Cohérence de réseaux de contraintes qualitatives spatio-temporelles à partir de décompositions arborescentes

Jean-François Condotta

Université Lille-Nord de France, Artois CRIL-CNRS, UMR 8188
rue de l'université, SP 16, F62307 Lens
condotta@cril.fr

Résumé : Dans le cadre d'une traduction des réseaux de contraintes temporelles de l'Algèbre des Intervalles en problèmes SAT, Li et al. [7] ont récemment proposé une nouvelle décomposition de ces réseaux de contraintes afin de réduire le nombre de clauses SAT à résoudre. Dans cet article, nous montrons que ces décompositions sont équivalentes à des décompositions arborescentes. Dans un deuxième temps, nous considérons la propriété de fermeture par faible composition sur laquelle sont basées les méthodes de filtrage pour la résolution de réseaux de contraintes qualitatives (RCQ). Nous montrons que pour certaines classes de RCQ, la satisfaction de cette propriété par les clusters issus d'une décomposition arborescente est suffisante pour décider de la cohérence. À partir de ce résultat, nous proposons un nouvel algortithme permettant de décider du problème de la cohérence des RCQ à l'aide d'une classe traitable et d'une décomposition arborescente.

1 Introduction

Le raisonnement sur des informations temporelles et spatiales est une tâche cruciale dans certaines applications informatiques, en particulier dans des domaines tels que les SIG (Systèmes d'Informations Géographiques), le traitement du langage naturel, la planification temporelle/spatiale. Le raisonnement spatio-temporel qualitatif est un domaine dans lequel sont proposés des formalismes permettant de représenter et de traiter qualitativement les informations sur des configurations d'entités spatiales/temporelles. Un formalisme qualitatif considère des éléments particuliers pour représenter les différentes entités et un ensemble fini de relations de base sur ces entités. Chaque relation de base est représentative d'une position relative particulière entre des entités, et correspond à une abstraction de configurations pouvant être caractérisées de manière numérique. Ces dernières années de nombreux formalismes ont été proposés et étudiés, pour le raisonnement temporel nous pouvons par exemple citer l'Algèbre des Intervalles proposée par Allen [1]. Ce formalisme considère comme entités les intervalles de la droite et treize relations de base représentant toutes les configurations possibles des quatre

bornes de deux intervalles (voir la figure 1). L'algèbre des Intervalles a inspiré la définition de nombreux formalismes qualitatifs pour le temps [9, 10, 14]. Concernant le raisonnement spatial, le plus connu et étudié des formalismes qualitatifs est certainement RCC [15, 16]. RCC permet de représenter qualitativement des configurations de régions à l'aide de relations de base d'ordre topologique.

Dans le cadre des formalismes qualitatifs, l'ensemble des informations temporelles ou spatiales est habituellement représenté par des réseaux de contraintes particuliers appelés réseaux de contraintes qualitatives (RCQ en abrégé). Dans un RCQ, chaque variable représente une entité spatiale/temporelle du système et une contrainte définit les configurations possibles entre des entités à l'aide d'un ensemble de relations de base permises. Étant donné un RCQ, le principal problème se posant est de déterminer s'il admet une solution (le problème de la cohérence). Ce problème est en règle général un problème NP–Complet, de nombreuses études ont été réalisées afin de proposer des méthodes pour résoudre efficacement en pratique ce problème. Deux types d'approches sont suivies pour l'élaboration de ces méthodes.
La première approche, initiée par Nebel [11], consiste en la définition d'algorithmes de recherche de type backtrack combinés avec une propagation locale de contraintes basée sur des classes de relations dites traitables et la méthode de fermeture par faible composition. Cette dernière permet de maintenir le RCQ $\diamond$–cohérent (propriété proche de la chemin-cohérence). À chaque étape de la recherche, une contrainte est découpée en sous-relations de la classe traitable utilisée. Puis, pour chacune des sous-relations issues de ce découpage, la contrainte est remplacée par la sous-relation et un filtrage est réalisé à l'aide de la fermeture par faible composition. Ce filtrage permet de supprimer des relations de base non satisfiables et de réduire l'espace de recherche. L'utilisation d'une classe traitable dans ce processus permet de réduire le facteur de branchement lors de la recherche. Les solveurs de contraintes qualitatives utilisent en général cette approche, en particulier GQR* [5] qui est actuellement le plus performant.

Une autre approche plus récente [13] consiste en la traduction des RCQ en SAT (problème de satisfiabilité propositionnelle). Typiquement, les problèmes SAT issus de telles traductions contiennent d'une part des clauses modélisant les contraintes explicites du RCQ, et d'autre part des clauses modélisant les configurations possibles compte tenu de l'opération de faible composition sur laquelle est basée la fermeture par faible composition. L'avantage d'une telle approche est que les instances SAT obtenues peuvent être résolues de manière efficace par les solveurs SAT actuels. Un inconvénient majeur est que ces instances SAT sont de tailles très importantes. Récemment, pour tenter de pallier cet inconvénient dans le cadre de RCQ de l'Algèbre des Intervalles, Li *et al*. [7] ont proposé une méthode décomposant récursivement à chaque étape un RCQ en deux RCQ équivalents portant chacun sur moins de variables. Les contraintes présentes dans le RCQ initial et non présentes dans l'un des deux RCQ sont caractérisées comme non nécessaires à la recherche d'une solution et ne sont donc pas considérées lors d'un processus de traduction en SAT. Les instances SAT obtenues en utilisant cette méthode sont de moindre tailles comparativement à des traductions complètes. De plus, Li *et al*. ont montré expérimentalement que ces instances étaient résolues plus rapidement.

