notons cependant qu'exploiter la relaxation n'est pas un processus complet pour le problème de la cohérence des RCQ. Dans notre cadre, nous souhaitons faciliter la preuve de l'incohérence de problèmes de nature temporelle ou spatiale. Par conséquent, nous souhaitons appliquer la relaxation sur des RCQ construits sur divers formalismes qualitatifs. Cependant, pour qu'il y ait un intérêt à la relaxation, il faut qu'il soit plus facile de prouver l'incohérence de \mathcal{P}' , que de prouver l'incohérence de \mathcal{P} . Il est donc nécessaire de s'appuyer sur des critères prédéfinis afin d'assurer que le processus de relaxation facilite la résolution. Par exemple, en caractérisant des critères basés sur une analyse structurelle du RCQ.

L'un des critères que nous souhaitons imposer est lié à la cardinalité des relations des contraintes qualitatives. La contrainte totale (ou contrainte universelle) exprime que toute relation atomique peut être choisie pour la construction d'un scénario. En pratique, si le nombre de contraintes totales est fort, alors le problème est trivialement satisfiable puisque les contraintes sont d'une grande permissivité. Si le nombre de contraintes totales est faible, alors le problème est trivialement insatisfiable puisque celui-ci est sur-contraint. Entre les deux, il existe une catégorie de problèmes très difficiles, face auxquels les algorithmes et les technologies modernes sont impuissants [NEBEL 1997]. Et l'une des causes de cette difficulté est liée à un choix, durant la recherche, parmi toutes les relations atomiques du formalisme qualitatif. Ce qui peut occasionner un espace de recherche à évaluer considérable. Pour cette raison, nous souhaitons faciliter la résolution de ce type de problèmes, en relaxant la contrainte totale. Le second critère est de maximiser la précision de la réponse à l'incohérence, c'est-à-dire de s'approcher des résultats d'une approche complète. Intuitivement, il existe un compromis entre précision de la réponse à l'insatisfiabilité et rapidité de la preuve de l'incohérence. Il est clair que plus la relaxation appliquée est forte, plus la précision de l'approche est faible. En d'autres termes, plus il y a de contraintes permissives, plus la probabilité de prouver l'insatisfiabilité alors que le RCQ ne l'est pas augmente. Au contraire, plus la relaxation est faible et plus nous nous approchons de la méthode complète. La précision est donc plus élevée, mais il peut être difficile de prouver l'insatisfiabilité.

Par définition de la relaxation, si une contrainte est universelle, alors elle ne peut être relaxée. Pour proposer un moyen de relaxer les contraintes totales, nous pouvons relaxer la contrainte globale des formalismes qualitatifs : la table de compositions. Si les contraintes totales ne sont plus soumises à la composition faible ou soumises à une forme affaiblie, alors elles sont relaxées. Cependant, nous ne pouvons pas réaliser cette approche dans le cadre des RCQ, reposant sur un formalisme qualitatif bien défini.

Pour exploiter la relaxation dans les RCQ et pour manipuler la table de compositions sans réinventer le cadre des RCQ, nous utilisons une généralisation de notre transformation T_{DCN} des RCQ vers les RCD.

Définition 90 ($T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$) Soit $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ construit sur un formalisme \mathcal{F} de relations atomiques \mathcal{B} et $R \in 2^{\mathcal{B}}$ une relation complexe. $T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$ est un RCD défini par $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ tel que :⁷

- pour tout $v_i, v_j \in \mathcal{V}$, avec $1 \leq i < j \leq n$, si $Q_{(v_i, v_j)} \neq R$, alors la variable $x_{ij} \in \mathcal{X}$ telle que $D_{x_{ij}} = Q_{(v_i, v_j)}$ est introduite ;
- pour tout i, j, k , avec $1 \leq i < k < j \leq n$, si $x_{ij}, x_{ik}, x_{kj} \in \mathcal{X}$, alors nous introduisons une contrainte ternaire $c_{(x_{ij}, x_{ik}, x_{kj})}$ définie par $c_{(x_{ij}, x_{ik}, x_{kj})} = \{(c, a, b) \in \mathcal{B}^3 : c \in a \diamond b\}$.

La définition de $T_{\text{DCN}}^{-\text{R}}$ est très proche de la définition de T_{DCN} . Il est évident que $T_{\text{DCN}}^{-\text{R}}$ est équivalent à T_{DCN} lorsque $R = \emptyset$. Elle diffère uniquement de certaines contraintes qualitatives qui ne sont pas modélisées en tant que variable : ainsi une variable x_{ij} n'est pas modélisée par $T_{\text{DCN}}^{-\text{R}}$ car toutes les contraintes qui l'impliquent sont universelles. Par conséquent, elles ne sont pas soumises à la table de compositions mais une contrainte universelle dans le RCD. Pour ne pas être si radical, nous pourrions juste utiliser une table de composition plus permissive. Le choix de la manipulation des RCQ par la transformation vers les RCD offre l'avantage de conserver la compacité de la représentation, et d'exploiter les algorithmes de recherche et les techniques d'inférence du cadre discret. Remarquons que si nous voulons trouver l'équivalence à cette approche par relaxation directement sur le RCQ, il est nécessaire de n'appliquer aucune modification à toute relation $Q_{(v_i, v_j)}$ telle que $Q_{(v_i, v_j)} = R$, ni même de la prendre en compte dans les opérations de compositions. Or, un RCQ est un graphe complet qui doit vérifier la composition faible. Il est donc nécessaire de passer outre le formalisme qualitatif en définissant de nouvelles propriétés ou d'adapter les algorithmes de recherche spécifiquement dans ce cadre.

Dans la suite, nous supposons que la relaxation est appliquée aux RCQ normalisés qui ne sont pas trivialement incohérents, et nous supposons que l'ensemble $2^{\mathcal{B}}$ est stable par inverse.

Proposition 20 Soit \mathcal{N} un RCQ. Si \mathcal{N} est cohérent alors $T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$ est cohérent.

Preuve Considérons $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ cohérent avec $|\mathcal{V}| = n$. Puisque T_{DCN} est correcte, nous pouvons affirmer que $T_{\text{DCN}}(\mathcal{N})$ admet une solution \mathcal{I} . Posons $v_i, v_j \in \mathcal{V}$ avec $i < j$. Nous définissons $X \subseteq \mathcal{X}$ tel que $X = \{x_{ij} \in \mathcal{X} : Q_{(v_i, v_i)} \neq R\}$, et l'instanciation $\mathcal{I}_X \subseteq \mathcal{I}$, telle que $\mathcal{I}_X(x_{ij}) = \mathcal{I}(x_{ij})$ pour tout $x_{ij} \in \mathcal{X}$. Pour tout triplet $x_{ij}, x_{ik}, x_{kj} \in X^3$ avec $1 \leq i < k < j \leq n$, nous savons que $(\mathcal{I}(x_{ij}), \mathcal{I}(x_{ik}), \mathcal{I}(x_{kj})) \in \text{TC}$, par conséquent $(\mathcal{I}_X(x_{ij}), \mathcal{I}_X(x_{ik}), \mathcal{I}_X(x_{kj})) \in \text{TC}$. Toute contrainte de $T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$ est satisfaite par \mathcal{I}_X , nous concluons que $T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$ est cohérent. \square

La relaxation en passant par les RCD permet de sortir du cadre bien défini des RCQ et d'autoriser localement la relaxation de la table de compositions par une contrainte universelle dans le RCD. Pour réaliser cela, nous choisissons la relaxation de la relation totale : $T_{\text{DCN}}^{-\psi}$. Bien qu'il semble contre-intuitif de relaxer une contrainte très permissive, ce choix est motivé par des critères énoncés précédemment, mais également par plusieurs caractéristiques de cette relation. Comme énoncé précédemment, nous souhaitons nous approcher de la complétude d'un algorithme de recherche exhaustive. Pour cela, nous savons que la relation totale est l'élément absorbant pour l'opération de composition. Nous pouvons donc supposer que la relaxation de la propriété de composition pour la relation totale a un impact moins important sur la complétude, par rapport à la relaxation d'autres relations complexes. Notre hypothèse est que la contrainte totale possède la probabilité la plus faible d'apparaître dans un sous-RCQ incohérent.

7. Remarquons que la définition $T_{\text{DCN}}^{-\text{R}}$ peut être définie pour un sous-ensemble $\{R_1, \dots, R_k\} \subseteq 2^{\mathcal{B}}$ plutôt que pour une relation $R \in 2^{\mathcal{B}}$. Cependant, nous nous concentrerons dans cette étude sur la relation d'une seule relation complexe.

Par conséquent, nous minimisons les risques de relaxer une contrainte faisant partie d'un sous-RCQ incohérent.

Les figures 8.2 et 8.3 illustrent la transformation d'un RCQ (Figure 8.1) en RCD, avec et sans relaxation. Ce RCQ construit sur l'algèbre des intervalles est insatisfiable, car il impose l'ordre « v_1 avant v_2 avant v_4 », mais également « v_1 après v_3 après v_4 ». La transformation classique par T_{DCN} modélise la table de composition pour chaque triplet de variables discrètes (Figure 8.2). Pour la transformation par relaxation $T_{DCN}^{-\psi}$, la variable $x_{2,3}$ n'est pas définie. Ce que nous pouvons considérer comme : toutes contraintes ternaires qui impliquent $x_{2,3}$ sont universelles. Par conséquent la variable $x_{2,3}$ n'est pas apparente (Figure 8.3). Trivialement, les deux RCD sont insatisfiables. Cependant, l'espace de recherche à évaluer pour prouver l'incohérence de $T_{DCN}^{-\psi}(\mathcal{N})$ est plus réduit que celui de $T_{DCN}(\mathcal{N})$.

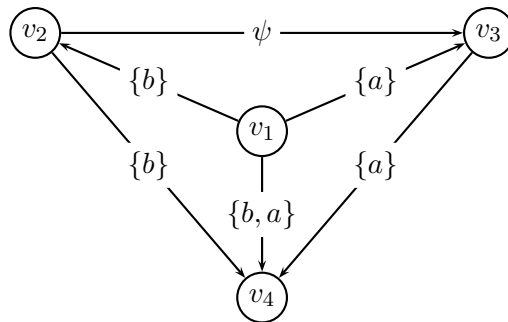


FIGURE 8.1 – RCQ \mathcal{N} insatisfiable construit sur le formalisme d'Allen

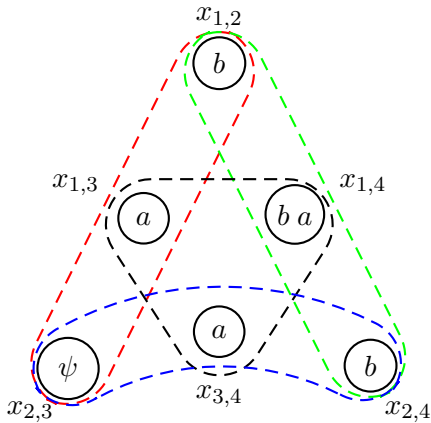


FIGURE 8.2 – $T_{DCN}(\mathcal{N})$

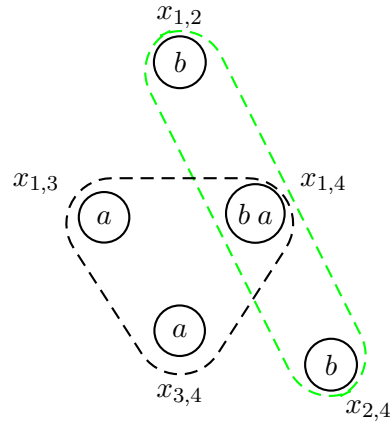


FIGURE 8.3 – $T_{DCN}^{-\psi}(\mathcal{N})$

8.2.2 Transformation inverse et complétion de l'approche

Dans la transformation T_{DCN} , nous avons défini une transformation inverse, appliquée en temps polynomiale, permettant à toute solution d'associer un scénario cohérent. Dans notre cadre, la transformation par relaxation est incomplète pour le problème de la cohérence. Par conséquent, si le RCD obtenu par T_{DCN}^{-R} est cohérent, nous ne pouvons pas décider de la cohérence du RCQ associé. Cependant, nous pouvons exploiter le travail accompli par l'algorithme de recherche sur le RCQ relaxé.

Définition 91 ($T_{\text{QCN}}^{+\text{R}}(\mathcal{P})$) Soient $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ et $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD tel que $\mathcal{P} = T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$ avec $\text{R} \subseteq 2^{\mathcal{B}}$. Soit $\mathcal{P}' \sqsubseteq \mathcal{P}$, la transformation $T_{\text{QCN}}^{+\text{R}}(\mathcal{P}')$ est le RCQ défini par $\mathcal{N}' = (\mathcal{V}, \mathcal{Q}')$ tel que :

- pour tout $v_i \in \mathcal{V} : Q'_{(v_i, v_i)} = \{\text{id}\}$;
- pour tout $v_i, v_j \in \mathcal{V}$ avec $i < j$: si $Q_{(v_i, v_j)} \in \text{R}$ alors $Q'_{(v_i, v_j)} = Q_{(v_i, v_j)}$, sinon $Q'_{(v_i, v_j)} = D_{x_{ij}}$;
- pour tout $v_i, v_j \in \mathcal{V}$ avec $i > j$: $Q'_{(v_j, v_i)} = Q'_{(v_i, v_j)}^{-1}$.

Pour tout \mathcal{N} un RCQ, la transformation $T_{\text{QCN}}^{+\text{R}}$ s'applique donc à tout sous-RCD de $T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$. En particulier, $T_{\text{QCN}}^{+\text{R}}$ s'applique au sous-RCD atomique de $T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$, que nous pouvons associer à une solution.

Proposition 21 Soient \mathcal{N} un RCQ et \mathcal{I} une solution de $\mathcal{P} = T_{\text{DCN}}^{-\text{R}}(\mathcal{N}) : T_{\text{QCN}}^{+\text{R}}(\mathcal{P}_{|\mathcal{I}}) \sqsubseteq \mathcal{N}$.

La preuve est triviale et suit la définition 91. Cette proposition donne l'idée de l'approche utilisée pour compléter notre approche. Puisque nous pouvons affirmer que si $T_{\text{QCN}}^{+\text{R}}(\mathcal{P}_{|\mathcal{I}})$ est cohérent alors \mathcal{N} l'est également. Par conséquent, nous pouvons exploiter les solutions de $T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$, afin de définir une approche complète. Nous définissons un algorithme permettant de répondre à la cohérence d'un RCQ en utilisant la relaxation.

L'algorithme 13 présente la recherche de la cohérence d'un RCQ en passant par la relaxation définie. À partir d'un RCQ \mathcal{N} , nous définissons un RCD \mathcal{P} par la transformation $T_{\text{DCN}}^{-\text{R}}$. La première étape consiste alors à déterminer une solution de \mathcal{P} . Cette approche peut être réalisée par un algorithme de type *recherche en profondeur d'abord* DFS. S'il n'existe pas de solution, alors \mathcal{P} est incohérent, et donc \mathcal{N} est incohérent. Dans le cas contraire, nous exploitons la solution de $T_{\text{DCN}}^{-\text{R}}$ afin de construire un nouveau RCQ par la transformation $T_{\text{QCN}}^{+\text{R}}$. Si ce RCQ est cohérent, alors \mathcal{N} l'est aussi. Sinon, nous cherchons une nouvelle solution de \mathcal{P} . Pour cela, nous supposons que l'algorithme DFS retourne les solutions de \mathcal{P} incrémentalement, à partir d'une solution qui lui est fournie. Celui-ci permet donc d'énumérer l'ensemble des solutions de $T_{\text{DCN}}^{-\text{R}}\mathcal{N}$.

Algorithme 13: Algorithme de recherche par relaxation

Entrées : $\mathcal{N} = (\mathcal{V}, \mathcal{Q})$ un RCQ

Sorties : Oui si \mathcal{N} est cohérent, Non sinon

```

1 début
2    $\mathcal{P} = T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$ 
3   solution  $\leftarrow$  DFS( $\mathcal{P}, \emptyset$ )
4   tant que (solution existe) faire
5     si (Recherche( $T_{\text{QCN}}^{+\text{R}}(\text{Solution})$ ) =  $\top$ ) alors
6       retourner Oui
7     solution  $\leftarrow$  DFS( $\mathcal{P}, \text{Solution}$ )
8   retourner Non

```

Proposition 22 L'algorithme 13 est correct et complet.

Preuve Si $T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$ est insatisfiable, alors \mathcal{N} est insatisfiable puisque $T_{\text{DCN}}^{-\text{R}}(\mathcal{N}) \sqsubseteq \mathcal{N}$. Si pour toute solution \mathcal{I} de $T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$, le RCQ $T_{\text{QCN}}^{+\text{R}}(\mathcal{I})$ est incohérent, alors \mathcal{N} est incohérent. Posons maintenant \mathcal{I} une solution de $\mathcal{P} = T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$ et \mathcal{S} un scénario cohérent de $T_{\text{QCN}}^{+\text{R}}(\mathcal{P}_{|\mathcal{I}})$. $T_{\text{QCN}}^{+\text{R}}(\mathcal{P}_{|\mathcal{I}}) \sqsubseteq \mathcal{N}$ par la proposition 21, donc \mathcal{S} est également un scénario cohérent de \mathcal{N} . L'algorithme est donc complet.

Enfin, supposons que \mathcal{N} admette un scénario cohérent \mathcal{S} . La transformation T_{DCN} est correcte pour le problème de la cohérence, donc il existe une solution \mathcal{I} de $\mathcal{P} = T_{\text{DCN}}(\mathcal{N})$ associée à \mathcal{S} . Par conséquent, il existe une solution \mathcal{I}' de $\mathcal{P}' = T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$ associée à \mathcal{S} qui est une restriction de \mathcal{I} . Par définition, $\mathcal{S} \subseteq \mathcal{N}$ et donc $\mathcal{S} \subseteq T_{\text{QCN}}^{+\text{R}}(\mathcal{P}_{|\mathcal{I}'})$. \mathcal{S} est donc un scénario cohérent de $T_{\text{QCN}}^{+\text{R}}(\mathcal{I}')$. L'algorithme est donc correct. \square

Cette approche peut être intéressante pour prouver l'incohérence, cependant il est facile de voir que si le RCQ initial est cohérent, alors l'arbre de recherche développé par notre algorithme est au moins aussi grand que l'arbre de recherche développé par un algorithme de recherche classique. En effet, supposons un RCQ \mathcal{N} cohérent. Posons \mathcal{I} une solution de $T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$. Si \mathcal{I} n'est pas localement cohérente dans $T_{\text{DCN}}(\mathcal{N})$, \mathcal{I} ne peut être associée à un scénario cohérent de \mathcal{N} . Par conséquent, l'évaluation de $T_{\text{QCN}}^{+\text{R}}(\mathcal{I})$ est inutile, et ce sous-réseau de \mathcal{N} n'aurait pas été exploré par une approche classique. L'approche complète ne semble pas efficace pour prouver la cohérence. Cependant, nous pouvons exploiter les propriétés de certaines algèbres pour l'améliorer.

8.2.3 Exploiter les relations traitables

Pour faciliter la tâche de résolution des RCQ par la méthode complète en utilisant la relaxation, nous pouvons exploiter les propriétés de certaines algèbres qualitatives, et notamment des classes traitables. Dans la suite, nous considérons donc les formalismes qualitatifs pour lesquels il existe une classe de relations traitables. Ainsi, nous pouvons décider de la cohérence d'un RCQ de l'une de ces classes par un procédé polynomial dans la taille du RCQ. Nous imposons à la classe traitable de contenir au moins l'ensemble des relations singleton. Par exemple, nous pouvons considérer le cas de l'algèbre des intervalles et de la classe des relations convexes \mathcal{C} , car $\{r\} \in \mathcal{C}$ pour tout $r \in \mathcal{B}_{\text{int}}$. Nous appellerons chaque relation complexe d'une classe traitable une *relation traitable*. Pour simplifier, nous supposons que la méthode de fermeture par faible composition permet de répondre à la cohérence des RCQ construits sur ces relations traitables.

Proposition 23 Soient \mathcal{N} un RCQ, \mathcal{I} une solution de $\mathcal{P} = T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$. Si \mathcal{R} est une traitable et $T_{\text{QCN}}^{+\text{R}}(\mathcal{P}_{\mathcal{I}})$ est fermé par composition faible, alors \mathcal{N} est cohérent.

Preuve La preuve est immédiate puisque $T_{\text{QCN}}^{+\text{R}}(\mathcal{P}_{\mathcal{I}})$ est composé de \mathcal{R} et de relations singletons. Par conséquent, $T_{\text{QCN}}^{+\text{R}}(\mathcal{I})$ est un RCQ construit sur un ensemble de relation traitable pour lequel la méthode de \diamond -fermeture est complète pour le problème de la cohérence. Donc si $T_{\text{QCN}}^{+\text{R}}(\mathcal{I})$ est \diamond -fermé, alors il est cohérent. Donc \mathcal{N} est cohérent. \square

En appliquant la relaxation à un ensemble de relations traitables, nous pouvons ainsi faciliter la vérification de la cohérence par un traitement polynomial. Ainsi, pour toute solution \mathcal{I} de $T_{\text{DCN}}^{-\text{R}}(\mathcal{N})$, nous pouvons décider de la cohérence de $T_{\text{QCN}}^{+\text{R}}(\mathcal{P}_{\mathcal{I}})$ en temps polynomial dans sa taille. Dans le cas de l'algèbre des intervalles, nous pouvons exploiter les classes des relations convexes, pointisables et ORD-Horn qui contiennent les relations singletons. En particulier, la relation totale ψ fait partie de ces classes de relations traitables.

La figure 8.4 présente une schématisation de l'algorithme de recherche complet exploitant la relaxation de la relation totale. Un RCQ est fourni en entrée et le résultat est \top (pour cohérent) ou \perp (pour incohérent). Chaque losange représente une décision caractérisée par une question, et chaque noeud est un état du problème. Le RCQ en entrée est transformé et relaxé par $T_{\text{DCN}}^{-\psi}$ en un RCD. Si celui-ci ne possède pas de solution, le RCQ est incohérent. Dans le cas contraire, nous utilisons la solution \mathcal{I} du RCQ pour obtenir un nouveau RCQ par $T_{\text{QCN}}^{+\psi}$. La cohérence de ce RCQ peut être décidée par \diamond -fermeture. S'il est cohérent, le RCQ initial est cohérent. Sinon, il est incohérent, et nous effectuons la demande d'une

nouvelle solution pour le RCD. Remarquons qu'en exploitant la propriété 19 du chapitre 7, nous savons que si $T_{\text{DCN}}^{-\psi}(\mathcal{N})$ est GAC, alors \mathcal{N} est \diamond -fermé. Par conséquent, nous pouvons éviter l'utilisation de la transformation $T_{\text{QCN}}^{+\psi}$, en intégrant les contraintes relaxées dans $T_{\text{DCN}}^{-\psi}(\mathcal{N})$ et en vérifiant s'il est GAC. Si c'est le cas, alors le RCQ associé est cohérent.

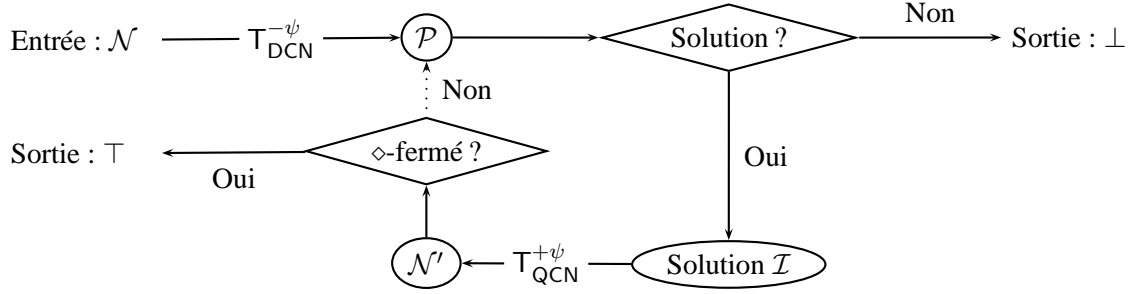


FIGURE 8.4 – Schema général de résolution d'un RCQ par la relaxation de la relation totale.

8.3 Expérimentations

Pour étudier l'intérêt pratique de notre approche, quelques expérimentations sur la détection de l'insatisfiabilité ont été menées sur la transformation $T_{\text{DCN}}^{-\psi}$. Nos expérimentations se limitent à des RCQ générés aléatoirement dû à l'absence de problèmes qualitatifs structurés. Ils sont générés sur le modèle $A(n, d, s)$ décrit par Nebel [NEBEL 1997]. Trois paramètres décrivent les caractéristiques d'un RCQ : n le nombre de variables du RCQ, d le degré moyen des variables du RCQ et s la taille moyenne des relations du RCQ qui ne sont pas totales. Dans notre cadre, nous maintenons fixe le nombre de variables à $n = 50$ et la taille moyenne des relations à $s = 3, 25$ et $s = 6, 5$, puis nous faisons varier le degré moyen de $d = 0,01$ à $d = 0,99$ avec un pas de $0,01$. Pour tout triplet de paramètres $A(n, d, s)$, 100 réseaux de contraintes ont été générés. Les plateformes QAT [CONDOTTA *et al.* 2006b] et Abscon [LECOUTRE & TABARY 2006] sont utilisées conjointement pour transformer chaque RCQ en RCD puis rechercher la satisfiabilité du RCD résultant de la transformation. Le temps de transformation et résolution a été limité à 20 minutes.

Les expérimentations visent à montrer l'efficacité de la réponse à la insatisfiabilité et la précision de notre approche, c'est-à-dire le taux de problèmes insatisfiables détectés par rapport à une approche classique par recherche exhaustive. La première remarque que nous pouvons faire est qu'un RCQ généré par le modèle $A(n, d, s)$ n'est pas nécessairement fermé par faible composition. Appliquer la fermeture par faible composition en prétraitements sur ces RCQ peut conduire à réduire certaines relations totales. Nous comparons donc trois approches : l'approche classique avec la transformation T_{DCN} , et deux relaxations avec la transformation $T_{\text{DCN}}^{-\psi}$, que nous nommerons *relaxation forte* et *relaxation faible*. À partir d'un RCQ \mathcal{N} , la relaxation forte est définie par $T_{\text{DCN}}^{-\psi}(\mathcal{N})$ et la relaxation faible par $T_{\text{DCN}}^{-\psi}(\diamond(\mathcal{N}))$ où \diamond est la méthode de fermeture par composition faible. Cela nous permet de visualiser l'impact de la fermeture par faible composition sur la relaxation appliquée, voire permettre de choisir en relaxation faible et forte. Intuitivement, la relaxation forte est au moins aussi forte que la relaxation faible. Si le RCQ initial est fermé par composition faible, les deux relaxations sont identiques.

La première série d'expérimentations a été menée sur des problèmes faciles. La figure 8.3 présente le nombre de RCQ détectés insatisfiables par les trois approches pour une cardinalité moyenne des relations

de 3.25 (à gauche) et 6.5 (à droite). Nous pouvons remarquer que la précision de la relaxation faible est très proche de l'approche classique, et la précision de la relaxation forte est évidemment plus faible.

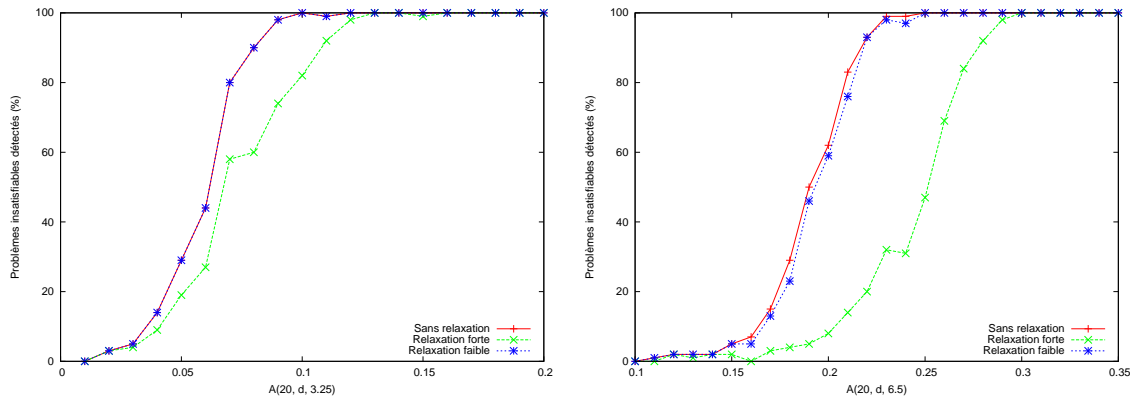


FIGURE 8.5 – Nombre de RCQ insatisfiables pour $A(50, d, 3.25)$ et $A(50, d, 6.5)$

La figure 8.3 présente les temps résolution des problèmes prouvés insatisfiables communément par la méthode complète et la méthode par relaxation faible. Malgré le faible écart entre les deux approches, nous pouvons néanmoins entrevoir un gain en temps pour la relaxation faible.

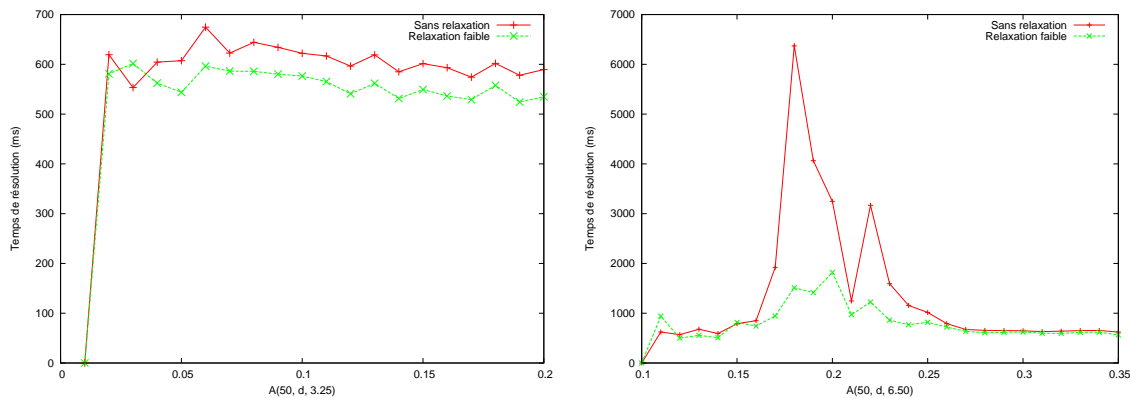


FIGURE 8.6 – Temps de résolution des RCQ insatisfiables pour $A(50, d, 3.25)$ et $A(50, d, 6.5)$

Ces expérimentations succinctes ne peuvent donner qu'une idée de l'efficacité en précision ou en temps de résolution sur les problèmes faciles. Pour parfaire notre protocole expérimentale, nous avons opter pour des paramètres de génération de RCQ aléatoires qui sont plus difficiles à résoudre pour une approche de résolution classique, sans relaxation. 100 réseaux de contraintes qualitatives ont été générées sur le modèle $A(50, 0.65, 9.75)$. Un algorithme de recherche du type MAC, pour lequel le temps de résolution a été limité à 20 mn, a permis de montrer que 35 RCQ sont satisfiables, que 33 sont insatisfiables, et l'algorithme n'a pas répondu dans le temps imparti pour 32 d'entre elles. Le tableau 8.1 décrit un comparatif des résultats d'une résolution de type MAC, de la relaxation forte et de la relaxation faible sur les 65 problèmes qui n'ont pas été prouvés satisfiables. Celui-ci présente, d'une part, le nombre de RCQ prouvés satisfiables et insatisfiables, et le nombre de RCQ pour lequel la méthode utilisée ne permet pas de répondre dans le temps imparti. D'autre part, ce tableau montre le temps de réponses pour les RCQ prouvés insatisfiables par l'approche classique et par la relaxation. Tout d'abord, nous pouvons

constater que les approches par relaxation faible et forte permettent de montrer l'insatisfiabilité de 16 RCQ supplémentaires, par rapport à l'approche sans relaxation. De plus, sur les 33 RCQ prouvés insatisfiables avec et sans relaxations, le temps de réponse des algorithmes et divisés par quatre. Remarquons que les relaxations faible et forte ont des résultats très proches. Ceci est dû au fait que le RCQ obtenu par la fermeture par faible composition est très proche du RCQ initial. L'approche semble intéressante pour montrer l'incohérence de certains RCQ difficiles à résoudre sans relaxation.

Méthodes	Sans relaxation	Relaxation forte	Relaxation faible
Nombre de RCQ	65	65	65
RCQ satisfiable	0	0	0
RCQ insatisfiable	33	49	49
RCQ sans réponse	32	16	16
Temps total en secondes	14,572	3,765	4,016
Temps moyen en secondes	441,5	114,1	121,7

TABLE 8.1 – 65 RCQ générées sur le modèle $A(50, 0.65, 9.75)$ non prouvés satisfiables

Enfin, le tableau 8.2 présente un comparatif des résultats obtenus sur un large panel de RCQ générés aléatoirement, pour l'approche sans relaxation, notée « SR », et avec relaxation faible. 100 RCQ ont été générés aléatoirement sur le modèle $A(50, d, s)$. Sont présentés dans ce tableau les RCQ difficiles pour chaque type de paramètres, dont au moins un RCQ a été résolu par les deux approches et au moins un RCQ est prouvé insatisfiable. 45 paramètres ont donc été retenus, et sur chaque paramètre, nous nous concentrons sur les RCQ qui n'ont pas été prouvés satisfiables. Trois critères sont comparés et les résultats globaux pour chaque critère apparaissent sur la ligne *Total*. Le premier critère correspond au nombre de RCQ non résolus, c'est-à-dire pour lesquels l'algorithme de recherche n'a pas répondu avant le temps imparti de 20 minutes. Nous présentons les résultats pour la Relaxation Faible « RF », l'approche Sans Relaxation « SR », ainsi que la différence entre les deux « RF-SR ». Nous pouvons ainsi constater que la relaxation faible permet de résoudre 90 RCQ de plus que l'approche sans relaxation. Ce gain est essentiellement dû aux problèmes difficiles pour ($s = 0.70, 0.75$ et 0.80). Le deuxième critère est le nombre de problèmes prouvés insatisfiables pour « RF » et « SR ». Nous comparons également la différence entre les deux approches (« RF-SR »), le nombre de RCQ en commun prouvés insatisfiables par les deux approches (« $RF \cap SR$ »), et la différence entre les résultats de la relaxation faible et le nombre de problèmes communs (« $RF - RF \cap SR$ »). Remarquons que l'approche par relaxation est moins intéressante globalement, puisqu'elle détecte 7 problèmes insatisfiables de moins que l'approche classique. Cette perte est dû principalement aux problèmes les plus faciles ($s = 0.55, 0.60$ et 0.65). Le nombre de problèmes prouvés insatisfiables communément par les deux approches montre un résultat intéressant : tout RCQ prouvé insatisfiable par « SR » l'est également par « RF » pour les problèmes difficiles ($s = 0.70, 0.75$ et 0.80). En d'autres termes, la relaxation appliquée aux RCQ facilite la preuve de l'incohérence sans l'altérer, pour ces catégories. Enfin, le dernier critère est le temps moyen de résolution (en secondes) des RCQ prouvés communément insatisfiables pour « RF » et « SR ». Là encore, le rapport des temps des deux approches « RF/SR » montre l'avantage d'utiliser la relaxation pour les problèmes difficiles, puisque « RF » est approximativement deux fois plus rapide que « SR » lorsque le temps moyen de résolution pour « SR » est de plus de dix secondes. Remarquons particulièrement les résultats pour la catégorie de problèmes difficiles $s = 0.80$ pour lesquels le gain en temps substantiel. Ce comparatif confirme l'utilité d'utiliser la relaxation de la relation totale pour prouver l'incohérence des RCQ difficiles en démontrant la précision de l'approche et l'efficacité de la relaxation pour la preuve de l'incohérence. Cependant, il existe toujours de nombreux problèmes très difficiles pour lesquelles la relaxation n'est pas assez forte.

En ce qui concerne l'approche complète par relaxation, les résultats sont très insatisfaisants puisque aucun RCQ de nos expérimentations n'a été prouvé satisfiable. Pour un RCQ de 50 variables le RCD obtenu par transformation T_{DCN} contient 1225 variables. Même si la moitié a été omise par $T_{DCN}^{-\psi}$, la solution obtenue pour le RCQ relâché porte sur 600 variables. Par conséquent, le problème vient ici de la difficulté à corriger cette instanciation si celle-ci n'est pas globalement cohérente. Après avoir inséré les contraintes relâchées par $T_{QCN}^{+\psi}$, la \diamond -cohérence est vérifiée. Si le RCQ n'est pas \diamond -cohérent, nous analysons la solution suivante, sans apprendre de l'échec de la \diamond -cohérence. Il résulte donc un simple retour-arrière sur une des 600 variables, puis l'approche est réitérée. Cependant, l'erreur peut provenir de la racine de l'arbre de recherche développé, et il est trop difficile de remonter à cette cause dans le temps imparti avec un algorithme de recherche classique, sans apprentissage et sans adapter l'approche.

8.4 Conclusion

Dans ce chapitre, nous avons présenté un cadre général de relaxation pour le raisonnement qualitatif. Cette approche est motivée par la détection des sources d'incohérence, la réparation de celles-ci ou simplement pour faciliter la preuve de l'incohérence. En nous appuyant sur la transformation vers les RCD, nous avons défini la relaxation de la table de compositions du formalisme qualitatif utilisé. Cette forme de relaxation permet donc de relaxer la propriété de composition sur tout type de relations du RCQ. En particulier, nous choisissons de relaxer la table de compositions sur la contrainte totale qui, par définition, est la contrainte la plus permissive. La précision de l'approche est très intéressante pour les RCQ difficiles, et le gain en temps est non négligeable. Un algorithme de résolution complet a été proposé pour le problème de la cohérence des RCQ, tout en intégrant la relaxation. L'utilisation des classes de relations traitables permet, à partir d'une solution de la relaxation, de vérifier en temps polynomial par la fermeture par faible composition si celle-ci est globalement cohérente dans le RCQ initial. Cependant, cette approche présente de très mauvais résultats pratiques en précision de détection de satisfiabilité et en temps. L'algorithme de recherche classique ne permet pas d'apprendre des erreurs rencontrées par la fermeture par faible composition, et il est donc difficile de revenir sur des choix non pertinents. Il serait ainsi intéressant de greffer des améliorations spécifiques à l'algorithme de recherche classique, notamment la pondération des contraintes et l'apprentissage.