Dans cet article, nous définissons formellement des décompositions de RCQ appelées décompositions de type PartRec. Ces décompositions particulières permettent de re-

présenter les décompositions de RCQ issues de l'approche suivie par Li *et al*. dans leur processus de traduction SAT. Nous montrons que les décompositions de type PartRec peuvent de manière équivalente se définir à l'aide de décompositions arborescentes de RCQ, décompositions intensivement étudiées dans le cadre des CSP discrets. Dans un second temps, nous étudions le problème de la cohérence des RCQ au travers des décompositions arborescentes. Nous montrons notamment qu'étant donnée une décomposition arborescente d'un RCQ définie sur un classe traitable possédant certaines propriétés, la fermeture par faible composition restreinte aux contraintes appartenant aux différents clusters est suffisante pour décider du problème de la cohérence du RCQ. Fort de ce résultat, nous proposons un algorithme permettant de décider du problème de la cohérence des RCQ à l'aide d'une classe traitable et d'une décomposition arborescente.

2 Préliminaires sur les formalismes qualitatifs

2.1 Les relations

Un formalisme qualitatif considère un ensemble fini B de relations appelées relations de base ou encore relations atomiques. Ces relations sont définies sur un domaine D et nous supposerons dans la suite que ces relations sont binaires. Il est à noter que peu de formalismes qualitatifs considèrent des relations de base d'arité supérieure à deux. Les éléments de D sont utilisés pour représenter les entités temporelles ou spatiales. L'ensemble B correspond à toutes les configurations qualitatives possibles entre deux entités temporelles ou spatiales et satisfait des propriétés particulières [17] : (1) B forme une partition de $\mathsf{D} \times \mathsf{D}$, (2) B contient la relation identité sur D et, (3) B est fermé pour l'inverse, *i.e*. la relation inverse d'une relation de base de B appartient également à B. Une relation (complexe) d'un formalisme qualitatif correspond à l'union de relations de base. Une relation est représentée par l'ensemble des relations de base incluses dans la relation. Ainsi, l'ensemble 2^{B} représentera l'ensemble des relations du formalisme qualitatif considéré. Parmi les relations de 2^{B} nous dénoterons par Ψ la relation contenant toutes les relations de base de B. Comme ensemble d'ensembles, 2^{B} est muni des opérations ensemblistes d'intersection et d'union. Comme ensemble de relations binaires il est également muni de l'opération inverse ($^{-1}$). L'inverse d'une relation $r \in 2^{\mathsf{B}}$ est l'union des relations de base inverses des relations de base appartenant à r. L'ensemble 2^{B} est aussi muni de l'opération binaire de faible composition, notée dans la suite par $\diamond$. Cette opération peut être définie de la manière suivante : $a \diamond b = \{c \in \mathsf{B} : \exists x, y, z \in \mathsf{D}$ avec $x\ a\ z, z\ b\ y$ et $x\ c\ y\}$, avec $a, b \in \mathsf{B}, r \diamond s = \bigcup_{a \in r, b \in s} \{a \diamond b\}$, avec $r, s \in 2^{\mathsf{B}}$. Remarquons que la relation $r \diamond s$ peut être également définie par la plus petite (pour l'inclusion) relation appartenant à 2^{B} incluant la composition relationnelle habituelle $r \circ s = \{(x, y) \in \mathsf{D} \times \mathsf{D} : \exists x, y, z \in \mathsf{D}$ avec $x\ a\ z, z\ b\ y$ et $x\ c\ y\}$. Pour certains formalismes qualitatifs $r \circ s$ et $r \diamond s$ correspondent à la même relation. Une classe de relations $\mathcal{C}$ est un sous-ensemble de 2^{B} fermé pour les opérations d'inverse, d'intersection et de faible composition, contenant les relations singletons de 2^{B} et la relation Ψ. Étant données une relation $r \in 2^{\mathsf{B}}$ et une classe de relations $\mathcal{C}$, la plus petite (pour l'inclusion) relation de $\mathcal{C}$ contenant r sera dénotée par $r^{\mathcal{C}}$ et sera appelée la fermeture de r par rapport à $\mathcal{C}$.

Comme illustration, considérons le formalisme qualitatif temporel bien connu appelé Algèbre des Intervalles [1]. Pour ce formalisme, les intervalles de la droite des rationnels sont utilisés pour représenter les entités temporelles. Ainsi, D est l'ensemble $\{(x^-, x^+) \in \mathbb{Q} \times \mathbb{Q} : x^- < x^+\}$. L'ensemble des relations de base consiste en un ensemble de treize relations binaires, chacune correspondant à une position relative particulière de deux intervalles de la droite. Ces relations sont illustrées dans la figure 1. Nous avons $\mathsf{B} = \{eq, p, pi, m, mi, o, oi, s, si, d, di, f, fi\}$. Chacune de ces relations de base peut être définie par des contraintes sur les bornes de deux intervalles. Par exemple, la relation de base *meets* (m) peut être définie de la manière suivante, $m = \{((x^-, x^+), (y^-, y^+)) \in \mathsf{D} \times \mathsf{D} : x^+ = y^-\}$. À titre d'exemple, l'ensemble $\{p, m\} \in 2^{\mathsf{B}}$ correspond à la relation $p \cup m$. Nous avons, $\{p, m\}^{-1} = \{p^{-1}, m^{-1}\} = \{pi, mi\}$ et $\{p, m\} \diamond \{m, s, eq\} = \{p, m\}$.

Relation	Symbole	Inverse	Illustration
precedes	p	pi	
meets	m	mi	
overlaps	o	oi	
starts	s	si	
during	d	di	
finishes	f	fi	
equals	eq	eq	

FIG. 1 – Les relations de base de l'Algèbre des Intervalles.

2.2 Les réseaux de contraintes qualitatives

Un réseau de contraintes qualitatives (RCQ) est un couple composé d'un ensemble de variables représentant les entités temporelles du système et d'un ensemble de contraintes défini par une application associant à chaque couple de variables un ensemble de relations de base (les configurations possibles). Formellement, un RCQ est défini de la manière suivante :

Définition 1 *Un* RCQ $\mathcal{N}$ *est un couple* (V, C) *où :*

- $V = \{v_1, \ldots, v_m\}$ *est un ensemble fini de* m *variables avec* m *un entier positif ;*
- C *est une application associant à chaque couple* (v_i, v_j) *de* $V \times V$ *un ensemble* $C(v_i, v_j)$ *de relations de base :* $C(v_i, v_j) \in 2^{\mathsf{B}}$. *Dans la suite,* $C(v_i, v_j)$ *sera également dénoté par* $\mathcal{N}[v_i, v_j]$. C *est telle que* $C(v_i, v_i) \subseteq \{\mathsf{Id}\}$, *avec* Id *la relation de base correspondant à la relation d'identité sur* D *et* $C(v_i, v_j) = C_(v_j, v_i)^{-1}$ *pour tout* $v_i, v_j \in V$.

Dans la figure 2(a) est représenté un RCQ $\mathcal{N}$ de l'Algèbre des Intervalles. Sur la figure ne sont par représentées les contraintes définies par la relation Ψ et la contrainte entre deux variables v_i et v_j lorsque $v_i = v_j$ ou lorsque la contrainte entre v_j et v_i est déjà présente. Concernant un RCQ $\mathcal{N} = (V, C)$ nous avons les notations et définitions suivantes : Une *instanciation partielle* de $\mathcal{N}$ sur $V' \subseteq V$ est une application s de V' vers D. Une *solution partielle* de $\mathcal{N}$ sur $V' \subseteq V$ est une instanciation partielle sur V' telle que $(s(v_i), s(v_j))$ satisfait $C(v_i, v_j)$ pour tout $v_i, v_j \in V'$, *i.e.* il existe une relation de base $b \in C(v_i, v_j)$ telle que $(s(v_i), s(v_j)) \in b$ pour tout $v_i, v_j \in V'$. Une *solution* de $\mathcal{N}$ est une solution partielle de $\mathcal{N}$ sur V. $\mathcal{N}$ est cohérent si, et seulement si, $\mathcal{N}$ admet une solution. $\mathcal{N}$ est dit trivialement incohérent dans le cas où il existe deux variables $v, v' \in V$ telles que $\mathcal{N}[v, v'] = \emptyset$. $\mathcal{N}$ est globalement cohérent si, et seulement si, toute solution partielle de $\mathcal{N}$ peut être étendue en une solution de $\mathcal{N}$. La projection du RCQ $\mathcal{N}$ sur V' avec $V' \subseteq V$, dénotée par $\mathcal{N}_{V'}$, est le RCQ (V', C_{proj}) avec C_{proj} la restriction de C sur V'. Un sous-RCQ $\mathcal{N}'$ de $\mathcal{N}$ est un RCQ (V, C') tel que $C'(v_i, v_j) \subseteq C(v_i, v_j)$, pour tout $v_i, v_j \in V$. Soient deux RCQ $\mathcal{N}^1$ et $\mathcal{N}^2$ définis respectivement sur les ensembles de variables V^1 et V^2, avec pour chaque couple de variables $v, v' \in V^1 \cap V^2, \mathcal{N}^1[v, v'] = \mathcal{N}^2[v, v']$. Le RCQ $\mathcal{N}^1 \cup \mathcal{N}^2$ est l'unique RCQ $\mathcal{N}$ défini sur l'ensemble de variables $V^1 \cup V^2$ tel que $\mathcal{N}[v, v'] = \mathcal{N}^1[v, v']$ pour tout $v, v' \in V^1, \mathcal{N}[v, v'] = \mathcal{N}^2[v, v']$ pour tout $v, v' \in V^2, \mathcal{N}[v, v'] = \Psi$ pour tout $v \in V^2 \setminus V^1$ et $v' \in V^1 \setminus V^2$.