En perspective à ces travaux, nous voyons la possibilité d'exploiter différemment la relaxation, par la détection ou la réparation d'incohérence. En adaptant cette approche, nous pouvons également proposer une manipulation moins radicale de la table de composition en l'affaiblissant. Il serait également intéressant d'étudier le comportement des algorithmes d'inférence face à la relaxation de contraintes particulières. Enfin, il serait intéressant d'étudier l'approche complète, afin d'améliorer les résultats en greffant certaines améliorations tels que l'analyse de conflits ou l'apprentissage.

s	d	Non résolus			Insatisfiables					Temps moyen $SR \cap RF$		
		SR	RF	SR-RF	SR	RF	RF-SR	$SR \cap RF$	$RF - SR \cap RF$	SR	RF	RF/SR
0.55	0.20	0	0	0	3	1	-2	1	0	5	10	1.96
0.55	0.25	1	1	0	58	37	-21	37	0	13	6	0.47
0.55	0.30	0	0	0	57	56	-5	56	0	11	6	0.57
0.55	0.35	0	0	0	15	15	0	15	0	3	5	1.4
0.60	0.30	10	10	0	69	38	-31	38	0	52	18	0.34
0.60	0.35	2	2	0	86	84	-2	84	0	21	7	0.35
0.60	0.40	0	0	0	74	74	0	74	0	7	5	0.70
0.60	0.45	0	0	0	40	40	0	40	0	4	5	1.31
0.60	0.50	0	0	0	17	17	0	17	0	3	5	1.62
0.60	0.55	0	0	0	3	3	0	3	0	3	4	1.42
0.65	0.35	24	24	0	37	16	-21	16	0	79	18	0.23
0.65	0.40	9	6	3	90	81	-9	78	3	118	46	0.39
0.65	0.45	0	0	0	97	97	0	97	0	40	10	0.26
0.65	0.50	0	0	0	94	94	0	94	0	8	5	0.65
0.65	0.55	0	0	0	91	91	0	91	0	5	5	1.05
0.65	0.60	0	0	0	67	67	0	67	0	3	5	1.49
0.65	0.65	0	0	0	59	59	0	59	0	3	5	1.58
0.65	0.70	0	0	0	34	34	0	34	0	3	5	1.71
0.65	0.75	0	0	0	12	12	0	12	0	3	6	1.87
0.65	0.80	0	0	0	13	13	0	13	0	3	6	1.99
0.65	0.85	0	0	0	3	3	0	3	0	3	5	1.49
0.65	0.95	0	0	0	1	1	0	1	0	3	6	2.13
0.70	0.45	67	63	4	25	27	2	23	4	364	157	0.43
0.70	0.50	15	6	9	84	93	9	84	9	313	95	0.30
0.70	0.55	0	0	0	99	99	0	99	0	120	30	0.25
0.70	0.60	0	0	0	98	98	0	98	0	36	13	0.37
0.70	0.65	0	0	0	99	99	0	99	0	14	8	0.60
0.70	0.70	0	0	0	99	99	0	99	0	6	7	1.09
0.70	0.75	0	0	0	99	99	0	99	0	4	7	1.53
0.70	0.80	0	0	0	95	95	0	95	0	4	6	1.73
0.70	0.85	0	0	0	91	91	0	91	0	3	6	1.87
0.70	0.90	0	0	0	87	87	0	87	0	3	6	1.99
0.70	0.95	0	0	0	75	75	0	75	0	3	7	2.14
0.75	0.60	75	66	9	24	33	9	24	9	649	280	0.43
0.75	0.65	31	17	14	68	82	14	68	14	590	224	0.38
0.75	0.70	4	0	4	95	99	4	95	4	285	122	0.43
0.75	0.75	0	0	0	99	99	0	99	0	123	54	0.44
0.75	0.80	0	0	0	99	99	0	99	0	57	28	0.50
0.75	0.85	0	0	0	99	99	0	99	0	24	19	0.78
0.75	0.90	0	0	0	99	99	0	99	0	13	14	1.05
0.75	0.95	0	0	0	99	99	0	99	0	9	11	1.34
0.80	0.80	97	91	6	2	8	6	2	6	1132	479	0.42
0.80	0.85	85	72	13	14	27	13	14	13	669	386	0.58
0.80	0.90	50	29	21	49	70	21	49	21	630	434	0.69
0.80	0.95	15	8	7	84	89	5	82	7	514	352	0.69
<i>Total</i>		485	395	90	2802	2798	-7	2708	90	5958	2940	0.49

TABLE 8.2 – Comparaison de la méthode de résolution avec et sans relaxation sur un panel de RCQ

Chapitre 9

ϕ -Substituabilité

Sommaire

9.1	Introduction	125
9.2	Substituabilité basée sur la cohérence d’arcs de singletons	126
	9.2.1 Définition et étude de la substituabilité au voisinage	126
	9.2.2 Substituabilité au prétraitement	127
	9.2.3 Recherche dynamique de la substituabilité	130
9.3	Expérimentations	133
	9.3.1 Intégration statique	133
	9.3.2 Intégration dynamique	136
9.4	Conclusion	138

9.1 Introduction

Lors de nos différentes études des transformations, nous avons cherché à faire le lien entre classes traitables des RCQ, notamment les relations convexes, et les RCD. Puisque la fermeture par faible composition est complète pour la classe des convexes, cette approche avait pour but de caractériser une classe de problèmes, potentiellement différentes de la classe traitable *row-convex* des RCD [VAN BEEK & DECHTER 1995], pour laquelle la cohérence d’arcs généralisée est suffisante pour décider de la cohérence du RCD. Nous avons émis l’hypothèse que la notion de substituabilité [FREUDER 1991] pouvait nous conduire vers cette classe particulière. La substituabilité peut prendre deux formes : complète ou locale. La substituabilité complète est très difficile à détecter, puisque pour vérifier la substituabilité complète d’une valeur par rapport à une autre, il faut énumérer l’ensemble des solutions. La substituabilité au voisinage est une forme de substituabilité locale, permettant de spécifier que toutes les valeurs en support avec une première valeur d’une variable, sont également supports avec une seconde valeur de cette variable. Dans ce cas, la première est *substituable* par la seconde, c’est-à-dire qu’elle peut localement être remplacée par la seconde. Puisque notre approche consiste à appliquer la cohérence d’arcs généralisée, nous avons défini la ϕ -substituabilité.

En nous appuyant sur les algorithmes de recherches modernes, nous avons défini une forme de substituabilité sur un cadre affectation/propagation. Une valeur est affectée à une variable, puis elle est propagée à travers les contraintes par ϕ , ce qui nous permet d’obtenir un RCD. Nous effectuons à nouveau ce traitement sur une autre valeur de cette variable afin d’obtenir un second RCD. En comparant la structure de ceux-ci, nous pouvons affirmer qu’à l’exception de la variable de référence, si le premier RCD est inclus dans le second, alors la première valeur est substituable à la seconde. La détection de

cette forme de symétrie faible nous conduit à développer deux approches. L'une est statique, réalisée en prétraitement et intégrée à l'algorithme SAC exploitant déjà le cadre affectation/propagation. Nous utilisons simplement le travail effectué par SAC pour vérifier la substituabilité. L'approche préservée ne préserve pas l'ensemble des solutions du RCD, mais elle consiste à réduire l'espace de recherche à évaluer. L'autre est dynamique, réalisée pendant la recherche et intégrée à l'algorithme MAC, s'appuyant également sur le cadre affectation/propagation. Cette approche est complète, puisqu'elle empêche l'évaluation de sous-espaces de recherche prouvés incohérent pour une valeur substituée.

Dans ce chapitre, nous définissons la notion de ϕ -substituabilité. Nous proposons deux approches, l'une intégrée à SAC, l'autre intégrée à MAC. Nous réalisons une étude expérimentale sur les deux approches afin de conclure de l'efficacité de l'approche.

9.2 Substituabilité basée sur la cohérence d'arcs de singletons

9.2.1 Définition et étude de la substituabilité au voisinage

La substituabilité et l'interchangeabilité sont des caractéristiques structurelles des réseaux de contraintes pouvant être exploitées dans les algorithmes de recherche. Démontrer la substituabilité est difficile [FREUDER 1991], cependant nous souhaitons exploiter un affaiblissement de ceux-ci. Il existe deux formes de substituabilité : *complète* ou *locale*.

Définition 92 (Substituabilité complète) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, $x \in \mathcal{X}$ et deux valeurs $v, v' \in D_x$. v est complètement substituable par v' si et seulement si pour toute solution \mathcal{I} de \mathcal{P} telle que $\mathcal{I}(x) = v$, il existe une solution \mathcal{I}' de \mathcal{P} telle que $\mathcal{I}'(x) = v'$ et $\mathcal{I}'(x') = \mathcal{I}(x')$ pour tout $x' \in \mathcal{X}$ et $x' \neq x$.

Définition 93 (Interchangeabilité complète) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, $x \in \mathcal{X}$ et $v, v' \in D_x$. v et v' sont complètement interchangeables si et seulement si v est complètement substituable par v' et v' est complètement substituable par v .

Cette définition donne une idée de la difficulté à déterminer l'interchangeabilité ou la substituabilité complète de deux valeurs. Dans un RCD $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$, pour tout $x \in \mathcal{X}$, montrer que $v, v' \in D_x$ sont complètement substituables nécessite d'énumérer l'ensemble des solutions de \mathcal{P} contenant (x, v) et (x, v') , et de comparer la restriction de chacune avec toutes les autres. Cependant, un affaiblissement permet de définir une forme de substituabilité et d'interchangeabilité locale dans un RCD [FREUDER 1991] : *substituabilité au voisinage* et *interchangeabilité au voisinage*.

Définition 94 (Substituabilité au voisinage) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, $x \in \mathcal{X}$ et $v, v' \in D_x$. v est substituable au voisinage par v' si et seulement si, pour toute contrainte $c \in \mathcal{C}$ telle que $x \in \text{scp}(c)$ et pour tout tuple $\tau \in c$ telle que $\tau[x] = v$, il existe un tuple $\tau' \in c$ tel que $\tau'[x] = v'$ et $\tau'[x'] = \tau[x']$ pour tout $x' \in \text{scp}(c)$ avec $x' \neq x$.

Dans la suite, nous nous concentrerons sur la substituabilité, que nous privilégierons à l'étude de l'interchangeabilité car déterminer la substituabilité suffit dans notre cadre. Cette définition se concentre sur les contraintes impliquant une variable et sur les tuples supports de celles-ci. Si nous étendons cette définition, nous pouvons définir une forme plus forte de substituabilité au voisinage.

Définition 95 (ϕ -substituabilité) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, $x \in \mathcal{X}$, $v, v' \in D_x$ et N l'ensemble de variables composé de x et de son voisinage. v est ϕ -substituable par v' si et seulement si, pour chaque instantiation partielle cohérente \mathcal{I}_N telle que $\mathcal{I}_N(x) = v$, il existe une instantiation partielle cohérente \mathcal{I}'_N telle que $\mathcal{I}'_N(x) = v'$ et $\mathcal{I}'_N(x') = \mathcal{I}_N(x')$, pour tout $x' \in N$ et $x' \neq x$.

Cette définition se concentre cette fois sur un ensemble de variables N au voisinage de x , et permet de déterminer la substituabilité d'instanciations partielles sur N . La définition 95 est plus forte que 94, car elle ne se concentre pas uniquement sur les contraintes entre x et son voisinage, mais elle implique également les contraintes entre les variables de N . L'objectif est de considérer la substituabilité au voisinage comme une approximation de la substituabilité complète. Par conséquent, plus les instanciations partielles localement cohérentes se rapprochent d'une instanciation partielle globalement cohérente, plus l'approximation est intéressante. En nous appuyant sur la définition 95, nous allons utiliser la fermeture par cohérence locale pour éliminer les incohérences locales afin de caractériser notre forme de substituabilité locale. Dans un premier temps, pour faciliter la compréhension des propositions qui suivent, nous donnons une définition d'un RCD pour lequel nous faisons omission d'une variable. Ceci nous permettra de ne pas traiter la variable pour laquelle nous recherchons des valeurs substituables localement.

Définition 96 ($\mathcal{P} \setminus x$) Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD et $x \in \mathcal{X}$. La suppression de la variable x dans \mathcal{P} est un RCD $\mathcal{P} \setminus x = (\mathcal{X}', \mathcal{D}', \mathcal{C}')$ défini par :

- $\mathcal{X}' = \mathcal{X} \setminus \{x\}$;
- $\mathcal{D}' = \{D'_{x'} : x' \neq x \wedge D_{x'} = D_x\}$;
- $\mathcal{C}' = \{c \in \mathcal{C} : x \notin \text{scp}(c)\}$.

Définition 97 (ϕ -substituabilité) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, $x \in \mathcal{X}$, $v, v' \in D_x$, et ϕ un algorithme de fermeture par cohérence locale. v est ϕ -substituable par v' si et seulement si $\phi(\mathcal{P}|_{x=v}) \setminus x \sqsubseteq \phi(\mathcal{P}|_{x=v'}) \setminus x$.

Proposition 24 Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, deux couples $x \in \mathcal{X}$, $v, v' \in D_x$, et ϕ un algorithme de fermeture par cohérence locale. Si v est ϕ -substituable par v' et $\phi(\mathcal{P}|_{x=v'}) = \perp$, alors $\phi(\mathcal{P}|_{x=v}) = \perp$.

Preuve La preuve est immédiate. Supposons que v soit ϕ -substituable par v' . Par définition, $\phi(\mathcal{P}|_{x=v}) \setminus x \sqsubseteq \phi(\mathcal{P}|_{x=v'}) \setminus x$. Donc si $\phi(\mathcal{P}|_{x=v'}) \setminus x = \perp$ alors $\phi(\mathcal{P}|_{x=v}) \setminus x = \perp$. Or, $\phi(\mathcal{P}|_{x=v'}) \setminus x$ est issu de $\phi(\mathcal{P}|_{x=v'})$ auquel x a été supprimé, et $\phi(\mathcal{P}|_{x=v}) \setminus x$ est issu de $\phi(\mathcal{P}|_{x=v})$ auquel x a été supprimé. Par conséquent, si $\phi(\mathcal{P}|_{x=v'}) = \perp$ alors $\phi(\mathcal{P}|_{x=v}) = \perp$. \square

Ce résultat forme le centre d'intérêt de notre contribution. En utilisant un opérateur ϕ de fermeture par cohérence locale, nous proposons une approximation de la substituabilité complète : ϕ -substituabilité. Cette forme de substituabilité est plus forte que la définition 94. Plutôt que de vérifier chaque instanciation partielle au voisinage d'une variable de référence et de vérifier leur cohérence, nous proposons de vérifier la microstructure des RCD obtenue après affectation et propagation. Cependant, cette étape ne nécessite pas de vérifier l'intégralité de la microstructure, puisque nous pouvons nous contenter de vérifier l'état du voisinage de la variable de référence. L'idée générale est d'intégrer notre approche aux algorithmes de prétraitement et de recherche reposant sur un cadre affectation/propagation.

Nous pouvons remarquer qu'il existe un facteur variable dans notre démarche, puisque nous pouvons utiliser toute forme de cohérence locale. Plus la forme de cohérence locale utilisée est forte et plus nous nous approcherons de la substituabilité complète, cependant cette précision à un coût induit par la complexité algorithmique de la cohérence locale utilisée. Puisque notre objectif est de greffer notre approche aux algorithmes classiques, nous concentrerons notre étude sur la fermeture par cohérence d'arcs AC, et sur l'intégration aux algorithmes SAC et MAC.

9.2.2 Substituabilité au prétraitement

Intégration à l'algorithme SAC

Afin de proposer un prétraitement dans lequel est intégré la détection de la substituabilité au voisinage, notre choix se porte naturellement sur l'algorithme SAC. Pour rappel, pour $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un

RCD et une variable $x \in \mathcal{X}$, une valeur $v \in D_x$ est *singleton arc-cohérente* (SAC) si et seulement si $\text{AC}(\mathcal{P}|_{x=v}) \not\equiv \perp$. La propagation par cohérence d'arcs après l'affectation de chaque couple (x, v) présente un grand intérêt puisque nous pouvons profiter du filtrage effectué par SAC et y greffer notre approche. Il est alors facile de faire l'analogie avec le cadre affectation/propagation utilisé pour notre intégration.

L'algorithme 14 présente l'intégration de notre approche à l'algorithme SAC, que nous nommons SNS pour « *SAC and Neighbourhood Substituability* ». Pour $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, chaque couple variable/valeur (x, v) du RCD est affecté et propagé par cohérence d'arcs. S'il résulte que \mathcal{P} est incohérent, alors le couple ne vérifie pas la cohérence d'arcs de singleton et est supprimé. Dans le cas contraire, nous vérifions la substituabilité des différents couples. Dans un premier temps, nous sauvegardons l'état du voisinage, noté $\mathcal{E}_{x=v}$, fonction de la variable x pour la valeur v .

Définition 98 (État du voisinage) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, $x \in \mathcal{X}$ et $v, v' \in D_x$. L'état du voisinage de x dans $\text{AC}(\mathcal{P}|_{x=v})$, noté $\mathcal{E}_{x=v}$, est une application qui, à tout $c \in \mathcal{C}$ tel que $x \in c$, associe un ensemble de tuples τ , tel que $\tau[x] = v$.

Définition 99 (Inclusion d'état du voisinage) Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD, $x \in \mathcal{X}$ et $v, v' \in D_x$. $\mathcal{E}_{x=v} \subseteq \mathcal{E}_{x=v'}$ si et seulement si, pour toute contrainte $c \in \mathcal{C}$ avec $x \in \text{scp}(c)$ et pour tout tuple $\tau \in \mathcal{E}_{x=v}(c)$, il existe un tuple $\tau' \in \mathcal{E}_{x=v'}(c)$ tel que $\tau'[\hat{x}] = \tau[\hat{x}]$.

Ainsi, pour tout $x \in \mathcal{X}$ et $v, v' \in D_x$ avec $v \neq v'$, si $\mathcal{E}_{x=v} \subseteq \mathcal{E}_{x=v'}$ alors v est AC-substituable par v' . Par conséquent, nous pouvons affirmer, par la proposition 24, que si $\phi(\mathcal{P}|_{x=v'})$ ne possède pas de solutions, alors $\phi(\mathcal{P}|_{x=v})$ n'en possède pas non plus. Par contraposée, si $\phi(\mathcal{P}|_{x=v})$ possède une solution alors $\phi(\mathcal{P}|_{x=v'})$ en possède également une. Puisque nous nous intéressons au problème CSP, nous pouvons choisir de ne pas parcourir l'espace de recherche associé à $\phi(\mathcal{P}|_{x=v})$. Pour cela, nous réfutons le couple (x, v) . De manière analogue, si $\mathcal{E}_{x=v'} \subseteq \mathcal{E}_{x=v}$, alors (x, v') est réfuté. La vérification de la cohérence d'arcs de singletons et de la détection de valeurs substituables est réitéré jusqu'à ce qu'il n'y ait plus de réfutations.

De part son intégration, SNS montre clairement qu'il est au moins aussi fort que SAC. Cependant, SNS ne préserve pas l'ensemble des solutions du réseau de contraintes initial. Les valeurs réfutées ne sont pas des valeurs globalement incohérentes, elles permettent seulement de faire abstraction d'un sous-espace de recherche qu'il n'est pas nécessaire de parcourir pour prouver la cohérence d'un RCD. En d'autres termes, chaque solution du RCD initial peut être liée à une solution du RCD fermé par SNS. La forme de substituabilité locale présentée permet la détection de formes de symétries faibles dans les RCD. Nous pourrions même faire mention de la substituabilité de sous-espace de recherche.

Complexité algorithmique de SNS

L'efficacité en temps et en espace d'un tel calcul d'inclusion des supports dépend directement de l'arité de la contrainte. Vérifier la substituabilité pour un RCD d'arité r impose de traiter des contraintes de taille rd^r en nombre de valeurs. En supposant, une répartition uniforme des e contraintes sur les n , chacune est impliquée dans $\frac{e}{n}$ contraintes. La sauvegarde d'un état pour un couple variable/valeur nécessite donc de maintenir $\frac{e}{n}rd^r$ valeurs dans le pire des cas en mémoire. La vérification d'inclusion d'un état dans un autre est donc réalisée en $\mathcal{O}(erd^r)$ dans le pire des cas. Cela permet de donner une idée de la complexité en temps et en espace de l'algorithme SNS.

En espace, la complexité algorithmique dans le pire des cas est essentiellement fonction des états à conserver. Pour comparer les états correspondant à chaque valeur d'une variable, SNS nécessite de conserver d états, ce qui correspond à une complexité en espace en $\mathcal{O}(erd^r)$ dans le pire des cas. Pour

Algorithme 14: Singleton Arc-Consistency and Neighbourhood Substitutability

Entrées : $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$: un RCD
Sortie : SNS(\mathcal{P})

```

1  début
2  |   fini ← Non
3  |   tant que (fini ≠ 0) faire
4  |   |   fini ← Oui
5  |   |   Q ← {(x, v) ∈ X × Dx}
6  |   |   tant que (Q ≠ Oui) faire
7  |   |   |   Choisir et Supprimer (xi, vik) ∈ Q
8  |   |   |   si (AC(P|(xi, vij)) = ⊥) alors
9  |   |   |   |   Supprimer (vij, Dxi)
10  |   |   |   |   fini ← Oui
11  |   |   |   sinon
12  |   |   |   |   Sauvegarder (Exi=vij)
13  |   |   |   |   pour chaque vik ∈ Dxi faire
14  |   |   |   |   |   si (Exi=vij ⊆ Exi=vik) alors
15  |   |   |   |   |   |   Supprimer (vij, Dxi)
16  |   |   |   |   |   |   fini ← Non
17  |   |   |   |   |   sinon
18  |   |   |   |   |   |   si (Exi=vik ⊆ Exi=vij) alors
19  |   |   |   |   |   |   |   Supprimer (vik, Dxi)
20  |   |   |   |   |   |   |   fini ← Non
21  |   |   |   si (Dxi = ∅) alors
22  |   |   |   |   retourner ⊥
23  |   retourner P

```

calculer la complexité en temps dans le pire des cas de SNS, il faut tout d'abord considérer que celui-ci est intégré à l'algorithme SAC. Tout d'abord, il est possible de noter qu'un traitement est effectué pour chaque nd couples variable/valeur. À chaque couple est appliqué la cohérence d'arcs généralisé, la sauvegarde de l'état associé au couple et la comparaison avec les états existants pour une même variable. Dans le cas général, chaque traitement est effectué en $\mathcal{O}(erd^r)$. Par conséquent, la complexité algorithmique en temps dans le pire des cas pour une itération de SNS est $\mathcal{O}(ern d^{r+1})$. Or, comme dans le cadre de l'algorithme SAC classique, la suppression d'une valeur dans le réseau de contraintes, quelle que soit la raison, implique une nouvelle itération. Comme le réseau possède nd valeurs, la complexité algorithmique en temps de l'algorithme SNS est $\mathcal{O}(ern^2 d^{r+2})$.