Un RCQ $\mathcal{N} = (V, C)$ est dit $\diamond$-cohérent ou fermé par faible composition si, et seulement si, $C(v_i, v_j) \subseteq C(v_i, v_k) \diamond C(v_k, v_j)$ pour tout $v_i, v_j, v_k \in V$. La fermeture par faible composition du RCQ $\mathcal{N}$, dénotée par $\diamond(\mathcal{N})$ est le plus grand (pour $\subseteq$) $\diamond$-cohérent sous-RCQ de $\mathcal{N}$. Cette fermeture peut-être obtenue par itération de l'opération de triangulation : $C(v_i, v_j) \leftarrow C(v_i, v_j) \cap (C(v_i, v_k) \diamond C(v_k, v_j))$ pour tout $v_i, v_j, v_k \in V$ jusqu'à ce qu'un point fixe soit atteint. Cette méthode peut-être implémentée avec des algorithmes de complexité temporelle $O(m^3)$, avec m le nombre de variables. Pour certaines classes de relations, le problème de la cohérence d'un RCQ peut-être résolu par la fermeture par faible composition. Pour l'Algèbre des Intervalles, nous avons par exemple tout RCQ $\mathcal{N}$ $\diamond$-cohérent (et non trivialement incohérent) dont les contraintes sont définies par des relations de la classe des relations préconvexes, appelées également relations ORD-Horn [12, 8], qui est cohérent. Pour la classe des relations convexes de l'Algèbre des Intervalles nous avons une propriété plus forte, en effet tout RCQ $\mathcal{N}$ $\diamond$-cohérent défini par des relations de cette classe est globalement cohérent [2].

Pour clore cette section nous donnons quelques notations concernant les arbres. Étant donné un arbre (un graphe acyclique connecté) enraciné $T = (X, F)$ et un noeud $X_i \in X$, nous dénoterons par $\mathsf{desc}(X_i)$ (resp. $\mathsf{asc}(X_i)$) l'ensemble des descendants de X_i (resp. des ascendants de X_i). Notons que $\mathsf{desc}(X_i)$ et $\mathsf{asc}(X_i)$ contiennent X_i. Étant donné X' un sous-ensemble non vide de noeuds de X, $\mathsf{lca}(X')$ dénotera le noeud de T qui est le plus bas ascendant commun des noeuds appartenant à X'. L'ensemble $\mathsf{leaves}(T)$ correspond à l'ensemble des feuilles de T. Et enfin, étant donné $X_i \in X$, T_{X_i} dénotera le sous-arbre de T de racine X_i.

3 Décompositions de RCQ

3.1 Décompositions arborescentes

La relation Ψ est composée de toutes les relations de base possibles de B et est satisfaite par toute paire d'éléments du domaine D. La définition d'une contrainte entre deux variables d'un RCQ à l'aide de la relation Ψ permet de spécifier que localement il n'y a pas de contrainte concernant la position relative des deux entités représentées par les deux variables. Ainsi, de manière naturelle nous définissons le graphe de contraintes d'un RCQ $\mathcal{N} = (V, C)$, par le graphe non orienté $\mathsf{G}(\mathcal{N}) = (V, E)$ avec $(v, v') \in E$ si, et seulement si, $\mathcal{N}[v, v'] \neq \Psi$ et $v \neq v'$. Dans la suite nous supposons qu'étant donné un RCQ $\mathcal{N}$, $\mathsf{G}(\mathcal{N})$ est connecté. Dans le cas contraire, $\mathcal{N}$ peut être trivialement partagé en deux RCQ indépendants sans variable commune. Comme dans le cadre des CSP discrets nous définissons une décomposition arborescente d'un RCQ comme une décomposition arborescente de son graphe de contraintes [18] :

Définition 2 *Soit $\mathcal{N} = (V, C)$ un* RCQ *et $\mathsf{G}(\mathcal{N}) = (V, E)$ son graphe de contraintes. Une décomposition arborescente de $\mathcal{N}$ est un arbre $T = (X = \{X_0, \ldots, X_n\}, F)$ avec n un entier positif, où X est une famille de sous-ensembles de variables de V ($X_i \subseteq V$), telle que :*
(1) $\bigcup\{X_i \in X\} = V$,
(2) $\forall (v, v') \in E$, il existe $X_i \in X$ tel que $v, v' \in X_i$,
(3) pour tout $X_i, X_j, X_k \in X$, si X_j est sur l'unique chemin entre X_i et X_k alors $X_i \cap X_k \subseteq X_j$.

Étant donnée une décomposition arborescente $T = (X = \{X_0, \ldots, X_n\}, F)$ d'un RCQ, la largeur de T est égale à $max\{|X_i| - 1 : X_i \in X\}$. De plus, chaque ensemble de variables X_i est appelé cluster. Dans la figure 2(c) se trouve représentée une décomposition arborescente du RCQ $\mathcal{N}$ de la figure 2(a).

3.2 Les décompositions de type PartRec

Récemment, Li *et al*. [7] ont proposé une méthode permettant de traduire un RCQ de l'Algèbre des Intervalles en problème SAT. Cette méthode décompose récursivement à chaque étape un RCQ $\mathcal{N} = (V, C)$ en deux RCQ $\mathcal{N}^1$ et $\mathcal{N}^2$ tels que $\mathcal{N} = \mathcal{N}^1 \cup \mathcal{N}^2$. Les contraintes définies par la relation Ψ dans le RCQ $\mathcal{N}$ non présentes dans $\mathcal{N}^1$ et $\mathcal{N}^2$ sont caractérisées comme non nécessaires à la recherche d'un scénario cohérent du RCQ global initial et donc non traduites en SAT. L'avantage d'une telle traduction est que l'instance SAT obtenue par cette méthode est de plus petite taille qu'une instance issue d'une traduction complète de l'ensemble des contraintes. En nous inspirant de cette méthode, nous définissons des décompositions de RCQ particulières, appelées dans la suite décompositions PartRec (pour partitionnement récursif), de la manière suivante :

Définition 3 *Soit $\mathcal{N}$ un* RCQ $= (V, C)$. *Une décomposition* PartRec *de $\mathcal{N}$ est un arbre enraciné $T = (X = \{X_0, \ldots, X_n\}, F)$, avec n un entier positif, où X est une famille de sous-ensembles de V ($X_i \subseteq V$) et $X_r = V$ avec X_r la racine de T. De plus, chaque*

noeud de T a deux fils ou aucun et étant donnés $X_i, X_j, X_k \in X$ tels que X_j et X_k sont fils de X_i, T satisfait les propriétés suivantes :

(1) $\mathcal{N}_{X_i} = \mathcal{N}_{X_j} \cup \mathcal{N}_{X_k}$, $X_j \setminus X_k \neq \emptyset$ *et* $X_k \setminus X_j \neq \emptyset$.

(2) *Pour tout* $X_m \in \mathsf{desc}(X_i)$, *si* $X_m \cap (X_j \cap X_k) \neq \emptyset$ *alors* $(X_j \cap X_k) \subseteq X_m$.

Intuitivement, d'après la condition (1) nous avons les contraintes pertinentes sur l'ensemble des variables X_i qui appartiennent à l'ensemble des contraintes sur X_j ou à celui des contraintes sur X_k, ou aux deux. De plus, l'ensemble X_j ou l'ensemble X_k ne regroupe pas la totalité des variables de X_i. La condition (2) stipule que toute contrainte entre deux variables appartenant à l'intersection de deux RCQ issus d'un découpage ne pourra pas être scindée lors d'un nouveau découpage. Dans la figure 2(b) est représentée une décomposition PartRec du RCQ $\mathcal{N}$ de la figure 2(a). Notons que le traitement récursif employé dans le cadre de la traduction des RCQ en clauses SAT proposée par Li *et al*. [7] définit implicitement une décomposition $T = (X, T)$ de type PartRec du RCQ $\mathcal{N}$ considéré. Les ensembles de variables $X_i \in X \setminus \mathsf{leaves}(T)$ correspondent aux ensembles de variables partagés au cours du traitement. Les ensembles de variables non partagés pour lesquels le traitement s'arrête correspondent aux éléments $X_i \in \mathsf{leaves}(T)$. Seules les contraintes portant sur des variables appartenant à un même ensemble $X_i \in \mathsf{leaves}(T)$ sont traduites en clauses SAT. Li *et al*. donne un algorithme polynomial permettant de calculer une partition de type PartRec . En pratique, lors des découpages récursifs, un équilibre concernant le nombre des variables des deux RCQ obtenus et une minimisation des contraintes appartenant à l'intersection de ces deux RCQ permettent de maximiser le nombre de contraintes à laisser de côté.
Nous définissons la largeur d'une décomposition PartRec $T = (X, F)$ par $max\{|X_i| - 1 : X_i \in \mathsf{leaves}(T)\}$ et appelons cluster de T tout élément $X_i \in \mathsf{leaves}(T)$. Remarquons que du fait de la propriété (1) de la définition précédente, $X_i \neq \emptyset$ pour tout $X_i \in X$. De plus, une décomposition PartRec satisfait les propriétés suivantes :

Proposition 1 *Soit* $T = (X = \{X_0, \ldots, X_n\}, F)$ *une décomposition* PartRec *d'un* RCQ $\mathcal{N} = (V, C)$. *Pour tout* $X_i \in X$, *nous avons :*

(1) Pour tout $X_j \in X$ *tel que* $i \neq j$, $X_i \neq X_j$ *et de plus, si* $X_j \in \mathsf{desc}(X_i)$ *alors* $X_j \subset X_i$ *;*

(2) T_{X_i} *est une décomposition* PartRec *de* $\mathcal{N}_{X_i}$ *;*

(3) $\mathcal{N}_{X_i} = \bigcup\{\mathcal{N}_{X_j} : X_j \in \mathsf{leaves}(T_{X_i})\}$ *;*

(4) $X_i = \bigcup\{X_j \in \mathsf{leaves}(T_{X_i})\}$.

Les preuves de ces propriétés sont omises et peuvent être directement établies à partir de la définition 3. Notons que du fait de la propriété (1), nous avons pour tout $X_i, X_j \in X$ les ensembles X_i et X_j qui sont différents si, et seulement si, $i \neq j$. Ainsi, pour tout $X_i \in X$, l'ensemble des variables de X_i caractérise un et un seul noeud de X. Soit une décomposition PartRec $T = (X, F)$ d'un RCQ $\mathcal{N} = (V, C)$. Étant donné $X_i \in X$, $\overline{X_i}$ dénotera l'ensemble X_i dans le cas où $X_i \in \mathsf{leaves}(T)$, l'ensemble $X_j \cap X_k$ avec X_j et X_k les deux fils de X_i dans le cas où $X_i \notin \mathsf{leaves}(T)$. Notons que $\overline{X_i} \subseteq X_i$ puisque $X_j \subseteq X_i$ et $X_k \subseteq X_i$. De plus, nous avons les propriétés suivantes :

Proposition 2 *Soit une décomposition* PartRec $T = (X = \{X_0, \ldots, X_n\}, F)$ *d'un* RCQ $\mathcal{N} = (V, C)$.