Les complexités algorithmiques en espace et en temps sont élevées du fait de l'arité des contraintes. Cependant, il faut noter qu'en pratique l'efficacité de l'approche dépend, bien évidemment, des traitements et des structures utilisées. Aussi, puisque cette complexité est fonction de l'arité, il est intéressant de voir le cas particulier des contraintes binaires, qui peut être considéré comme le cas le plus simple pour SNS. La complexité algorithmique en espace est en $\mathcal{O}(ed^2)$ et la complexité algorithmique en temps est $\mathcal{O}(en^2 d^4)$, c'est-à-dire comparable à l'algorithme SAC classique (non optimal) dans le pire des cas. Le cas des RCD binaires nous intéresse particulièrement car les états correspondent aux domaines des variables au voisinage d'une variable, il est donc facile d'optimiser les traitements de sauvegarde des états et des vérifications d'inclusions. La détection de valeurs substituables au voisinage dans le cadre d'un prétraitement peut donc s'avérer intéressante notamment par son intégration facile dans l'algorithme SAC, permettant d'approfondir leurs capacités en exploitant des traitements systématiques de cet algorithme.

9.2.3 Recherche dynamique de la substituabilité

Le développement des différentes étapes de l'approche statique introduit les éléments que nous intégrons aux algorithmes de recherche pour la détection de valeurs substituables. La modification constante du RCD au cours de la recherche permet d'exploiter dynamiquement la substituabilité locale. Dans le cas d'un algorithme de type MAC, le maintien de la cohérence d'arcs s'appuie sur un cadre affectation/propagation, ce qui montre tout de suite la similitude avec SNS. À chaque affectation d'une variable x du RCD, l'état du voisinage de x est sauvegardé et permet la vérification de la substituabilité au voisinage de cette variable. Notre approche peut donc se greffer intuitivement, comme dans le cadre statique, et être appliquée à chaque nœud dans l'arbre de recherche. Cependant, à l'instar de SNS, l'ensemble des solutions du RCD est préservé. Dans SNS, lorsqu'un sous-espace de recherche n'est pas évalué pour cause de substituabilité locale, alors nous pouvons déduire qu'un autre sous-espace de recherche sera plus pertinent. Dans le cas de l'intégration dynamique, nous évitons l'évaluation d'un sous-espace de recherche lorsque celui-ci peut être prouvé incohérent par la substituabilité locale, en exploitant la proposition 24.

Branchement *dway* et branchement binaire

Considérons $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD. À chaque affectation/propagation d'un couple variable/valeur (x, v) , l'état du voisinage $\mathcal{E}_{x=v}$ est conservé. S'il s'avère, que $\mathcal{P}|_{x=v}$ est incohérent, le couple doit être réfuté. Par conséquent, nous associons implicitement l'état du voisinage $\mathcal{E}_{x=v}$ à un état incohérent pour le RCD. Lorsque vient le choix d'affectation de la valeur v' à la variable x , il suffit de vérifier si $\mathcal{E}_{x=v'} \subseteq \mathcal{E}_{x=v}$. Si c'est le cas, v' est substituable localement à v , et puisque l'affectation $x = v$ est globalement incohérente dans \mathcal{P} , alors l'affectation $x = v'$ est globalement incohérente dans \mathcal{P} . Par conséquent, ce couple est réfuté avant l'évaluation de l'arbre de recherche associé. Ceci constitue la base de notre étude pour l'intégration dynamique de la substituabilité, et nous pouvons trivialement déduire l'intégration à un algorithme de type MAC avec branchement *dway* (Algorithme 15). Très peu de modifications sont à apporter à l'algorithme de recherche classique puisque nous maintenons uniquement une liste des valeurs supprimées, afin de parcourir l'ensemble des états qui leur correspondent et vérifier l'inclusion.

En considérant \mathcal{P} le RCD initial, la proposition 24 est vérifiée pour tout $\mathcal{P}|_{\mathcal{I}_X}$, quelle que soit l'instanciation partielle \mathcal{I}_X avec $X \subseteq \mathcal{X} \setminus \{x\}$. C'est donc sur ce résultat que repose les algorithmes SNS et MAC + NS avec branchement *dway*. Cependant, la substituabilité prend une dimension plus intéressante lorsque celle-ci est intégrée à l'algorithme MAC avec branchement binaire.

Proposition 25 Soient $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD tel que $x \in \mathcal{X}$ et $v, v' \in D_x$, ainsi que \mathcal{I}_X une instanciation partielle avec $X \subseteq \mathcal{X} \setminus \{x\}$. v est ϕ -substituable par v' si et seulement si $\phi(\mathcal{P}|_{\mathcal{I}_X \cup \{x=v\}}) \setminus x \subseteq \phi(\mathcal{P}|_{x=v'}) \setminus x$.

Preuve D'après la proposition 97, v est ϕ -substituable par v' si et seulement si $\phi(\mathcal{P}|_{x=v}) \setminus x \subseteq \phi(\mathcal{P}|_{x=v'}) \setminus x$. Or, quelle que soit \mathcal{I}_X une instanciation partielle avec $X \subseteq \mathcal{X} \setminus \{x\}$, $\phi(\mathcal{P}|_{\mathcal{I}_X \cup \{x=v\}}) \subseteq \phi(\mathcal{P}|_{x=v})$. Il en résulte que $\phi(\mathcal{P}|_{\mathcal{I}_X \cup \{x=v\}}) \subseteq \phi(\mathcal{P}|_{x=v'})$. \square

La proposition 25 peut ainsi être exploitée pour intégrer la ϕ -substituabilité aux algorithmes de recherche avec branchement binaire. Posons \mathcal{P} un RCD tel que $x \in \mathcal{X}$ et $v, v' \in D_x$, puis posons \mathcal{I}_X une instanciation partielle de \mathcal{P} telle que $x \notin X$. Supposons que $\phi(\mathcal{P}|_{x=v})$ soit incohérent. En développant l'arbre de recherche, nous cherchons à déterminer la cohérence de $\mathcal{P}|_{\mathcal{I}_X \cup \{x=v'\}}$ avec $v' \neq v$. Si v' est ϕ -substituable par v , alors $\phi(\mathcal{P}|_{\mathcal{I}_X \cup \{x=v'\}}) \setminus x \subseteq \phi(\mathcal{P}|_{x=v}) \setminus x$. Par conséquent, puisque $\phi(\mathcal{P}|_{x=v})$ est incohérent, $\phi(\mathcal{P}|_{\mathcal{I}_X \cup \{x=v'\}})$ l'est également. Notons que l'un des avantages de cette approche exploitant le branchement binaire, est que plus l'instanciation \mathcal{I}_X est grande, et plus la probabilité d'une inclusion

Algorithme 15: MAC + NS avec branchement *dway*

Entrées : $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$: un réseau de contraintes
Sortie : \mathcal{P} modélise-t-il un problème satisfiable?

```

1 début
2    $\mathcal{P}' = \text{GAC}(\mathcal{P})$ 
3   si ( $\mathcal{P}' = \perp$ ) alors
4     | retourner Non
5   sinon
6     | retourner Recherche( $\mathcal{P}'$ )

```

7 **fonction** Recherche(\mathcal{P}) : Booléen

```

8 début
9   si ( $\mathcal{P}$  est atomique) alors
10    | retourner GAC( $\mathcal{P}$ ) est-il cohérent ?
11  sinon
12    |  $Q \leftarrow \emptyset$ 
13    |  $x \leftarrow$  Choisir une variable dans futures( $\mathcal{X}$ )
14    | pour chaque ( $v \in D_x$ ) faire
15      |  $\mathcal{P}' = \text{GAC}(\mathcal{P}_{|x=v})$ 
16      | si ( $\mathcal{P}' \neq \perp$ ) alors
17        | Sauvegarder  $\mathcal{E}_{x=v}$ 
18        | si ( $\forall v' \in Q : \mathcal{E}_{x=v} \not\subseteq \mathcal{E}_{x=v'}$ ) et (Recherche( $\mathcal{P}', X \cup \{x\}$ ) répond Oui) alors
19          | | retourner Oui
20        | sinon
21          | |  $Q \leftarrow Q \cup \{v\}$ 
22    | retourner Non

```

$\phi(\mathcal{P}_{|\mathcal{I}_X \cup \{x=v'\}}) \setminus x \sqsubseteq \phi(\mathcal{P}_{|x=v}) \setminus x$ est grande. Par conséquent, il est plus facile de détecter la substituabilité lorsque la différence des niveaux d'affectations de $x = v$ et $x = v'$ est grande. De ce fait, l'impact de la détection de la substituabilité est plus importante pour le branchement binaire que pour le branchement *dway*.

L'algorithme 16 présente l'intégration de la détection de la substituabilité à l'algorithme MAC avec branchement binaire. Il faut donc conserver les états au voisinage et nous définissons une application Q qui à toute variable associe un ensemble de valeurs, correspondant aux états sauvegardés. Dans l'algorithme MAC + NS avec branchement *dway*, les états sauvegardés dépendent de la variable choisie, ce qui permet de n'avoir qu'à maintenir un nombre d'états correspondant au nombre de valeurs de cette variable. Pour le branchement binaire, deux états pour des couples variable/valeur peuvent être comparés à des niveaux différents, nous devons donc avoir une gestion fine de ceux-ci. L'application Q contient simplement la liste des états à comparer pour vérifier la substituabilité. C'est la seule modification majeure par rapport à l'intégration effectuée dans l'algorithme MAC + NS avec branchement *dway*. Après l'affectation et la propagation d'un couple (x, v) , nous sauvegardons l'état du voisinage pour ce couple. Nous vérifions ensuite s'il y a substituabilité avec les valeurs précédemment réfutées, en comparant les états au voisinage. Si ce n'est pas le cas, l'arbre de recherche associé à $x = v$ est évalué. S'il contient une solution, alors le RCD est cohérent. Sinon, l'état au voisinage $\mathcal{E}_{x=v}$ est donc associé à un état incohérent, et nous permettons la comparaison avec des états futurs afin de détecter la substituabilité et d'élaguer l'arbre de recherche. La valeur v est ensuite réfutée et l'algorithme se poursuit de manière classique.

Notons que le cas du branchement *dway* n'est autre qu'un cas particulier du branchement binaire. Comme nous l'avons évoqué précédemment, détecter les valeurs substituables dans le cas de MAC + NS avec branchement binaire a un impact plus important qu'avec branchement *dway*. Cependant, cet avan-

Algorithme 16: MAC + NS avec branchement binaire

Entrées : $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$: un réseau de contraintes
Sortie : \mathcal{P} modélise-t-il un problème satisfiable ?

```

1  début
2  |   Q ← ∅
3  |    $\mathcal{P}' \leftarrow \text{GAC}(\mathcal{P})$ 
4  |   si ( $\mathcal{P}' = \perp$ ) alors
5  |   |   retourner Non
6  |   sinon
7  |   |   retourner Recherche( $\mathcal{P}'$ , Q)

```

8 **fonction** Recherche(\mathcal{P} , Q) : **Booléen**
9 **début**
10 | si (\mathcal{P} est atomique) alors
11 | | retourner Oui
12 | sinon
13 | | $x \leftarrow$ Choisir une variable dans $\text{futures}(\mathcal{X})$
14 | | $v \leftarrow$ Choisir une valeur dans D_x
15 | | $\mathcal{P}' \leftarrow \text{GAC}(\mathcal{P}_{|x=v})$
16 | | si ($\mathcal{P}' \neq \perp$) alors
17 | | | Sauvegarder $\mathcal{E}_{x=v}$
18 | | | si ($\forall v' \in Q(x) : \mathcal{E}_{x=v} \not\subseteq \mathcal{E}_{x=v'}$) alors
19 | | | | si (Recherche(\mathcal{P}' , Q) répond Oui) alors
20 | | | | | retourner Oui
21 | | | | sinon
22 | | | | | $Q(x) \leftarrow Q(x) \cup \{v\}$
23 | | | $\mathcal{P} \leftarrow \text{GAC}(\mathcal{P}_{|x \neq v})$
24 | | | si ($\mathcal{P} = \perp$) alors
25 | | | | retourner Non
26 | | | sinon
27 | | | | Aller à l'instruction : 10

tage a un coût en mémoire.

Complexité

Afin de calculer la complexité algorithmique de notre approche dynamique, nous n'étudions pas la complexité globale des algorithmes MAC + NS avec branchement binaire ou *dway*, mais nous calculons uniquement la complexité algorithmique en temps et en espace de la phase de détection de la substituabilité.

Pour la complexité algorithmique en espace dans le pire des cas, nous devons séparer les cas des branchements binaire et *dway*. Dans le cas du branchement *dway*, nous conservons $d - 1$ états du voisinage dans le pire des cas. Par conséquent, la complexité en espace est en $\mathcal{O}(erd^r)$. Dans le cas du branchement binaire, nous pouvons supposer qu'il y ait eu $n \times (d - 1)$ réfutations effectuées, et donc autant d'états sauvegardés. MAC + NS avec branchement binaire requiert un espace en $\mathcal{O}(enrd^r)$. L'approche exploitant le branchement *dway* possède donc l'avantage de l'espace.

La détection de la substituabilité durant la recherche est appliquée à chaque affectation durant la recherche. Nous calculons donc la complexité algorithmique en temps dans le pire des cas de la détection de la substituabilité pour une affectation. Le pire des cas correspond à la réfutation de $d - 1$ valeurs du domaine de taille d , ce qui induit $d - 1$ vérification d'inclusion des états. Puisque chaque comparaison

nécessite le parcours de $e(r-1)d^{r-1}$ valeurs, la complexité algorithmique en temps dans le pire des cas pour vérifier la substituabilité d'une valeur du domaine d'une valeur est $\mathcal{O}(erd^r)$. Notons que ce résultat est identique pour les approches exploitant le branchement binaire et le branchement *dway*.

Pour donner un ordre d'idée sur le cas binaire, la complexité en espace dans le pire cas est $\mathcal{O}(end^2)$ (resp. $\mathcal{O}(ed^2)$) pour le branchement binaire (resp. *dway*) et la complexité en temps dans le pire des cas est $\mathcal{O}(ed^2)$ pour déterminer si la valeur affectée peut être substituée. Comme pour SNS, le cas des RCD binaires nous intéresse fortement car la complexité algorithmique en temps est proche de celle de la vérification de la cohérence d'arcs. Malgré l'avantage donné à MAC + NS avec branchement *dway* en terme d'espace, nous choisissons d'étudier principalement MAC + NS avec branchement binaire.

9.3 Expérimentations

Avant de développer les différents cadres étudiés précédemment, quelques petites précisions sur le protocole expérimental. Celui-ci est divisé en deux parties : l'étude de l'approche statique SNS et l'étude de l'approche dynamique MAC + NS avec branchement binaire, que nous noterons *ns+dual*. Chaque approche a été comparée à son homologue dit « de base », c'est-à-dire sans détection de substituabilité au voisinage (respectivement SAC et dual). Les expérimentations ont été réalisées dans le cadre binaire essentiellement. Le cas des instances d'arités supérieures montrent des lacunes en ce qui concernent la complexité en espace. L'implémentation a été réalisée en JAVA à l'aide de la plateforme Abscon [LECOUTRE & TABARY 2006] développé au CRIL, et les expérimentations ont été menées sur un processeur Xéon 3GHz avec 1Go de RAM. Les problèmes utilisées sont ceux de la troisième compétition internationale de solveurs CSP réalisée en 2008 (<http://cpai.ucc.ie/08/>). Les problèmes sont divisés en quatre catégories : les contraintes en extension ou en intention, ainsi que les problèmes satisfiables ou insatisfiables. Le temps de résolution a été limité à 20 minutes et la mémoire à 900Mo.

9.3.1 Intégration statique

Pour donner un aperçu des performances de l'intégration statique, nous appliquons SNS en pré-traitement du problème, puis nous tentons la résolution à l'aide d'un algorithme MAC classique. Nous comparons cette approche avec le pré-traitement SAC et le même algorithme MAC pour la résolution. Le tableau 9.1 présente les résultats de la résolution de tous les problèmes binaires. Dans celui-ci, chaque tableau représente une catégorie de problème. Nous comparons, pour chaque approche, le nombre de problèmes résolus, le temps de résolution total en seconde, le nombre de problèmes touchés par notre pré-traitement, c'est-à-dire pour lesquelles au moins une valeur a été supprimée, et le nombre de valeurs supprimées sur la totalité des problèmes. Remarquons, dans le cas de SNS, que les valeurs supprimées correspondent aux valeurs non-SAC et aux valeurs substituables. Enfin, la ligne « -SNS » signifie que les résultats de l'approche SNS ont été réduits aux problèmes résolus conjointement par les deux approches.

Les résultats globaux sont à l'avantage de SNS. Le critère du nombre de problèmes résolu dans le temps imparti donne un petit avantage à SNS, puisqu'elle permet de décider de la satisfiabilité pour 4 problèmes de plus que l'approche utilisant SAC. Un résultat également intéressant, toutes catégories confondues, en ce qui concerne le nombre de valeurs supprimées par SAC (59771 valeurs sur 229 problèmes touchés) bien plus faible que celui de SNS (182041 valeurs sur 308 problèmes touchés), ce qui montrent la puissance de filtrage de notre approche. Rappelons que SNS ne se contente pas de détecter les valeurs substituables, mais également les valeurs ne vérifiant pas la cohérence d'arcs de singletons. Par conséquent, une seule valeur substituable détectée et supprimée peut conduire à détecter d'autres valeurs non SAC, et *vice versa*. Cependant, cette efficacité est obtenue au dépend du temps de résolution total, puisque sur ce critère notre approche ne prend pas l'ascendant. Malgré un filtrage fort, le temps

de résolution des problèmes avec prétraitement SAC est de 57622 secondes, tandis que notre approche passe 60698 secondes pour la résolution des mêmes problèmes.

Extension Satisfiable				
Méthode	Nb problèmes	Temps	Nb touchés	Nb supprimées
SAC	296	20624	57	4373
SNS	296	21019	72	41909

Extension Insatisfiable				
Méthode	Nb problèmes	Temps	Nb touchés	Nb supprimées
SAC	195	13824	68	6869
SNS	195	13807	78	8372

Intention Satisfiable				
Méthode	Nb problèmes	Temps	Nb touchés	Nb supprimées
SAC	253	14867	57	16404
SNS	254	18054	110	82432
-SNS	253	17855	109	79618

Intention Insatisfiable				
Méthode	Nb problèmes	Temps	Nb touchés	Nb supprimées
SAC	202	8307	47	24382
SNS	205	10857	51	41546
-SNS	202	8017	49	37690

TABLE 9.1 – Résultats globaux : SAC contre SNS

Le nuage de points Figure 9.1 permet de donner une vue globale des temps de résolution des instances. Sur celui-ci chacun des points représente un problème, dont les coordonnées sont désignées par les temps de résolution : en abscisse, le temps de résolution par l’approche utilisant le prétraitement SAC, et en ordonnée le temps de résolution par l’approche utilisant le prétraitement SNS. Le temps limite étant de 1200 secondes, les abscisses et ordonnées sont définies dans $[0 ; 1200]$. La droite d’équation $x = 1200$ montre ainsi les quatre problèmes résolus par SNS et non par SAC. Hormis ceux-ci, nous pouvons constater les mauvais résultats pour les temps de résolution de SNS, notamment en observant une nuée de point proche des coordonnées $(0, 200)$. Ceux-ci constituent la majorité du temps perdu par notre approche, que nous détaillerons plus loin. Malgré ces quelques instances, la majorité des points se situent autour de la droite $y = x$, ce qui montre des temps très proches pour les deux approches.

Bien que notre approche s’avère moins intéressante si nous recherchons la performance en temps, certaines classes de problèmes peuvent être soulignées. En particulier, le groupement de point énoncé précédemment correspond à une classe de problèmes : les JobShop. Tout d’abord, les problèmes job-shop utilisés [SADEH & FOX 1996] sont des problèmes satisfiables de 50 variables, de domaines de taille comprise entre 100 à 250, et comprenant 265 contraintes binaires. La plupart de ces instances ont été résolues facilement par l’approche SAC, sans pour autant bénéficier d’un filtrage. Sur 46 instances résolues par les deux approches, 29 sont trivialement résolues, c’est-à-dire qu’il a été possible de répondre à la satisfiabilité sans faire de retour-arrière. Le tableau 9.2 présente une comparaison des deux approches étudiées, dont les critères sont le nombre de problèmes résolus, en particulier le nombre total et le nombre de problèmes prouvés satisfiables sans retour-arrière, le temps des algorithmes, en particulier le temps total en secondes et le temps de prétraitement en secondes, ainsi que des statistiques sur le prétraitement et l’arbre de recherche, tel que le nombre de valeurs supprimées par le prétraitement et le nombre de nœuds de l’arbre de recherche. Ces résultats confirment l’efficacité du traitement effectué par SNS pour le critère de la suppression de valeurs. Quelques instances sont trivialement satisfaites lorsque notre prétraitement filtre les valeurs substituables. De plus, le nombre de nœuds de l’arbre de recherche

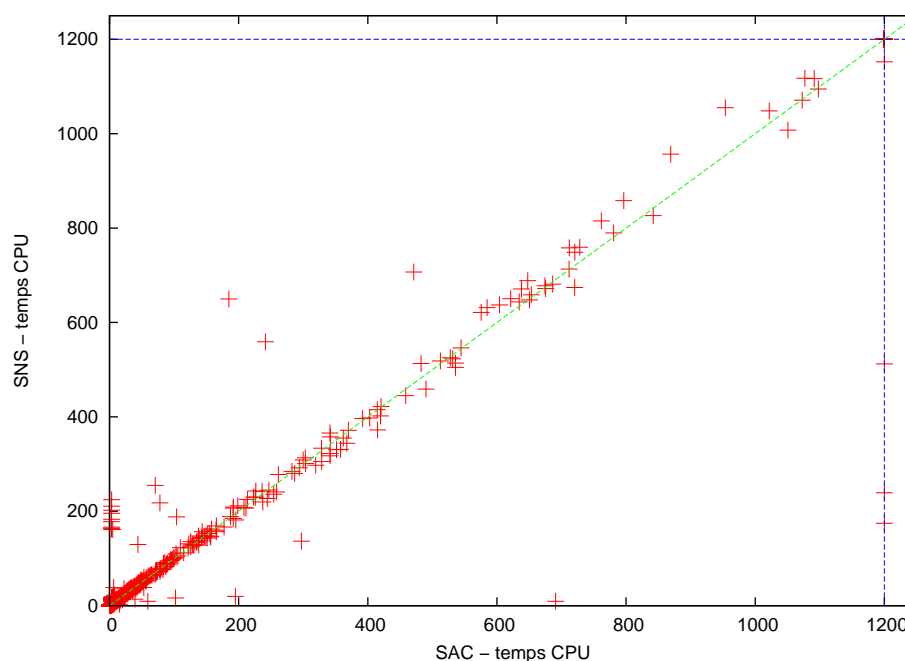


FIGURE 9.1 – Comparaison des temps CPU : SAC contre SNS

est fortement réduit. Le gros inconvénient vient du temps de prétraitement (93% du temps total) et c'est sur ce critère que notre approche démontre des lacunes. Notre approche passe beaucoup de temps afin de réduire considérablement l'espace de recherche, mais lorsque les problèmes sont très faciles, cela se traduit par un gaspillage de temps. Lorsque nous regardons les résultats globaux des problèmes satisfiables avec contraintes en intention, nous apercevons une perte en temps de l'ordre de 3000 secondes, alors que ces problèmes JobShop montre une perte de temps de près de 2000 secondes. Cependant, nous pouvons également noter des gain significatif dans cette classe de problème puisque que SNS permet de résoudre en 10s (dont 8s de prétraitement) une 46e instance que SAC résout en 691s (dont 2s de prétraitement). SNS supprime 517 valeurs et réduit le nombre de nœuds de l'arbre de recherche à 192, tandis que l'approche classique ne permet aucun filtrage, et impose un arbre de recherche de 4 millions de nœuds. Intégré ce résultat à notre synthèse n'aurait donc pas été pertinent.

Prétraitement	Problèmes résolus		Temps			Statistique	
	Total	Trivial	Total	Prétraitement	Ratio	Suppression	Nœuds
SAC	45	29	175	85	48,6%	0	208181
SNS	45	35	2369	2208	93,2%	19293	8801

TABLE 9.2 – Synthèses des résultats des problèmes jobShop

Pour terminer notre analyse de l'approche statique, nous pouvons étudier une deuxième classe d'instances intéressante : les instances Taillard OpenShop [TAILLARD 1993]. Ces problèmes d'optimisations sont des instances dont le nombre de variables est n^2 , de taille maximale avoisinant les 300 valeurs et le nombre de contraintes est de $4n^2$ pour $n \in \{4, 5, 7, 10, 20\}$ dans notre cas. 43 instances ont été résolues par les deux approches, et celles-ci n'effectuent aucun filtrage pour 18 d'entre elles. Notons que le temps de prétraitement pour ces 18 instances est d'environ 305s pour SNS et 303s pour SAC. La vérification des états n'a donc pas été trop coûteuse en temps. Le tableau 9.3 se compose des mêmes critères que

pour les jobShop. Nous pouvons donc vérifier, sur cette classe de problèmes, que le temps de prétraitement occupe la même part du temps total que pour la classe précédente. En particulier, le prétraitement SNS occupe 91% du temps total pour diviser l'espace de recherche à évaluer par 6 environ.

Prétraitement	Problèmes résolus		Temps			Statistique	
	Total	Trivial	Total	Prétraitement	Ratio	Suppression	Nœuds
SAC	25	3	1161	590	50,8%	10850	870351
SNS	25	4	1950	1775	91%	67105	151663

TABLE 9.3 – Synthèses des résultats des problèmes Taillard-OS

Chacune des catégories fait ressortir une classe de problèmes particulière, plus ou moins facile à résoudre. Pour autant, il est difficile de déterminer une classe d'instances complète pour laquelle l'approche SNS est meilleure en temps. Le prétraitement SNS s'avère très efficace pour réduire l'espace de recherche, mais ceci a un coût. Le prétraitement est peu efficace en temps lorsque le filtrage est fort, cependant il facilite largement la recherche en réduisant l'espace de recherche à évaluer, ce qui lui permet de résoudre plus d'instances qu'une approche dont le prétraitement est SAC.

9.3.2 Intégration dynamique

Afin de vérifier l'efficacité de notre approche dynamique, nous utilisons l'algorithme MAC + NS avec branchement binaire, que nous notons **ns+dual**. Celui-ci est comparé à l'algorithme MAC avec branchement binaire. Le tableau 9.4 présente les résultats sur les problèmes binaires décrits précédemment. Pour chaque algorithme, nous comparons le nombre de problèmes prouvés satisfiables, le temps de résolution en secondes, le nombre de problèmes touchés par notre approche, c'est-à-dire ayant au moins une valeur substituable détectée, ainsi le nombre de valeurs substituables détectées. Nous pouvons donc remarquer que le nombre de problèmes résolus est exactement le même, cependant les deux approches donnent des résultats différents pour huit problèmes. Ainsi, l'approche **ns+dual** décide de la cohérence de quatre problèmes que MAC ne sait pas décider dans le temps imparti, et *vice versa*. Sur les 943 problèmes résolus communément par les deux approches, MAC montre un temps cumulé de 50065 secondes, tandis que notre approche cumule 48228 secondes. Bien que les gains ne soient pas extraordinaires, détecter la substituabilité en cours de recherche peut donc s'avérer intéressant malgré un traitement systématique. Cette présentation globale des résultats nous permet de pointer quelques aspects particuliers du compromis entre temps de calcul et élagage pour faciliter la recherche. Sur les problèmes satisfiables décrits en extension, la détection de la substituabilité s'avère toujours utile malgré le faible nombre de valeurs supprimées (132 valeurs sur la résolution de 28 problèmes touchés par notre approche). Au contraire, pour les problèmes insatisfiables décrits en extension, la forte suppression de valeurs sur les 36 problèmes touchés est pénalisante, et le compromis entre détection et élagage est beaucoup moins intéressant. La précision de l'approche nous semble être un mauvais résultat, puisque seulement 10% à 20% des problèmes traités sont touchés par la détection de la substituabilité. Enfin, pour les instances communes résolues, l'approche **ns+dual** est toujours meilleure, ce que l'on remarque aussi sur le nuage de points.

Le nuage de points 9.2 illustre bien les résultats et montrent le léger gain en temps pour la détection de la substituabilité. Une majorité de points se situent sous la droite $y = x$, c'est-à-dire en faveur de la détection de la substituabilité. Cependant, quatre problèmes montrent un avantage pour l'approche classique. Nous pouvons également remarquer les problèmes par l'une ou (exclusif) l'autre des approches. Nous en tirons un point positif car sur les quatre problèmes résolus uniquement par notre approche, tous le sont avant 700 secondes.

Extension Satisfiable				
Méthode	Nb problèmes	Temps	Nb touchés	Nb supprimées
MAC	296	20845	-	-
ns+dual	294	17972	28	132
-MAC	294	18601	-	-
Extension Insatisfiable				
Méthode	Nb problèmes	Temps	Nb touchés	Nb supprimées
MAC	196	13717	-	-
ns+dual	196	13771	36	66302
Intention Satisfiable				
Méthode	Nb problèmes	Temps	Nb touchés	Nb supprimées
MAC	253	9869	-	-
ns+dual	254	9391	58	22439
-MAC	251	9479	-	-
-ns+dual	251	8696	55	18444
Intention Insatisfiable				
Méthode	Nb problèmes	Temps	Nb touchés	Nb supprimées
-MAC	202	8268	-	-
ns+dual	203	8435	29	152014
-ns+dual	202	7789	28	104917

TABLE 9.4 – Résultats globaux : dual contre ns+dual

Malgré le peu de problèmes touchés par notre traitement dynamique, il est possible de dégager quelques classes de problèmes, notamment les deux classes étudiées pour l’approche statique. Tout d’abord, pour les 46 problèmes des jobShop résolus par les deux approches, 29 sont trivialement satisfiables. Sur les 17 problèmes restants, 10 sont touchés par la détection de la substituabilité. En particulier, l’approche MAC met près de 660 secondes pour répondre à la satisfiabilité pour l’un des problèmes, alors que ns+dual répond en 1 seconde, grâce à la détection de 3 valeurs substituables. Les autres problèmes sont particulièrement faciles à résoudre et les deux approches sont très proches l’une de l’autre (15s pour ns+dual et 17s pour MAC). La deuxième classe de problèmes correspond aux problèmes Taillard OpenShop, dont la synthèse est présentée Tableau 9.5. Celui-ci présente le nombre de problèmes résolus sur 45, le temps de résolution en seconde, le nombre de valeurs détecté substituable et le nombre de nœuds de l’arbre évalué au cours de la recherche. Sur 45 problèmes, 2 problèmes sont résolues uniquement par ns+dual et 23 sont résolus communément par MAC et ns+dual. Nous pouvons remarquer le gain de temps sur ces 23 problèmes, qui n’est cependant pas proportionnel à l’élégage de l’arbre de recherche, puisque ns+dual permet d’évaluer deux fois moins de nœuds. Le constat est néanmoins que la détection de la substituabilité pour les problèmes Taillard OpenShop est également intéressante pour la détection dynamique.

Méthode	Nb problèmes	Temps	Nb touchés	Nb supprimées
MAC	23	1724	-	1046687
ns+dual	25	1999	3008	532313
-ns+dual	23	1443	2611	460024

TABLE 9.5 – Synthèses des résultats des problèmes Taillard-OS

Enfin, quelques problèmes montrent des résultats peu intéressants, desquels nous pouvons retirer quelques informations. La grande taille de l’instance de certains problèmes non résolus, comme un problème d’allocation de fréquence avec 2500 variables de taille maximum 360, montre les limites de la détection dynamique de la substituabilité. D’autres problèmes simples sont pénalisés par une forte dé-

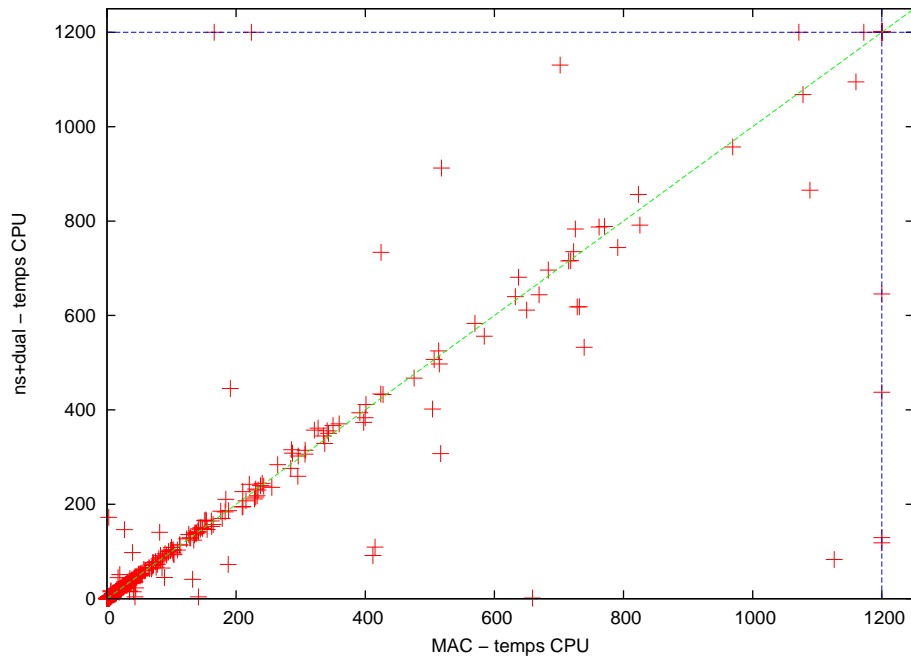


FIGURE 9.2 – Comparaison des temps CPU : dual contre ns+dual

tection de valeurs substituables.

9.4 Conclusion

Nous avons présenté une nouvelle forme de substituabilité dans le cadre des RCD, en nous appuyant sur le cadre affectation et propagation de la cohérence locale des algorithmes de recherche. Cette ϕ -substituabilité, plus forte que la substituabilité au voisinage [FREUDER 1991], peut être intégrée dans deux cadres différents. Nous proposons un prétraitement, dans lequel la détection de valeurs substituables se traduit par leurs suppressions puisque nous savons que si elles participent à une solution, alors nous pourrions également trouver une solution sans elles. Le prétraitement ne préserve donc pas l'ensemble des solutions, il consiste à détecter une forme de symétrie faible afin de réduire l'espace de recherche à évaluer. Lorsque la détection de la substituabilité est intégrée à un algorithme de recherche, elle permet de supprimer des valeurs qui ne conduiront trivialement pas à une solution, sans avoir à parcourir le sous-espace de recherche associé. Cette approche montre des aspects intéressants dans le cadre d'un algorithme de recherche avec branchement binaire, puisque la probabilité de détecter des valeurs ϕ -substituables augmente. Les résultats expérimentaux des deux approches sont globalement meilleurs qu'une approche classique. Le prétraitement avec détection de la ϕ -substituabilité permet de décider de la cohérence de quelques problèmes supplémentaires par rapport à une approche classique, tandis que l'intégration dynamique montre des résultats équivalents à une approche classique sur ce critère. Nous pouvons extraire certaines classes de problèmes pour lesquelles la détection de la ϕ -substituabilité s'avère particulièrement efficace, comme les problèmes *jobshop* et *openshop*. Cependant, notre approche montre quelques lacunes lorsque les problèmes sont facilement résolus par une approche classique. La détection de la ϕ -substituabilité peut être pénalisante, alors que l'instance peut être prouvée trivialement. Pour de très grands problèmes, l'intégration dynamique est également pénalisée par le maintien de structure pour détecter la ϕ -substituabilité.

Nous entrevoyons de nombreuses perspectives au sujet de leur intégration dans différents cadres. L'idée de cette approche nous vient d'un constat effectué en transformant un RCD en formule CNF. Ce qui nous a amené à définir la ϕ -substituabilité. Cependant, dans le cadre propositionnel, appliquer à la lettre notre définition pour définir une UP-substituabilité est différente de notre intuition initiale. Dans le cadre des RCQ, cette approche peut être également très utile, en particulier parce que nous connaissons *a priori* la table de compositions qui est la seule contrainte imposée dans le RCQ. Outre l'adaptation à d'autres cadres, il nous semble intéressant d'étudier l'impact de la suppression de valeurs substituables sur les techniques d'inférences modernes, tels que l'analyse de conflits, la pondération des contraintes ou l'apprentissage. Parmi les questions que nous souhaitons explorer, nous pouvons nous demander quel est l'impact de l'élagage d'un sous-espace de recherche sur la pondération de sous-problèmes incohérents, ou comment exploiter la substituabilité pour déduire de nouvelles informations pendant l'apprentissage. Enfin, nous souhaitons également étudier l'intérêt que peut susciter la détection de la ϕ -substituabilité pour des problèmes tels que MAXCSP ou CSP pondéré, pour notre prétraitement et par une intégration dynamique.

Conclusion générale

Les modèles de représentations étudiés dans cette thèse sont liés par la résolution de problèmes sous forme de contraintes. Chacun constitue un domaine de recherche spécifique et chaque domaine a sa propre vision de l'approche à adopter face à un problème. Un cadre de modélisation est choisi en fonction de la nature du problème, de la simplicité de la modélisation, de la compacité de sa représentation, de son expressivité ou pour l'efficacité de ses outils algorithmiques. Cependant, chaque domaine peut contribuer à développer de nouvelles approches originales et efficaces dans d'autres formalismes. L'objectif de nos travaux a été de mettre en évidence et de comprendre les liens entre les domaines des *réseaux de contraintes discrets*, les *réseaux de contraintes qualitatives* et les *formules propositionnelles*, afin de proposer de nouvelles approches pour décider de la cohérence de problèmes de nature temporelle ou spatiale pour le raisonnement qualitatif. L'étude de ces différents domaines nous a conduit à définir différentes transformations et exploiter les propriétés théoriques et les outils algorithmiques développés dans chacun des cadres.

Dans un premier temps, en étudiant des transformations bien connues des RCD vers les formules CNF, nous avons mis en évidence l'intérêt d'aller au-delà des transformations classiques. En nous appuyant sur la logique propositionnelle et ses propriétés, nous avons caractérisé de nouvelles formules CNF à partir des formules obtenues à partir des transformations classiques. Décider de la satisfiabilité de ces formules permet de décider de la cohérence du RCD initial. Cependant, le sens de la formule obtenue sort du cadre classique. L'analogie entre modèle et RCD montre que certains domaines sont vides, mais qu'il est possible de décider de la cohérence du RCD. De plus, en exploitant les règles de résolution sur une transformation basée sur la cohérence d'arcs, nous avons montré qu'il est possible de faire le lien entre une formule CNF et des formes de cohérence locale plus fortes que la cohérence d'arcs. En particulier, nous pouvons définir une formule CNF modélisant la cohérence de chemins.

L'étude de la transformation des RCD vers les formules CNF nous a ensuite conduit à définir une transformation des RCQ vers les formules CNF en nous appuyant sur une approche similaire. Cette transformation est réalisable pour tout formalisme qualitatif binaire. Dans un premier temps, chaque élément de la microstructure d'un RCQ est retranscrit par une variable propositionnelle. Pour la modélisation des contraintes, nous nous appuyons sur les propriétés d'inverse et de composition faible des formalismes qualitatifs. Cependant, nous avons montré, dans le cadre de RCQ normalisé, que la modélisation pouvait se concentrer sur des arcs orientés du RCQ. Ainsi, une seule contrainte est modélisée : la table de composition du formalisme qualitatif. Nous proposons deux approches, par les conflits et par les supports, afin de satisfaire la cohérence locale du RCQ. La transformation proposée, polynomiale en temps dans la taille du RCQ, est correcte mais n'est pas complète dans le cas général. Quel que soit le formalisme qualitatif, elle permet de mettre en évidence des scénarios \diamond -cohérents. Cependant, cette condition n'est pas suffisante pour décider de la cohérence d'un scénario pour certains formalismes qualitatifs. Pour de nombreux formalismes qualitatifs, tels que l'algèbre des intervalles ou le formalisme RCC, la transformation est correcte et complète pour décider de la cohérence du RCQ.

En sortant du cadre des transformations classiques, nous avons également défini une transformation

s'appuyant sur les classes traitables de certains formalismes qualitatifs : les relations convexes. Un ordre sur les relations atomiques de certains formalismes qualitatifs permet de mettre en évidence une structure de treillis. Chaque intervalle de ce treillis caractérise une relation convexe. Nous modélisons les propriétés de ce treillis - infimum, supremum, inverse, distributivité, composition faible - pour décrire les relations convexes. Cette transformation est polynomiale en temps dans la taille du RCQ. Dans le cas de RCQ convexes, nous remarquons que la transformation produit une formule CNF de Horn. De plus, puisque toute relation du formalisme qualitatif considéré peut être découpée en sous-relations convexes, nous définissons une transformation complète et correcte pour décider de la cohérence d'un RCQ.