(1) Pour tout $X_i \in X$, $\overline{X_i} \neq \emptyset$.
(2) Soient $X_i, X_j, X_k \in X$ *avec* $X_k = \mathsf{lca}(\{X_i, X_j\})$. *Si* $X_k \neq X_i$ *et* $X_k \neq X_j$ *alors* $X_i \cap X_j \subseteq \overline{X_k}$.
(3) Pour tout $X_i \in X$, *il existe* $X_j \in \mathsf{leaves}(X)$ *tel que* $\overline{X_i} \subseteq X_j$.

Pour une meilleure lisibilité, la preuve de cette proposition et celles des suivantes sont données en Annexe.

3.3 Des décompositions PartRec vers les décompositions arborescentes

Dans cette section nous allons montrer qu'à partir d'une décomposition PartRec T d'un RCQ nous pouvons définir une décomposition arborescente de même largeur qui, comme nous le verrons par la suite, considère les mêmes regroupements de variables. Avant cela, nous introduisons une nouvelle notation : étant donnés $X_i, X_j \in X$ avec X_j un noeud fils de X_i, $\widetilde{X_i X_j}$ dénotera l'ensemble $\{X_k : X_k \in \mathsf{desc}(X_j) \text{ et } \overline{X_i} \subseteq \overline{X_k}\}$. Concernant l'ensemble $\widetilde{X_i X_j}$, nous avons les propriétés suivantes.

Proposition 3 *Soit une décomposition* PartRec $T = (X, F)$ *d'un* RCQ $\mathcal{N} = (V, C)$ *et* $X_i, X_j \in X$ *avec* X_j *fils de* X_i. *Nous avons : (1)* $\widetilde{X_i X_j} \neq \emptyset$, *(2)* $\mathsf{lca}(\widetilde{X_i X_j}) \in \widetilde{X_i X_j}$.

Maintenant, nous définissons à partir d'une décomposition PartRec T d'un RCQ $\mathcal{N}$, un arbre dénoté par $\overline{T}$. Nous montrerons par la suite que cet arbre est une décomposition arborescente de $\mathcal{N}$ possédant certaines particularités.

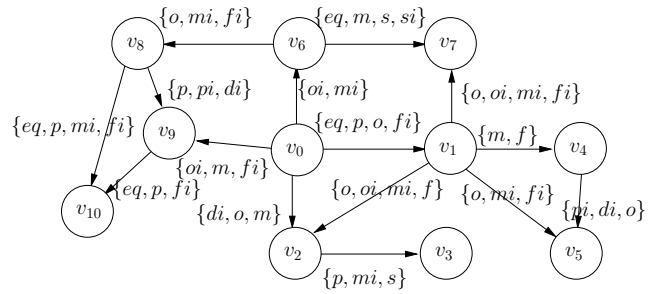
Définition 4 *Soient* $\mathcal{N}$ *un* RCQ *et une décomposition* PartRec $T = (X = \{X_0, \ldots, X_n\}, F)$ *de* $\mathcal{N}$. *À partir de* T *et d'un élément* $X_i \in X$, *nous définissons inductivement un arbre* $\overline{T}_{X_i} = (X_{X_i}, F_{X_i})$ *de racine* $\overline{X_i}$ *de la manière suivante :*
- *Cas de base :* $X_i \in \mathsf{leaves}(T)$, $\overline{T}_{X_i} = (\{\overline{X_i}\}, \emptyset)$.
- *Cas Inductif :* $X_i \notin \mathsf{leaves}(T)$. *Dans* T, *soient* X_j *et* X_k *les deux noeuds fils de* X_i, $X_l = \mathsf{lca}(\widetilde{X_i X_j})$ *et* $X_m = \mathsf{lca}(\widetilde{X_i X_k})$. X_{X_i} *et* F_{X_i} *sont définis par* $X_{X_i} = X_{X_j} \cup X_{X_k} \cup \{\overline{X_i}\}$, $F_{X_i} = F_{X_j} \cup F_{X_k} \cup \{(\overline{X_i}, \overline{X_l}), (\overline{X_i}, \overline{X_m})\}$.

$\overline{T}$ *est défini par l'arbre de racine* $\overline{T}_{X_r}$ *avec* $X_r \in X$ *la racine de* T.