Nous avons également proposé une transformation des RCQ vers les RCD, qu'il est possible de réaliser quel que soit le formalisme qualitatif binaire. Afin de décrire la microstructure du RCQ à l'aide d'un RCD, nous faisons la correspondance entre contraintes qualitatives du RCQ et variables du RCD, ainsi qu'entre relations atomiques et valeurs. En modélisant la table de compositions comme une contrainte sur tout triplet de variables du RCD, nous imposons la \diamond -cohérence entre les valeurs du domaine des variables du RCD. Cette transformation est polynomiale en temps dans la taille du RCQ, et elle est correcte et complète pour décider de la cohérence d'un RCQ dans le cas des formalismes qualitatifs pour lesquels la \diamond -cohérence permet de vérifier si un scénario est cohérent. Nous montrons qu'un RCQ \diamond -fermé est un RCD fortement 3-cohérent par notre transformation. Nous nous intéressons plus particulièrement au résultat inverse, en montrant qu'appliquer la cohérence d'arcs généralisée sur un RCD résultant de notre transformation, est équivalent à appliquer la \diamond -fermeture sur le RCQ.

En exploitant la transformation vers les RCD, nous proposons une approche originale pour la relaxation de contraintes qualitatives afin de faciliter la preuve l'incohérence de certains RCQ difficiles. En particulier, nous souhaitons relaxer la relation totale, qui autorise tous les choix pour un couple de variables. Ce choix est motivé par la taille de la relation totale, la plus grande relation complexe des formalismes qualitatifs, et par les propriétés de cette relation. Puisque les RCQ doivent satisfaire la \diamond -fermeture, et que la relation totale est l'élément absorbant dans la composition faible, relaxer cette relation nous semble intéressant pour nous approcher de la complétude. Comme nous ne pouvons pas rendre cette contrainte plus permissive, nous appliquons la relaxation à la table de composition. Ainsi, les contraintes totales ne sont plus soumises à la table de composition du formalisme qualitatif, mais à une contrainte universelle. À l'aide d'un procédé inverse, visant à restaurer les contraintes relaxées, nous proposons un algorithme permettant de compléter l'approche, en nous appuyant sur les classes de relations traitables contenant la relation totale et les relations singletons. Lorsqu'une solution d'un RCQ relaxé a été trouvée, celle-ci est restaurée avec les contraintes relaxées précédemment. Nous obtenons un RCQ composée uniquement de relations atomiques et de relations totales, et donc nous avons un RCQ traitable.

Enfin, nous définissons une nouvelle forme de substituabilité dans le cadre des RCD. Plus forte que la substituabilité au voisinage, la ϕ -substituabilité s'appuie sur le schéma affectation/propagation des algorithmes modernes. En propageant deux valeurs du domaine d'une variable par ϕ -fermeture à travers un RCD, nous obtenons deux nouveaux RCD. Si l'un est inclus dans l'autre, à l'exception de la variable contenant les valeurs, alors une valeur est substituable à l'autre. La ϕ -substituabilité est donc une forme de symétrie faible. Dans un premier temps, nous proposons d'exploiter cette approche en prétraitement. Nous définissons un algorithme, intégré à l'algorithme de cohérence d'arcs de singletons, afin d'éliminer les valeurs substituables. Dans ce cadre, une valeur éliminée n'est pas globalement incohérente, cependant si elle conduit à une solution, alors il existe une autre valeur menant également à une solution. Cette approche ne préserve pas l'ensemble des solutions, mais nous réduisons l'espace de recherche à parcourir tout en conservant la complétude. Nous proposons ensuite une intégration dynamique de cette approche, en vérifiant la ϕ -substituabilité à chaque décision positive durant la recherche. Cela permet de montrer un autre aspect de la ϕ -substituabilité, puisqu'elle permet de supprimer des valeurs dont le sous-espace de recherche associé a été prouvé incohérent. Cette approche est donc complète. Globalement, manipuler la ϕ -substituabilité s'avère intéressant en pratique. Le prétraitement montre des résultats intéressants mais

l'approche peine un peu pour des problèmes triviaux, en passant beaucoup de temps à détecter la substituable inutilement. Malgré le maintien systématique de la détection au cours de la recherche, cette approche se montre également intéressante. Elle amène à faciliter la tâche de la recherche par rapport à une approche sans détection de la ϕ -substituabilité. Nous pouvons extraire certaines classes de problèmes intéressantes de ces résultats. En particulier, les problèmes de *jobshop* et d'*openshop* sur lesquels notre approche s'avère utile.

Les perspectives à nos travaux sont nombreuses et de natures diverses. Pour les transformations en général, nous souhaitons naturellement étudier et adapter les techniques algorithmiques des différents cadres. En particulier, l'analyse de conflits et l'apprentissage sont très peu développés pour le cadre qualitatif. Chaque RCQ est construit sur un formalisme qualitatif bien défini pour lequel la table de composition et les différentes propriétés liées à sa nature sont connues. Cela nous semble être un atout majeur du cadre qualitatif, et il nous paraît possible de développer des techniques d'inférences spécifiques à la nature de certains formalismes qualitatifs. D'autant plus que la notion de *classe traitable* est très présente dans les algorithmes de recherche et qu'elle constitue un atout majeur pour la résolution de problèmes qualitatifs. Lier analyse de conflits et apprentissage avec l'étude des classes traitables semble d'un grand intérêt. Dans notre étude, nous avons également fait le lien entre GAC et la \diamond -fermeture, et ces formes de cohérence locale sont omniprésentes dans les diverses approches développées pour améliorer les techniques de résolution. Adapter des algorithmes optimaux de GAC, en maintenant des supports résiduels ou en maintenant une liste de révision, peut amener à faciliter tous les outils liés à la \diamond -cohérence dans les RCQ.

Pour les transformations définies vers la logique propositionnelle, le premier aspect à explorer semble lié avant tout à la modélisation. Étudier le comportement des algorithmes de recherche et d'inférence en fonction de la modélisation semble être un aspect intéressant. Non pas pour multiplier les approches, mais pour en extraire l'essence et comprendre ce qui constitue des informations pertinentes et essentielles, dans la modélisation des contraintes. Pour construire la formule propositionnelle à partir d'un RCQ, nous avons défini des variables propositionnelles à partir de propositions booléennes. En s'appuyant sur la transformation avec support, la règle de résolution nous permet d'imaginer de nouvelles formes de cohérence locales, plus forte que la cohérence d'arcs. Or, dans le cadre des RCQ, il est difficile de calculer ou de manipuler des formes de cohérences locales plus fortes que la \diamond -cohérence, qui imposerait le maintien d'une table de composition de dimensions supérieures à 2. Exploiter le cadre propositionnel peut amener à automatiser et à faciliter cette tâche, ainsi qu'à définir de nouvelles transformations s'appuyant sur des formes de cohérences locales fortes. En considérant la transformation basée sur les conflits, nous obtenons une méthode complète pour décider de la cohérence du RCQ même sans imposer la contrainte spécifiant qu'il est nécessaire que le domaine de chaque variable doit contenir au plus une valeur. Cette information constitue, pour nous, une voie à explorer pour caractériser dans quelle condition et pour quel problème cette approche est plus intéressante. De plus, lorsque nous utilisons la règle de résolution sur certaines clauses, nous obtenons une formule propositionnelle totalement opposée, permettant de montrer qu'il est possible de répondre au problème de la cohérence sans imposer aux domaines des variables de contenir au moins une valeur. Cette approche donne une vision différente du problème, que nous pouvons étudier et adapter dans différents cadres. Le second aspect à développer dans le cadre des transformations des RCQ vers la logique propositionnelle est lié à la taille des formules propositionnelles. En effet, puisque chaque tuple de la table de composition est modélisé pour chaque triplet de variables, la formule obtenue est de taille considérable, ce qui peut engendrer des problèmes en pratique. Pourtant, bien qu'elles portent sur des variables différentes, ces clauses ont une structure et un sens commun. Nous pouvons imaginer exploiter la compilation des connaissances afin de réduire le nombre de ces clauses à un sous-ensemble essentiel. D'autres formalismes pour la modélisation des RCQ peuvent également être envisagés tel que *SAT Modulo Theory*, problème de décision s'appuyant sur des

théories exprimées en logique du premier ordre. Notons, à ce titre, qu'exploiter la logique du premier ordre pour modéliser la table de composition rappelle la transformation des RCD, pour laquelle la table de compositions n'est maintenue qu'une seule fois pour tout triplet de variables. Ainsi, exploiter la logique du premier ordre uniquement pour la propagation unitaires peut être une perspectives prometteuse à explorer afin de réduire la taille des formules tout en conservant la puissance des outils algorithmiques du problème SAT.

Plus spécifiquement, pour la transformation vers les formules CNF exploitant la classe des relations convexes, nous souhaitons développer cette approche en nous appuyant sur d'autres classes traitables, tels que la classe des relations préconvexes, qui est un sur-ensemble de la classe des convexes, afin d'obtenir un découpage plus faible des relations complexes du RCQ. Nous entrevoyons également la possibilité d'adapter cette approche aux transformations des RCQ vers les RCD, pour bénéficier de la compacité des RCD tout en exploitant la structure de treillis de certains formalismes qualitatifs.

La relaxation des contraintes qualitatives en exploitant la relaxation de la table de composition nous semble être une approche intéressante pour la détection ou la réparation d'incohérence. Elle laisse entrevoir la possibilité de manipuler localement la table de composition. Nous entrevoyons quelques applications, notamment pour des problèmes globalement incohérents pour lesquels le formalisme qualitatif n'est pas adapté. Par exemple, supposons la modélisation de préférences transitives à l'aide de l'algèbre des intervalles, l'incohérence peut être due au choix du formalisme qualitatif et non au problème lui-même. Des préférences non-transitives, modélisés à l'aide de l'algèbre des intervalles cycliques, auraient conduit à un autre résultat.

La substituabilité nous amène à autant de perspectives. La première, et la plus naturelle, est d'étudier différentes formes de cohérence locale ϕ pour la détection de la ϕ -substituabilité. Ensuite, nous envisageons leur intégration aux différents cadres. La substituabilité au voisinage et la ϕ -substituabilité sont des approches à étudier, tant pour le problème SAT que pour le problème CSP qualitatif. Dans le cadre des RCQ, cette approche peut bénéficier de la connaissance de la table de composition, qui reste invariante tout au long de la recherche. Une autre perspective est l'étude de l'impact de la suppression de valeurs substituables sur les techniques d'inférences modernes, tels que l'analyse de conflits, la pondération des contraintes ou l'apprentissage. Parmi les questions que nous souhaitons explorer, nous pouvons nous demander quel est l'impact de l'élagage d'un sous-espace de recherche sur la pondération de sous-problèmes incohérents, ou comment exploiter la substituabilité pour déduire de nouvelles informations pendant l'apprentissage. Nous pouvons étendre cette perspective à tous les formalismes et à toute technique d'inférence s'appuyant sur l'élagage d'un sous-espace de recherche. Enfin, la ϕ -substituabilité est une forme de symétrie qui apporte des résultats pertinents à la vue des expérimentations. Nous souhaitons donc adapter la ϕ -substituabilité à des problèmes tels que MAXCSP ou CSP pondéré.

La dernière perspective que nous entrevoyons est liée à l'apport que peut constituer le cadre qualitatif dans les autres domaines. Les formalismes qualitatifs et le cadre théorique sur lesquels ils sont définis a permis la mise en évidence de classes traitables, très exploitées dans les outils algorithmiques pour le problème CSP qualitatif. Définir les transformations des formules CNF et des RCD vers les RCQ est un aspect qui peut être interprété de différentes manières et qui nous intéresse particulièrement. Cela permettrait d'offrir une vision différentes du problème SAT et du problème CSP, et compléterait l'étude de systèmes de contraintes pour le raisonnement qualitatif.

Annexe A

Algèbre des Intervalles

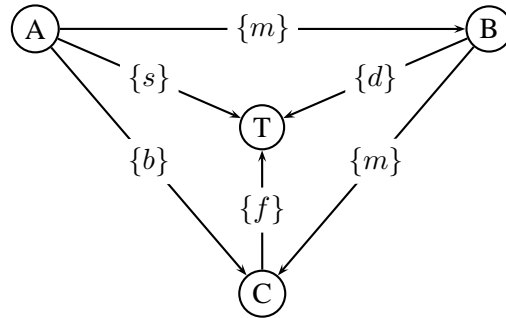
A.1 Table de composition

◇	eq	b	a	d	di	o	oi	m	mi	s	si	f	fi
eq	eq	b	a	d	di	o	oi	m	mi	s	si	f	fi
b	b	b	ψ	b o m d s	b	b	b o m d s	b	b o m d s	b	b	b o m d s	b
a	a	ψ	a	a o i m i d f	a	a o i m i d f	a	a o i m i d f	a	a o i m i d f	a	a	a
d	d	b	a	d	ψ	b o m d s	a o i m i d f	b	a	d	a o i m i d f	d	b o m d s
di	di	b o m d i f i	a o i d i m i s i	o o i d s f d i s i f i e q	di	o d i f i	o i d i s i	o d i f i	o i d i s i	o d i f i	di	o i d i s i	di
o	o	b	a o i d i m i s i	o d s	b o m d i f i	b o m	eq d d i o o i s s i f f i	b	o i d i s i	o	o d i f i	d s o	b o m
oi	oi	b o m d i f i	a	o i d f	a o i m i d i s i	eq d d i o o i s s i f f i	a o i m i	o d i f i	a	o i d f	a o i m i	oi	o i d i s i
m	m	b	a o i d i m i s i	o d s	b	b	o d s	b	eq f f i	m	m	o d s	b
mi	mi	b o m d i f i	a	o i d f	a	o i d f	a	eq s s i	a	o i d f	a	mi	mi
s	s	b	a	d	b o m d i f i	b o m	o i d f	b	mi	s	eq s s i	d	b o m
si	si	b o m d i f i	a	o i d f	di	o d i f i	oi	o d i f i	mi	eq s s i	si	oi	di
f	f	b	a	d	a o i m i d i s i	o d s	a o i m i	m	a	d	a o i m i	f	eq f f i
fi	fi	b	a o i d i m i s i	o d s	di	o	o i d i s i	m	o i d i s i	o	di	eq f f i	fi

FIGURE A.1 – Table de composition pour l’algèbre des intervalles

A.2 Algèbre des intervalles affaiblie

Reprenons l'exemple des trois étudiants corrigeant les exercices durant la séance de cours. Cependant, nous imposons « la correction A commence le cours T », « la correction B est pendant le cours T », « la correction C termine le cours », « la correction A suit directement la correction B », « la correction A est avant la correction C » et enfin « la correction C suit directement la correction B ». Ces informations sont donc décrites le RCQ construits sur le formalisme d'Allen suivant :



Supposons maintenant que l'enseignant impose à une correction à Denise, notée D , toujours pendant le temps de cours, et sans interférer avec la correction de ses camarades. En ajoutant l'information $D\{d\}T$, la \diamond -fermeture permet de déduire les informations $D\{a\}A$, $D\{b, a\}B$ et $D\{b\}C$. Par conséquent, la correction D doit se dérouler *après* la correction A , et *avant* la correction C , et puisque les relations atomiques sont complètes et mutuellement exclusives, la correction D doit être réalisée *avant* ou *après* la correction B . Or, quel que soit le choix effectué entre B et D , il n'est pas possible de construire un scénario cohérent intégrant la correction D .

Pour illustrer le raisonnement effectué par Renz et Ligozat, imaginons un formalisme qualitatif proche du formalisme d'Allen, que nous nommerons *formalisme qualitatif affaibli* qui diffère du formalisme d'Allen par l'ensemble des relations atomiques \mathcal{B}' . Dans celui-ci, la relation complexe $\{b, a\}$ est factorisée en une seule relation atomique $ppi = b \cup a$ permettant de préciser que deux intervalles soient distincts, tel que $ppi = ppi^{-1}$. L'ensemble $\mathcal{B}' = \{ppi, m, o, s, d, f, eq, fi, di, si, oi, mi\}$ forme bien une partition de $\mathbb{Q}^2 \times \mathbb{Q}^2$, et toutes les propriétés des formalismes qualitatifs sont vérifiées. Les compositions faibles et fortes sont équivalentes. Il est possible de déduire la table de composition faible du formalisme affaibli à partir de celle du formalisme d'Allen. Pour cela, il suffit de considérer que toute relation atomique a et b du formalisme d'Allen et affaibli en ppi . Les compositions suivent directement la définition de la composition faible, puisque celle-ci est la plus petite relation contenu dans $2^{\mathcal{B}}$, toute composition contenant a ou b dans le formalisme d'Allen contiendront ppi dans le formalisme affaibli.

Reprenons l'exemple de la répartition des corrections dans ce nouveau formalisme, et retranscrivons les informations obtenues.

Formalisme d'Allen	Formalisme affaibli
$A\{m\}B$	$A\{m\}B$
$B\{m\}C$	$B\{m\}C$
$A\{b\}C$	$A\{ppi\}C$
$A\{s\}T$	$A\{s\}T$
$B\{d\}T$	$B\{d\}T$
$C\{f\}T$	$C\{f\}T$
$D\{a\}A$	$D\{ppi\}A$
$D\{b, a\}B$	$D\{ppi\}B$
$D\{b\}C$	$D\{ppi\}C$

Ceci, dans notre formalisme affaibli, constitue bien un scénario, qui plus est fermé algébriquement mais incohérent. La \diamond -cohérence ne suffit donc plus pour répondre à la cohérence d'un scénario. L'affaiblissement du formalisme d'Allen par une approximation de la relation $\{b, a\}$ rend donc le formalisme affaibli non fermé.

Annexe B

Des RCD aux formules CNF

B.1 Transformation en exploitant la cohérence d'arcs

Pour illustrer la transformation exploitant les supports, nous définissons $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un RCD tel que : $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$, avec $D_{x_1} = D_{x_2} = D_{x_3} = D_{x_4} = \{0, 1, 2\}$. Deux contraintes sont définies dans \mathcal{C} , décrites par les relations :

$$c_{(x_1, x_2, x_3)} = \{(0, 1, 2), (1, 0, 2), (2, 1, 0)\};$$

$$c_{(x_2, x_3, x_4)} = \{(0, 1, 2), (1, 0, 2), (2, 1, 0)\}$$

Notons que ce RCD ne possède qu'une solution : $\mathcal{I} = \{(x_1, 2), (x_2, 1), (x_3, 0), (x_4, 2)\}$.