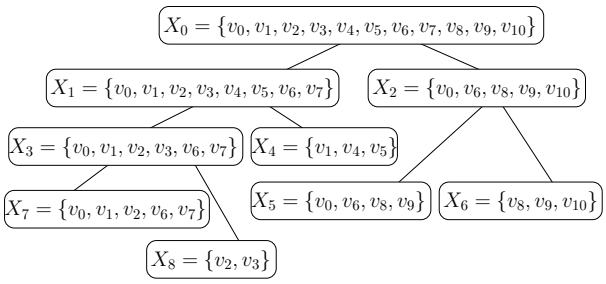
L'arbre $\overline{T}$ est une décomposition arborescente du RCQ $\mathcal{N}$ pour lequel T est une décomposition PartRec :

Proposition 4 *Soient* $\mathcal{N} = (V, C)$ *un* RCQ *et une décomposition* PartRec $T = (X = \{X_0, \ldots, X_n\}, F)$ *de* $\mathcal{N}$. *Nous avons :* $\overline{T} = (\overline{X} = (\overline{X} = \{\overline{X_0}, \ldots, \overline{X_n}\}, \overline{F})$ *est une décomposition arborescente de* $\mathcal{N}$ *telle que pour tout* $\overline{X_i} \in \overline{X}$ *il existe* $X_j \in \mathsf{leaves}(T)$ *avec* $\overline{X_i} \subseteq X_j$ *et telle que pour tout* $X_j \in \mathsf{leaves}(T)$ *il existe* $\overline{X_i} \in \overline{X}$ *avec* $\overline{X_i} = X_j$.

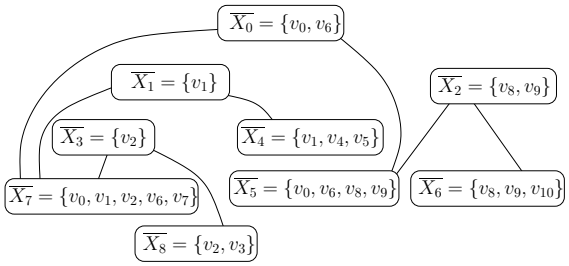
Ainsi, à toute décomposition T de type PartRec peut être associée une décomposition arborescente dont les clusters considèrent les mêmes regroupements de variables que ceux issus de $\mathsf{leaves}(T)$. Sous ce point de vue, une décomposition de type PartRec peut donc être considérée comme une décomposition arborescente particulière. À titre



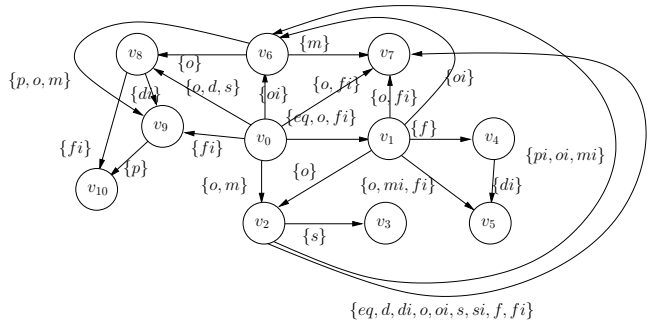
(a)



(b)



(c)



(d)

FIG. 2 – (a) un RCQ $\mathcal{N} = (V, C)$ de l'Algèbre des Intervalles, (b) une décomposition PartRec $T = (X, F)$ de $\mathcal{N}$, (c) une décomposition arborescente de $\mathcal{N}$ correspondant à $\overline{T}$, (d) un sous-RCQ de $\mathcal{N}$ ORD-Horn et $\overset{\diamond}{X}$–cohérent.

d'illustration, considérons le RCQ $\mathcal{N} = (V, C)$ de la figure 2(a) et $T = (X, F)$ une de ses décompositions de type PartRec illustrée dans la figure 2(b). La décomposition arborescente $\overline{T}$ est représentée dans la figure 2(c).

4 Décompositions arborescentes et cohérences de RCQ

Dans cette section, nous montrons que pour certains RCQ, une décomposition arborescente permet de réduire le nombre de contraintes à considérer dans la cadre de la résolution du problème de la cohérence. En particulier, nous montrons que pour certaines classes de relations, la fermeture par faible composition restreinte aux clusters issus d'une décomposition arborescente est complète pour le problème de la cohérence. Tout d'abord, nous introduisons une nouvelle cohérence locale pour les RCQ correspondant à la propriété de $\diamond$-cohérence restreinte à des ensembles de variables d'un RCQ :

Définition 5 *Soient $\mathcal{N} = (V, C)$ un RCQ et $X = \{X_0, \ldots, X_n\}$ une famille de sous-ensembles de V. $\mathcal{N}$ est ${}^{\diamond}_{X}$-cohérent si, et seulement si, pour tout $X_i \in X$, le RCQ $\mathcal{N}_{X_i}$ est un RCQ $\diamond$-cohérent.*

Notons que la ${}^{\diamond}_{X}$-cohérence est proche de la chemin-cohérence partielle proposée dans [3]. Étant donnés un RCQ $\mathcal{N} = (V, C)$ et $X = \{X_0, \ldots, X_n\}$ une famille de sous-ensembles de V, nous dénoterons par ${}^{\diamond}_{X}(\mathcal{N})$ le plus grand (pour $\subseteq$) ${}^{\diamond}_{X}$-cohérent sous-RCQ de $\mathcal{N}$.
Le résultat suivant concerne des RCQ dont les contraintes sont définies par des relations issues d'une classe $\mathcal{C}$ pour laquelle tout RCQ fermé par faible composition est globalement cohérent. À titre d'illustration, nous pouvons citer la classe des relations convexes de l'Algèbre des Intervalles pour laquelle les RCQ possèdent une telle propriété.