Dans la suite, nous présentons chaque formule sous la forme d'un tableau de clause. Le tableau B.1 illustre la modélisation de la formule « Au-Moins-Une », permettant de garantir que chaque variable doit être affectée à au moins une valeur. Le tableau B.2 présente la modélisation de la formule « Au-Plus-Une », afin de garantir que chaque variable soit affectée à au plus une valeur. Ces deux formules permettent de garantir la mutuelle exclusivité des valeurs du domaine de chaque variable. Enfin, le tableau B.2 présente la formule supports, exploitant la notion de cohérence d'arcs. Nous présentons les clauses associées à chaque contrainte, telle que chaque valeur de chaque variable est liée aux autres par ces tuples supports. Remarquons que cette formule possède certaines clauses unitaires. Ceci représente, par la définition que nous donnons de la transformation, l'ensemble des valeurs qui ne sont pas arcs-cohérentes.

$$\begin{array}{l} \wedge \\ \wedge \\ \wedge \\ \wedge \end{array} \begin{array}{l} [x_1 = 0] \vee [x_1 = 1] \vee [x_1 = 2] \\ [x_2 = 0] \vee [x_2 = 1] \vee [x_2 = 2] \\ [x_3 = 0] \vee [x_3 = 1] \vee [x_3 = 2] \\ [x_4 = 0] \vee [x_4 = 1] \vee [x_4 = 2] \end{array}$$

TABLE B.1 – $\Sigma_{\mathcal{P}}^{ALO}$

$$\begin{array}{l} \neg[x_1 = 0] \vee \neg[x_1 = 1] \\ \wedge \neg[x_1 = 0] \vee \neg[x_1 = 2] \\ \wedge \neg[x_1 = 1] \vee \neg[x_1 = 2] \\ \wedge \neg[x_2 = 0] \vee \neg[x_2 = 1] \\ \wedge \neg[x_2 = 0] \vee \neg[x_2 = 2] \\ \wedge \neg[x_2 = 1] \vee \neg[x_2 = 2] \\ \wedge \neg[x_3 = 0] \vee \neg[x_3 = 1] \\ \wedge \neg[x_3 = 0] \vee \neg[x_3 = 2] \\ \wedge \neg[x_3 = 1] \vee \neg[x_3 = 2] \\ \wedge \neg[x_4 = 0] \vee \neg[x_4 = 1] \\ \wedge \neg[x_4 = 0] \vee \neg[x_4 = 2] \\ \neg[x_4 = 1] \vee \neg[x_4 = 2] \end{array}$$

TABLE B.2 – $\Sigma_{\mathcal{P}}^{AMO}$

$c_{(x_1, x_2, x_3)} = \{(0, 1, 2), (1, 0, 2), (2, 1, 0)\}$	$c_{(x_2, x_3, x_4)} = \{(0, 1, 2), (1, 0, 2), (2, 1, 0)\}$
$\neg[x_1 = 0] \vee [x_2 = 1]$	$\neg[x_2 = 0] \vee [x_3 = 1]$
$\neg[x_1 = 0] \vee [x_3 = 2]$	$\neg[x_2 = 0] \vee [x_4 = 2]$
$\neg[x_1 = 1] \vee [x_2 = 0]$	$\neg[x_2 = 1] \vee [x_3 = 0]$
$\neg[x_1 = 1] \vee [x_3 = 2]$	$\neg[x_2 = 1] \vee [x_4 = 2]$
$\neg[x_1 = 2] \vee [x_2 = 1]$	$\neg[x_2 = 2] \vee [x_3 = 1]$
$\neg[x_1 = 2] \vee [x_3 = 0]$	$\neg[x_2 = 2] \vee [x_4 = 0]$
$\neg[x_2 = 0] \vee [x_1 = 1]$	$\neg[x_3 = 0] \vee [x_2 = 1]$
$\neg[x_2 = 0] \vee [x_3 = 2]$	$\neg[x_3 = 0] \vee [x_4 = 2]$
$\neg[x_2 = 1] \vee [x_1 = 0] \vee [x_1 = 2]$	$\neg[x_3 = 1] \vee [x_2 = 0] \vee [x_2 = 2]$
$\neg[x_2 = 1] \vee [x_3 = 0] \vee [x_3 = 2]$	$\neg[x_3 = 1] \vee [x_4 = 0] \vee [x_4 = 2]$
$\neg[x_2 = 2]$	$\neg[x_3 = 2]$
$\neg[x_3 = 0] \vee [x_1 = 2]$	$\neg[x_4 = 0] \vee [x_2 = 2]$
$\neg[x_3 = 0] \vee [x_2 = 1]$	$\neg[x_4 = 0] \vee [x_3 = 1]$
$\neg[x_3 = 1]$	$\neg[x_4 = 1]$
$\neg[x_3 = 2] \vee [x_1 = 0] \vee [x_1 = 1]$	$\neg[x_4 = 2] \vee [x_2 = 0] \vee [x_2 = 1]$
$\neg[x_3 = 2] \vee [x_2 = 1] \vee [x_2 = 0]$	$\neg[x_4 = 2] \vee [x_3 = 1] \vee [x_3 = 0]$

 TABLE B.3 – $\Sigma_{\mathcal{P}}^{\text{AC}}$

B.2 Vers de nouvelles modélisations de cohérence locale

Cette section a pour but de présenter la modélisation d'autres formes de cohérence locale que la cohérence d'arcs en formule CNF. Le tableau B.4 décrit la cohérence de chemins par la formule CNF $\Sigma_{\mathcal{P}}^{\text{PC}}$ pour chaque contrainte du RCD. Par exemple, $(x_2, 0)$ et $(x_3, 2)$ (clauses bleues) présente un support commun dans $c_{(x_1, x_2, x_3)}$ tel que $\{(x_2, 0), (x_1, 1)\}$ et $\{(x_3, 2), (x_1, 1)\}$ sont cohérentes. Ainsi, si $(x_2, 0) \in \mathcal{I}$ et $(x_3, 2) \in \mathcal{I}$ alors $(x_1, 1) \in \mathcal{I}$ pour satisfaire cette contrainte. Or, dans $c_{(x_2, x_3, x_4)}$, $(x_2, 0)$ et $(x_3, 2)$ n'ont aucun support commun. Par conséquent, si $(x_2, 0) \in \mathcal{I}$ et $(x_3, 2) \in \mathcal{I}$ alors le résultat est incohérent. $(x_2, 0)$ et $(x_3, 2)$ ne sont pas chemins-cohérents et ne peuvent pas apparaître ensemble dans une instanciation sans la rendre incohérente. Au contraire, $(x_2, 0)$ et $(x_3, 2)$ (clauses vertes) sont chemins-cohérentes. Par conséquent, elles admettent un support communs dans x_1 et dans x_4 .

$c_{(x_1, x_2, x_3)} = \{(0, 1, 2), (1, 0, 2), (2, 1, 0)\}$	$c_{(x_2, x_3, x_4)} = \{(0, 1, 2), (1, 0, 2), (2, 1, 0)\}$
$\neg[x_1 = 0] \vee \neg[x_2 = 0]$	$\neg[x_2 = 0] \vee \neg[x_3 = 0]$
$\neg[x_1 = 0] \vee \neg[x_2 = 1] \vee [x_3 = 2]$	$\neg[x_2 = 0] \vee \neg[x_3 = 1] \vee [x_4 = 2]$
$\neg[x_1 = 0] \vee \neg[x_2 = 2]$	$\neg[x_2 = 0] \vee \neg[x_3 = 2]$
$\neg[x_1 = 1] \vee \neg[x_2 = 0] \vee [x_3 = 2]$	$\neg[x_2 = 1] \vee \neg[x_3 = 0] \vee [x_4 = 2]$
$\neg[x_1 = 1] \vee \neg[x_2 = 1]$	$\neg[x_2 = 1] \vee \neg[x_3 = 1]$
$\neg[x_1 = 1] \vee \neg[x_2 = 2]$	$\neg[x_2 = 1] \vee \neg[x_3 = 2]$
$\neg[x_1 = 2] \vee \neg[x_2 = 0]$	$\neg[x_2 = 2] \vee \neg[x_3 = 0]$
$\neg[x_1 = 2] \vee \neg[x_2 = 1] \vee [x_3 = 0]$	$\neg[x_2 = 2] \vee \neg[x_3 = 1] \vee [x_4 = 0]$
$\neg[x_1 = 2] \vee \neg[x_2 = 2]$	$\neg[x_2 = 2] \vee \neg[x_3 = 2]$
$\neg[x_1 = 0] \vee \neg[x_3 = 0]$	$\neg[x_2 = 0] \vee \neg[x_4 = 0]$
$\neg[x_1 = 0] \vee \neg[x_3 = 1]$	$\neg[x_2 = 0] \vee \neg[x_4 = 1]$
$\neg[x_1 = 0] \vee \neg[x_3 = 2] \vee [x_2 = 1]$	$\neg[x_2 = 0] \vee \neg[x_4 = 2] \vee [x_3 = 1]$
$\neg[x_1 = 1] \vee \neg[x_3 = 0]$	$\neg[x_2 = 1] \vee \neg[x_4 = 0]$
$\neg[x_1 = 1] \vee \neg[x_3 = 1]$	$\neg[x_2 = 1] \vee \neg[x_4 = 1]$
$\neg[x_1 = 1] \vee \neg[x_3 = 2] \vee [x_2 = 0]$	$\neg[x_2 = 1] \vee \neg[x_4 = 2] \vee [x_3 = 0]$
$\neg[x_1 = 2] \vee \neg[x_3 = 0] \vee [x_2 = 1]$	$\neg[x_2 = 2] \vee \neg[x_4 = 0] \vee [x_3 = 1]$
$\neg[x_1 = 2] \vee \neg[x_3 = 1]$	$\neg[x_2 = 2] \vee \neg[x_4 = 1]$
$\neg[x_1 = 2] \vee \neg[x_3 = 2]$	$\neg[x_2 = 2] \vee \neg[x_4 = 2]$
$\neg[x_2 = 0] \vee \neg[x_3 = 0]$	$\neg[x_3 = 0] \vee \neg[x_4 = 0]$
$\neg[x_2 = 0] \vee \neg[x_3 = 1]$	$\neg[x_3 = 0] \vee \neg[x_4 = 1]$
$\neg[x_2 = 0] \vee \neg[x_3 = 2] \vee [x_1 = 1]$	$\neg[x_3 = 0] \vee \neg[x_4 = 2] \vee [x_2 = 1]$
$\neg[x_2 = 1] \vee \neg[x_3 = 0] \vee [x_1 = 2]$	$\neg[x_3 = 1] \vee \neg[x_4 = 0] \vee [x_2 = 2]$
$\neg[x_2 = 1] \vee \neg[x_3 = 1]$	$\neg[x_3 = 1] \vee \neg[x_4 = 1]$
$\neg[x_2 = 1] \vee \neg[x_3 = 2] \vee [x_1 = 0]$	$\neg[x_3 = 1] \vee \neg[x_4 = 2] \vee [x_2 = 0]$
$\neg[x_2 = 2] \vee \neg[x_3 = 0]$	$\neg[x_3 = 2] \vee \neg[x_4 = 0]$
$\neg[x_2 = 2] \vee \neg[x_3 = 1]$	$\neg[x_3 = 2] \vee \neg[x_4 = 1]$
$\neg[x_2 = 2] \vee \neg[x_3 = 2]$	$\neg[x_3 = 2] \vee \neg[x_4 = 2]$

TABLE B.4 – $\Sigma_{\mathcal{P}}^{\text{PC}}$

Bibliographie

- [ALLEN 1981] James F. ALLEN. « An interval-based representation of temporal knowledge ». Dans *Proceedings of the Seventh International Joint Conference on Artificial Intelligence (IJCAI'81)*, pages 221–226, 1981.
- [ALLEN 1983] James F. ALLEN. « Maintaining knowledge about temporal intervals ». *Community of the ACM*, 26(11) :832–843, 1983.
- [BACCHUS & VAN BEEK 1998] Fahiem BACCHUS et Peter van BEEK. « On the conversion between non-binary constraint satisfaction problems ». Dans *Proceedings of the Fifteenth national conference on Artificial intelligence (AAAI'98)*, pages 311–318, 1998.
- [BALBIANI & CONDOTTA 2002] Philippe BALBIANI et Jean-François CONDOTTA. « Spatial Reasoning About Points in a Multidimensional Setting ». *Applied Intelligence*, 17(3) :221–238, 2002.
- [BALBIANI & OSMANI 2000] Philippe BALBIANI et Aomar OSMANI. « A model for reasoning about topologic relations between cyclic intervals ». Dans *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2000)*, pages 378–385, 2000.
- [BALBIANI *et al.* 1998] Philippe BALBIANI, Jean-François CONDOTTA, et Luis FARIÑAS DEL CERRO. « A model for reasoning about bidimensional temporal relations ». Dans *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 124–130, 1998.
- [BALBIANI *et al.* 1999] Philippe BALBIANI, Jean-François CONDOTTA, et Luis FARIÑAS DEL CERRO. « A tractable subclass of the block algebra : constraint propagation and preconvex relations ». Dans *The Ninth Portuguese Conference on Artificial Intelligence (EPIA'99)*, pages 75–89, 1999.
- [BALBIANI *et al.* 2003a] Philippe BALBIANI, Jean-François CONDOTTA, et Luis FARIÑAS DEL CERRO. « Tractability results in the block algebra ». *Journal of Logic and Computation*, 12(5) :885–909, 2003.
- [BALBIANI *et al.* 2003b] Philippe BALBIANI, Jean-François CONDOTTA, et Gérard LIGIZAT. « Reasoning about cyclic space : axiomatic and computational aspects ». *Spatial cognition III : Routes and navigation, human memory and learning, spatial representation and spatial learning*, pages 348–371, 2003.

- [BELDICEANU *et al.* 2009] Nicolas BELDICEANU, Mats CARLSSON, Jean-Xavier RAMPON, et Sophie DEMASSEY. « Global Constraint Catalog ». <http://www.emn.fr/z-info/sdemasse/gccat/>, 2009.
- [BESSIERE & DEBRUYNE 2008] Christian BESSIERE et Romuald DEBRUYNE. « Theoretical analysis of singleton arc consistency and its extensions ». *Artificial Intelligence*, 172(1) :29–41, 2008.
- [BESSIÈRE & RÉGIN 1996] Christian BESSIÈRE et Jean-Charles RÉGIN. « MAC and Combined Heuristics : Two Reasons to Forsake FC (and CBJ ?) on Hard Problems ». Dans *Proceedings of the 2nd International Conference on Principles and Practice of Constraint Programming (CP'96)*, pages 61–75, 1996.
- [BESSIÈRE & RÉGIN 2001] Christian BESSIÈRE et Jean-Charles RÉGIN. « Refining the basic constraint propagation algorithm ». Dans *Proceedings of the 17th international joint conference on Artificial intelligence (IJCAI'01)*, pages 309–315, 2001.
- [BESSIÈRE *et al.* 1996] Christian BESSIÈRE, Amar ISLI, et Gérard LIGOZAT. « Global Consistency in Interval Algebra Networks : Tractable Subclasses ». Dans *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI'96)*, pages 3–7, 1996.
- [BESSIÈRE *et al.* 1999] Christian BESSIÈRE, Eugene C. FREUDER, et Jean-Charles RÉGIN. « Using constraint metaknowledge to reduce arc consistency computation ». *Artificial Intelligence*, 107(1) :125–148, 1999.
- [BESSIÈRE *et al.* 2005] Christian BESSIÈRE, Jean-Charles RÉGIN, Roland H. C. YAP, et Yuanlin ZHANG. « An optimal coarse-grained arc consistency algorithm ». *Artificial Intelligence*, 165(2) :165–185, 2005.
- [BESSIÈRE 1994] Christian BESSIÈRE. « Arc-consistency and arc-consistency again ». *Artificial Intelligence*, 65(1) :179–190, 1994.
- [BESSIÈRE 2006] Christian BESSIÈRE. « Constraint Propagation ». Rapport technique, LIRMM, 2006.
- [BISTARELLI *et al.* 2000] Stefano BISTARELLI, Philippe CODOGNET, et Francesca ROSSI. « An abstraction framework for soft constraints and its relationship with constraint propagation ». Dans *Proceedings of SARA'2000*, pages 71–86, 2000.
- [BISTARELLI *et al.* 2002] Stefano BISTARELLI, Philippe CODOGNET, et Francesca ROSSI. « Abstracting soft constraints : framework, properties, examples ». *Artificial Intelligence*, 139 :175–211, 2002.
- [BITNER & REINGOLD 1975] James R. BITNER et Edward M. REINGOLD. « Backtrack programming techniques ». *Communication of the ACM*, 18(11) :651–656, 1975.
- [BOUSSEMART *et al.* 2004a] Frédéric BOUSSEMART, Fred HEMERY, Christophe LECOUTRE, et Lakhdar SAÏS. « Boosting Systematic Search by Weighting Constraints ». Dans *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI'2004)*, pages 146–150, 2004.

-
- [BOUSSEMART *et al.* 2004b] Frédéric BOUSSEMART, Fred HEMERY, et Christophe LECOUTRE. « Revision ordering heuristics for the Constraint Satisfaction Problem ». Dans *Proceedings of the 1st International Workshop on Constraint Propagation and Implementation (CPAI'04)*, pages 29–43, 2004.
- [BROXVALL & JONSSON 2003] Mathias BROXVALL et Peter JONSSON. « Point algebras for temporal reasoning : algorithms and complexity ». *Artificial Intelligence*, 149(2) :179–220, 2003.
- [CASEAU 1991] Yves CASEAU. « Abstract interpretation of constraints on ordered domains ». Dans *Proceedings of ISLP'91*, pages 435–452, 1991.
- [CHMEISS *et al.* 2008] Assef CHMEISS, Vincent KRAWCZYK, et Lakhdar SAÏS. « Redundancy in CSPs ». Dans *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI'08)*, pages 907–908, 2008.
- [CHOUEIRY & NOUBIR 1998] Berthe Y. CHOUEIRY et Guevara NOUBIR. « On the computation of local interchangeability in discrete constraint satisfaction problems ». Dans *Proceedings of AAAI'98*, pages 326–333, 1998.
- [CHOUEIRY *et al.* 1995] Berthe Y. CHOUEIRY, Boi FALTINGS, et Rainer WEIGEL. « Abstraction by interchangeability in resource allocation ». Dans *Proceedings of IJCAI'95*, pages 1694–1710, 1995.
- [CONDOTTA & D'ALMEIDA 2007] Jean-François CONDOTTA et Dominique D'ALMEIDA. « Qualitative Constraints Representation for the Time and Space in SAT ». Dans *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'07)*, pages 74–77, 2007.
- [CONDOTTA *et al.* 2006a] Jean-François CONDOTTA, Dominique D'ALMEIDA, Christophe LECOUTRE, et Lakhdar SAÏS. « From Qualitative to Discrete Constraint Networks ». Dans *Proceedings of the International Workshop on Qualitative Constraint Calculi held with KI'06 (Wks-QCC-KI'06)*, pages 54–64, 2006.
- [CONDOTTA *et al.* 2006b] Jean-François CONDOTTA, G. LIGOZAT, et Mahmoud SAADE. « A Qualitative Algebra Toolkit ». Dans *2nd IEEE International Conference on Information Technologies : from Theory to Applications (ICTTA'06)*, pages 1251–1252, 2006.
- [CONDOTTA *et al.* 2007a] Jean-François CONDOTTA, Dominique D'ALMEIDA, Christophe LECOUTRE, et Lakhdar SAÏS. « Relaxation of qualitative constraint networks ». Dans *Proceedings of the 7th International conference on Abstraction, reformulation, and approximation (SARA'07)*, pages 93–108, 2007.
- [CONDOTTA *et al.* 2007b] Jean-François CONDOTTA, Gérard LIGOZAT, et Mahmoud SAADE. « Eligible and Frozen Constraints for Solving Temporal Qualitative Constraint Networks ». Dans *Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming (CP'07)*, pages 806–814, 2007.
- [CONDOTTA *et al.* 2008] Jean-François CONDOTTA, Souhila KACI, Pierre MARQUIS, et Nicolas SCHWIND. « Utiliser la logique propositionnelle pour la fu-

- sion de réseaux de contraintes qualitatives ». Dans *Représentation et Raisonnement sur le Temps et l'Espace (RTE'08)*, 2008.
- [COOPER 1989] Martin C. COOPER. « An optimal k-consistency algorithm ». *Artificial Intelligence*, 41(1) :89–95, 1989.
- [COUSOT & COUSOT 1992] Patrick COUSOT et Radhia COUSOT. « Abstract interpretation frameworks ». *Logic and Computation*, 2(4) :447–511, 1992.
- [D'ALMEIDA & SAÏS 2009] Dominique D'ALMEIDA et Lakhdar SAÏS. « Enhancing Neighbourhood Substitutability Thanks to Singleton Arc Consistency ». Dans *Proceedings of the 2009 21st IEEE International Conference on Tools with Artificial Intelligence (ICTAI '09)*, pages 91–98, 2009.
- [DAVIS *et al.* 1962] Martin DAVIS, George LOGEMANN, et Donald LOVELAND. « A machine program for theorem-proving ». *Communications of the ACM*, 5(7) :394–397, 1962.
- [DEBRUYNE & BESSIÈRE 1997] Romuald DEBRUYNE et Christian BESSIÈRE. « Some Practicable Filtering Techniques for the Constraint Satisfaction Problem ». Dans *Proceedings of the 15th international joint conference on Artificial intelligence (IJCAI'97)*, pages 412–417, 1997.
- [DECHTER & MEIRI 1989] Rina DECHTER et Itay MEIRI. « Experimental evaluation of pre-processing techniques in constraint satisfaction problems ». Dans *Proceedings of the 11th international joint conference on Artificial intelligence (IJCAI'89)*, pages 271–277, 1989.
- [DECHTER 1990] Rina DECHTER. « On the Expressiveness of Networks with Hidden Variables ». Dans *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI'90)*, pages 556–562, 1990.
- [DRAKENGREN & JONSSON 1997] Thomas DRAKENGREN et Peter JONSSON. « Twenty-one large tractable subclasses of Allen's algebra ». *Artificial Intelligence*, 93(1-2) :297–319, 1997.
- [FREUDER & SABIN 1995] Eugene C. FREUDER et Daniel SABIN. « Interchangeability supports abstraction and reformulation for constraint satisfaction ». Dans *Proceedings of SARA'95*, 1995.
- [FREUDER 1978] Eugene C. FREUDER. « Synthesizing constraint expressions ». *Communication of the ACM*, 21(11) :958–966, 1978.
- [FREUDER 1991] Eugene C. FREUDER. « Eliminating Interchangeable Values in Constraint Satisfaction Problems ». Dans *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI'91)*, pages 227–233, 1991.
- [FROST & DECHTER 1995] Daniel FROST et Rina DECHTER. « Look-ahead value ordering for constraint satisfaction problems ». Dans *Proceedings of the 14th international joint conference on Artificial intelligence (IJCAI'95)*, pages 572–578, 1995.
- [GANTNER *et al.* 2008] Zeno GANTNER, Matthias WESTPHAL, et Stefan WÖLFL. « GQR - A fast reasoner for binary qualitative constraint calculi ». Dans *Workshop on Spatial and Temporal Reasoning (AAAI'08)*, 2008.

-
- [GASCHNIG 1978] John GASCHNIG. « Experimental case studies of backtrack vs. Waltz-type vs. new algorithms for satisficing assignment problems ». Dans *Proceedings of the 2nd Canadian Conference on Artificial Intelligence (CSCSI'78)*, pages 268–277, 1978.
- [GENT 2002] Ian GENT. « Arc consistency in SAT ». Dans *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI'2002)*, pages 121–125, 2002.
- [GIUNCHIGLIA & WALSH 1992] Fausto GIUNCHIGLIA et Toby WALSH. « A theory of abstraction ». *Artificial Intelligence*, 57(2-3) :323–389, 1992.
- [GOMES *et al.* 2000] Carla P. GOMES, Bart SELMAN, Nuno CRATO, et Henry KAUTZ. « Heavy-Tailed Phenomena in Satisfiability and Constraint Satisfaction Problems ». *Journal of Automated Reasoning*, 24(1-2) :67–100, 2000.
- [GRÉGOIRE *et al.* 2006] Éric GRÉGOIRE, Bertrand MAZURE, et Cédric PIETTE. « Extracting MUSes ». Dans *Proceeding of the 2006 conference on ECAI 2006*, pages 387–391, 2006.
- [HARALICK & ELLIOTT 1979] Robert M. HARALICK et Gordon L. ELLIOTT. « Increasing tree search efficiency for constraint satisfaction problems ». Dans *Proceedings of the 6th international joint conference on Artificial intelligence (IJCAI'79)*, pages 356–364, 1979.
- [HIRAYAMA & YOKOO 1993] Katsutoshi HIRAYAMA et Makoto YOKOO. « Constraint relaxation in distributed constraint satisfaction problems ». Dans *Proceedings of ICTAI'03*, pages 56–63, 1993.
- [HOARE 1961] Charles A. R. HOARE. « Algorithm 64 : Quicksort ». *Communications of the ACM*, 4(7) :321, 1961.
- [HWANG & MITCHELL 2005] Joey HWANG et David G. MITCHELL. « 2-way vs d-way branching for CSP ». Dans *Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming (CP'05)*, pages 343–357, 2005.
- [ISLI & COHN 2000] Amar ISLI et Anthony G. COHN. « A New Approach to Cyclic Ordering of 2D Orientations Using Ternary Relation Algebras ». *Artificial Intelligence*, 122(1–2) :137–187, 2000.
- [JEROSLOW & WANG 1990] Robert G. JEROSLOW et Jinchang WANG. « Solving Propositional Satisfiability Problems ». *Annals of Mathematics and Artificial Intelligence*, 1 :167–187, 1990.
- [JUSSIEN & BOIZUMAULT 1996] Narendra JUSSIEN et Pascale BOIZUMAULT. « Implementing Constraint Relaxation over Finite Domains using ATMS ». Dans *Over-Constrained Systems*, Lecture Notes in Computer Science, pages 265–280, 1996.
- [JUSSIEN 1997] Narendra JUSSIEN. « *Relaxation de Contraintes pour les problèmes dynamiques* ». PhD thesis, Université de Rennes I, 1997.
- [KACI & PIETTE 2010] Souhila KACI et Cédric PIETTE. « Ordering Intervals : From Qualitative Temporal Constraint Problems to Preference Representation ». Dans *Proceedings of the 10th Workshop on Preferences and Soft*

- Constraints (Co-located with CP 2010)(SOFT10)*, page to appear, 2010.
- [KASIF 1990] Simon KASIF. « On the parallel complexity of discrete relaxation in constraint satisfaction networks ». *Artificial Intelligence*, 45(3) :275–286, 1990.
- [LADKIN 1986] Peter B. LADKIN. « Time Representation : A Taxonomy of Internal Relations ». Dans *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI'86)*, pages 360–366, 1986.
- [LAURIÈRE 1978] Jean-Louis LAURIÈRE. « A Language and a Program for Solving and Solving Combinatorial Problems ». *Artificial Intelligence*, 10(1) :29–127, 1978.
- [LECOUTRE & CARDON 2005] Christophe LECOUTRE et Stéphane CARDON. « A greedy approach to establish singleton arc consistency ». Dans *Proceedings of the 19th international joint conference on Artificial intelligence (IJCAI'05)*, pages 199–204, 2005.
- [LECOUTRE & HEMERY 2006] Christophe LECOUTRE et Fred HEMERY. « Une étude des supports résiduels pour la consistance d'arc ». Dans *Actes des Deuxièmes Journées Francophones de Programmation par Contraintes (JFPC'06)*, pages 267–276, 2006.
- [LECOUTRE & PROSSER 2006] Christophe LECOUTRE et P. PROSSER. « Maintaining Singleton Arc Consistency ». Dans *Third International Workshop on Constraint Propagation And Implementation held with CP'06(CPAI'06)*, pages 47–61, 2006.
- [LECOUTRE & TABARY 2006] Christophe LECOUTRE et Sébastien TABARY. « Abscon 109 : a generic CSP solver ». Dans *Second International Constraint Solver Competition (CSC'06-CP'06)*, pages 55–63, 2006.
- [LECOUTRE & TABARY 2008] Christophe LECOUTRE et Sébastien TABARY. « Abscon 112 : towards more robustness ». Dans *3rd International Constraint Solver Competition (CSC'08)*, pages 41–48, 2008.
- [LECOUTRE *et al.* 2003] Christophe LECOUTRE, Frédéric BOUSSEMART, et Fred HEMERY. « Exploiting multidirectionality in coarse-grained arc-consistency algorithms ». Dans *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP'03)*, volume 2833, pages 480–494, 2003.
- [LECOUTRE *et al.* 2007] Christophe LECOUTRE, Lakhdar SAÏS, et Julien VION. « Using SAT Encodings to derive CSP value ordering heuristics ». *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)*, 1 :169–186, 2007.
- [LECOUTRE *et al.* 2008] Christophe LECOUTRE, Chavalit LIKITVIVATANAVONG, Scott SHANNON, Roland YAP, et Yuanlin ZHANG. « Maintaining Arc Consistency with Multiple Residues ». *Constraint Programming Letters*, 2 :3–19, 2008.
- [LECOUTRE 2009] Christophe LECOUTRE. *Constraint Networks : Techniques and Algorithms*. International Scientific and Technical Encyclopedia (ISTE Ltd) - John Wiley Inc., 2009.

-
- [LI & RENZ 2010] Jason Jingshi LI et Jochen RENZ. « In Defense of Large Qualitative Calculi ». Dans *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10)*, page to appear, 2010.
- [LI et al. 2009] Jason Jingshi LI, Jinbo HUANG, et Jochen RENZ. « A divide-and-conquer approach for solving interval algebra networks ». Dans *Proceedings of the 21st international joint conference on Artificial intelligence (IJCAI'09)*, pages 572–577, 2009.
- [LIGOZAT & RENZ 2004] Gérard LIGOZAT et Jochen RENZ. « What Is a Qualitative Calculus ? A General Framework ». Dans *Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence (PRICAI'04)*, pages 53–64, 2004.
- [LIGOZAT 1991] Gérard LIGOZAT. « On Generalized Interval Calculi ». Dans *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI'91)*, pages 234–240, 1991.
- [LIGOZAT 1994] Gérard LIGOZAT. « Tractable relations in temporal reasoning : pre-convex relations ». Dans *Proceedings of the ECAI'94 Workshop on Spatial and Temporal Reasoning*, pages 99–108, 1994.
- [LIGOZAT 1998a] Gérard LIGOZAT. « "Corner" Relations in Allen's algebra ». *Constraints*, 3(2/3) :165–177, 1998.
- [LIGOZAT 1998b] Gérard LIGOZAT. « Reasoning about Cardinal Directions ». *Journal of Visual Languages and Computing*, 1(9) :23–44, 1998.
- [MACKWORTH 1977] Alan K. MACKWORTH. « Consistency in Networks of Relations ». *Artificial Intelligence*, 8(1) :99–118, 1977.
- [MARRIOTT 1993] Kim MARRIOTT. « Frameworks for abstract interpretation ». *Acta Informatica*, 30 :103–129, 1993.
- [MERCHEZ et al. 2006] Sylvain MERCHEZ, Christophe LECOUTRE, et Frédéric BOUSSE-MART. « Abstraction de réseaux de contraintes ». *Revue d'Intelligence Artificielle*, 20(1) :31–62, 2006.
- [MITRA 2002] Debasis MITRA. « A Class of Star-Algebras for Point-Based Qualitative Reasoning in Two-Dimensional Space ». Dans *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference*, pages 486–491, 2002.
- [MOHR & HENDERSON 1986] Roger MOHR et Thomas C. HENDERSON. « Arc and Path Consistency Revisited ». *Artificial Intelligence*, 28(2) :225–233, 1986.
- [MOHR & MASINI 1988] Roger MOHR et Gérald MASINI. « Good Old Discrete Relaxation ». Dans *Proceedings of the 8th European Conference on Artificial Intelligence (ECAI'88)*, pages 651–656, 1988.
- [MONTANARI 1974] Ugo MONTANARI. « Networks of constraints : Fundamental properties and applications to picture processing ». *Information Sciences*, 7 :95–132, 1974.
- [NEBEL & BÜRCKERT 1995] Bernhard NEBEL et Hans-Jürgen BÜRCKERT. « Reasoning about temporal relations : a maximal tractable subclass of Allen's interval algebra ». *Journal of the ACM*, 42(1) :43–66, 1995.

- [NEBEL 1997] Bernhard NEBEL. « Solving Hard Qualitative Temporal Reasoning Problems : Evaluating the Efficiency of Using the ORD-Horn Class ». *Constraints*, 1(3) :175–190, 1997.
- [NÖKEL 1991] Klaus NÖKEL. *Temporally distributed symptoms in technical diagnosis*. Springer-Verlag New York, Inc., 1991.
- [PAPADIMITRIOU 1994] Christos M. PAPADIMITRIOU. *Computational complexity*. Addison-Wesley, 1994.
- [PHAM *et al.* 2006] Duc Nghia PHAM, John THORNTON, et Abdul SATTAR. « Towards an Efficient SAT Encoding for Temporal Reasoning ». Dans *Proceedings of the 12th International Conference on Principles and Practice of Constraint Programming (CP'06)*, pages 421–436, 2006.
- [PHAM *et al.* 2008] Duc Nghia PHAM, John THORNTON, et Abdul SATTAR. « Modelling and solving temporal reasoning as propositional satisfiability ». *Artificial Intelligence*, 172(15) :1752–1782, 2008.
- [PUJARI *et al.* 1999] Arun K. PUJARI, G. Vijaya KUMARI, et Abdul SATTAR. « INDU : An interval and duration network ». Dans *Australian Joint Conference on Artificial Intelligence*, pages 291–303, 1999.
- [RANDELL & COHN 1989] David A. RANDELL et Anthony G. COHN. « Modelling Topological and Metrical Properties in Physical Processes ». Dans *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'89)*, pages 55–66, 1989.
- [RANDELL *et al.* 1992] David A. RANDELL, Zhan CUI, et Anthony G. COHN. « A Spatial Logic Based on Regions and Connection ». Dans *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, pages 165–176, 1992.
- [REGIN 1999] Jean-Charles REGIN. « The symmetric alldiff constraint ». Dans *Proceedings of the 16th international joint conference on Artificial intelligence (IJCAI'99)*, pages 420–425, 1999.
- [RENTZ & LIGOZAT 2005] Jochen RENTZ et Gérard LIGOZAT. « Weak Composition for Qualitative Spatial and Temporal Reasoning ». Dans *Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming (CP'05)*, pages 534–548, 2005.
- [RENTZ & NEBEL 2001] Jochen RENTZ et Bernhard NEBEL. « Efficient methods for qualitative spatial reasoning ». *Journal of Artificial Intelligence Research*, 15(1) :289–318, 2001.
- [ROBINSON 1965] John A. ROBINSON. « A Machine-Oriented Logic Based on the Resolution Principle ». *Journal of the ACM*, 12(1) :23–41, 1965.
- [ROSSI *et al.* 1990] Francesca ROSSI, Charles J. PETRIE, et Vasant DHAR. « On the Equivalence of Constraint Satisfaction Problems ». Dans *Proceedings of the 9th European Conference on Artificial Intelligence (ECAI'90)*, pages 550–556, 1990.
- [ROUSSEL 2005] Olivier ROUSSEL. « Une autre conversion de SAT vers CSP ». Dans *Premières Journées Francophones de la Programmation par Contraintes (JFPC'05)*, pages 49–57, 2005.

-
- [SAADE 2008] Mahmoud SAADE. « *Étude du raisonnement temporel basé sur la résolution de contraintes* ». PhD thesis, Université d'Artois, 2008.
- [SABIN & FREUDER 1994] Daniel SABIN et Eugene C. FREUDER. « Contradicting Conventional Wisdom in Constraint Satisfaction ». Dans *Proceedings of the 11th European Conference on Artificial Intelligence (ECAI'94)*, pages 125–129, 1994.
- [SADEH & FOX 1996] Norman SADEH et Mark S. FOX. « Variable and value ordering heuristics for the job shop scheduling constraint satisfaction problem ». *Artificial Intelligence*, 86(1) :1–41, 1996.
- [SELMAN *et al.* 1992] Bart SELMAN, Hector J. LEVESQUE, et David G. MITCHELL. « A New Method for Solving Hard Satisfiability Problems ». Dans *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI'92)*, pages 440–446, 1992.
- [SELMAN *et al.* 1995] Bart SELMAN, Henry KAUTZ, et Bram COHEN. « Local Search Strategies for Satisfiability Testing ». Dans *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 521–532, 1995.
- [SHRAG & MIRANKER 1996] Robert SHRAG et Daniel MIRANKER. « Abstraction and the CSP phase transition boundary ». Dans *Proceedings of the 4th International Symposium on Artificial Intelligence and Mathematics*, pages 138–141, 1996.
- [SINGH 1996] Moninder SINGH. « Path Consistency Revisited ». *International Journal on Artificial Intelligence Tools*, 5(1-2) :127–141, 1996.
- [SMITH & GRANT 1998] Barbara M. SMITH et Stuart A. GRANT. « Trying Harder to Fail First ». Dans *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI'98)*, pages 249–253, 1998.
- [STERGIOU & WALSH 1999] Kostas STERGIOU et Toby WALSH. « Encodings of non-binary constraint satisfaction problems ». Dans *Proceedings of the sixteenth National Conference on Artificial intelligence (AAAI '99)*, pages 163–168, 1999.
- [TAILLARD 1993] Éric TAILLARD. « Benchmarks for basic scheduling problems ». *European Journal of Operational Research*, 64(2) :278–285, 1993.
- [TAMURA *et al.* 2008] Naoyuki TAMURA, Tomoya TANJO, et Mutsunori BANBARA. « System Description of a SAT-based CSP Solver Sugar ». Dans *3rd International Constraint Solver Competition (CSC'08)*, pages 71–75, 2008.
- [TAMURA *et al.* 2009] Naoyuki TAMURA, Akiko TAGA, Satoshi KITAGAWA, et Mutsunori BANBARA. « Compiling finite linear CSP into SAT ». *Constraints*, 14(2) :254–272, 2009.
- [VALDÉS-PÉREZ 1987] Raúl E. VALDÉS-PÉREZ. « The Satisfiability of Temporal Constraint Networks ». Dans *Proceedings of the 4th National Conference on Artificial Intelligence (AAAI'87)*, pages 256–260, 1987.

- [VAN BEEK & COHEN 1990] Peter van BEEK et Robin COHEN. « Exact and approximate reasoning about temporal relations ». *Computational Intelligence*, 6(3) :132–147, 1990.
- [VAN BEEK & DECHTER 1995] Peter van BEEK et Rina DECHTER. « On the minimality and global consistency of row-convex constraint networks ». *Journal of the ACM*, 42(3) :543–561, 1995.
- [VAN BEEK & MANCHAK 1996] Peter van BEEK et Dennis W. MANCHAK. « The design and experimental analysis of algorithms for temporal reasoning ». *Journal of Artificial Intelligence Research*, 4(1) :1–18, 1996.
- [VILAIN & KAUTZ 1986] M. VILAIN et H. KAUTZ. « Constraint Propagation Algorithms for Temporal Reasoning ». Dans *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI'86)*, pages 377–382, 1986.
- [VILAIN *et al.* 1990] Marc VILAIN, Henry KAUTZ, et Peter van BEEK. « Constraint propagation algorithms for temporal reasoning : a revised report ». *Readings in qualitative reasoning about physical systems*, pages 373–381, 1990.
- [WALLGRÜN *et al.* 2007] Jan Oliver WALLGRÜN, Lutz FROMMBERGER, Diedrich WOLTER, Frank DYLLA, et Christian FREKSA. « Qualitative spatial representation and reasoning in the SparQ-toolbox ». Dans *Proceedings of the 2006 international conference on Spatial Cognition V (SC'06)*, pages 39–58, 2007.
- [WALSH 2000] Toby WALSH. « SAT v CSP ». Dans *Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming (CP'00)*, pages 441–456, 2000.
- [WESTPHAL & WÖLFL 2009] Matthias WESTPHAL et Stefan WÖLFL. « Qualitative CSP, finite CSP, and SAT : comparing methods for qualitative constraint-based reasoning ». Dans *Proceedings of the 21st international joint conference on Artificial intelligence (IJCAI'09)*, pages 628–633, 2009.
- [WESTPHAL *et al.* 2010] Matthias WESTPHAL, Stefan WÖLFL, et Jason Jingshi LI. « Restarts and Nogood Recording in Qualitative Constraint-based Reasoning ». Dans *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI'10)*, pages 1093–1094, 2010.
- [WILLIAMS *et al.* 2003] Ryan WILLIAMS, Carla P. GOMES, et Bart SELMAN. « Backdoors to typical case complexity ». Dans *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 1173–1178, 2003.

Résumé

Mots-clés: Intelligence Artificielle, problème CSP, problème SAT, Raisonnement Qualitatif, Transformation, Classes traitables, Relaxation, Substituabilité

La modélisation et la résolution de problèmes sous contraintes constituent un domaine majeur en IA. Par la nature diverse des contraintes, différents formalismes de représentation ont été proposés pour les exprimer de manière simple et compacte tout en garantissant une efficacité des outils de résolutions associés. Les formules propositionnelles, les réseaux de contraintes discrets (RCD) et qualitatives (RCQ) sont des cadres de modélisation répondant à ces critères. Pour les informations temporelles ou spatiales, les RCQ constituent un modèle de choix avec de nombreuses applications comme l'ordonnancement de tâches, la planification temporelle ou spatiale, les systèmes d'informations géographiques. Nos contributions visent à étudier les liens des RCQ vers les RCD et les formules propositionnelles, afin d'adapter les outils issus des divers domaines et de proposer de nouvelles approches. Tout d'abord, nous nous concentrons sur l'aspect structurel des RCQ, en adaptant la méthode de la composition faible dans les différents cadres. Nous exploitons ensuite les propriétés des classes traitables de certains formalismes qualitatifs, afin de définir une transformation vers la logique propositionnelle. En exploitant la transformation vers les RCD, nous proposons une méthode incomplète facilitant la preuve de l'incohérence des RCQ par la relaxation de la propriété de composition faible, puis nous complétons l'approche en exploitant les classes traitables. Enfin, ces études nous conduisent à proposer une nouvelle forme de substituabilité locale, dont les détections statique et dynamique permettent d'obtenir une amélioration algorithmique dans le cadre des RCD.

Abstract

Keywords: Artificial Intelligence, CSP, SAT, Qualitative Reasoning, Encoding, Tractable Classes, Relaxation, Substitutability

Modelling and solving constraints problems is a major domain in Artificial Intelligence. By the various natures of the constraints, different formalisms were proposed to express them in a simple and compact way while guaranteeing the effectiveness of the associated solution tools. Propositional formulae, discrete constraint networks (DCNs), and qualitative constraint networks (QCNs) are the well known frameworks that guaranty these requirements. For temporal or space information, QCNs constitute a model of choice with many real world applications such as scheduling, temporal or spatial planning and geographic information systems. Our contributions aim at studying the links between QCNs, DCNs and propositional formulas, in order to adapt the tools developed in these fields and to propose new approaches. First of all, we focus on the structural aspects of QCNs, by transforming weak composition within the various frameworks. In order to define a transformation towards propositional logic we then exploit the properties of tractable classes of some qualitative formalism. Exploiting the transformation towards DCNs, we propose an incomplete method simplifying the proof of the inconsistency for QCNs by relaxing the weak composition property. Then, we propose a complete approach thanks to tractable classes. Finally, these studies lead us to propose a new form of local substitutability, whose static and dynamic detections significantly improve search algorithms for DCNs.