Théorème 1 *Soit $\mathcal{N} = (V, C)$ un RCQ défini sur une classe de relations $\mathcal{C}$ pour laquelle tout RCQ $\diamond$-cohérent est globalement cohérent et soit $T = (X = \{X_0, \ldots, X_n\}, F)$ une décomposition arborescente de $\mathcal{N}$. Si $\mathcal{N}$ est un RCQ ${}^{\diamond}_{X}$-cohérent et non trivialement incohérent alors $\mathcal{N}$ est cohérent.*

Nous allons caractériser un résultat similaire pour la classe des relations ORD-Horn de l'Algèbre des Intervalles. Étant donné un RCQ $\mathcal{N}$ fermé par faible composition défini par des relations de cette classe, dans le cas général $\mathcal{N}$ n'est pas forcément globalement cohérent. Malgré tout, nous avons une propriété proche de la globale cohérence qui est satisfaite : toute solution partielle de maximale de $\mathcal{N}$ peut être étendue en une solution maximale de $\mathcal{N}$ (voir Proposition 6 dans [8]). Dans [8], Ligozat attribue une dimension (un entier compris entre 0 et 2) à chacune des relations de base de l'Algèbre des Intervalles. Une solution partielle de dimension maximale est une solution satisfaisant pour chaque couple de variables une relation de base de dimension maximale par rapport aux dimensions des autres relations de base présentes dans la contrainte. Ainsi, nous pouvons établir la propriété suivante :

Théorème 2 *Soit $\mathcal{N} = (V, C)$ un RCQ défini sur la classe des relations ORD-Horn de l'Algèbre des Intervalles et soit $T = (X = \{X_0, \ldots, X_n\}, F)$ une décomposition*

arborescente de $\mathcal{N}$. Si $\mathcal{N}$ est un RCQ $\overset{\diamond}{\underset{X}{}}$*-cohérent et non trivialement incohérent alors $\mathcal{N}$ est cohérent.*

Preuve La preuve est similaire à la preuve de la proposition 5 si ce n'est que les instanciations partielles manipulées sont des instanciations partielles maximales. $\dashv$

Nous avons démontré dans la section précédente qu'étant donné un RCQ $\mathcal{N}$ et une décomposition PartRec $T = (X = \{X_0, \ldots, X_n\}, F)$ d'un RCQ nous pouvions définir une décomposition arborescente $T' = \overline{T} = (X' = \{X'_0, \ldots, X'_n\}, F)$ telle que pour tout $X'_i \in X'$, il existe $X_j \in \mathsf{leaves}(T)$ avec $X'_i \subseteq X_j$. À partir de cela et des deux théorèmes précédents découlent les propriétés suivantes.

Corollaire 1 *Soient $\mathcal{N} = (V, C)$ un* RCQ *défini sur une classe de relations $\mathcal{C}$ et $T = (X = \{X_0, \ldots, X_n\}, F)$ une décomposition* PartRec *de $\mathcal{N}$.*

- *Si $\mathcal{C}$ est telle que tout* RCQ *$\diamond$-cohérent défini sur $\mathcal{C}$ est globalement cohérent et si $\mathcal{N}$ est $\overset{\diamond}{\underset{\mathsf{leaves}(T)}{}}$-cohérent alors $\mathcal{N}$ est cohérent.*
- *Si $\mathcal{C}$ est la classe des relations ORD-Horn de l'Algèbre des Intervalles et si $\mathcal{N}$ est $\overset{\diamond}{\underset{\mathsf{leaves}(T)}{}}$-cohérent alors $\mathcal{N}$ est cohérent.*

5 Algorithme

Dans cette section nous proposons un algorithme permettant de résoudre le problème de la cohérence d'un RCQ $\mathcal{N} = (V, C)$ à partir d'un ensemble X de clusters issus d'une décomposition arborescente $T = (X, F)$ de $\mathcal{N}$ et d'une classe de relations $\mathcal{C}$. La classe $\mathcal{C}$ est supposée être la classe des relations ORD-Horn de l'Algèbre des Intervalles ou bien une classe pour laquelle tout RCQ $\diamond$-cohérent est globalement cohérent. À l'aide d'une recherche exhaustive de type backtrack, l'algorithme proposé tente de caractériser un sous-RCQ de $\mathcal{N}$ $\overset{\diamond}{\underset{X}{}}$-cohérent et défini sur $\mathcal{C}$. L'intérêt de l'algorithme proposé réside d'une part dans le fait que seules les contraintes définies par des variables d'un même cluster seront considérées lors de la recherche. Et d'autre part, lors de la recherche, la fermeture par faible composition est réalisée uniquement au travers d'opérations de triangulation impliquant des triplets de variables issues d'un même cluster.

Présentons en détails l'algorithme proposé. La fonction principale Cohérence débute son traitement en dupliquant le RCQ $\mathcal{N}$ et le poursuit en réalisant un pré-traitement sur $\mathcal{N}$ à l'aide d'un appel de la fonction preTreatment. Cette fonction non définie explicitement doit retourner un sous-RCQ équivalent à $\mathcal{N}$. En règle générale, la méthode par faible composition est utilisée comme pré-traitement du fait de sa rapidité. Malgré tout, des méthodes avec des forces de filtrage plus élevées telles que celles proposées dans [4] peuvent être utilisées. Si le pré-traitement conduit à un RCQ trivialement incohérent l'algorithme se termine (lignes 4-5). À la ligne 6, est initialisé l'ensemble E par l'ensemble des couples de variables appartenant à au moins un des clusters de X. L'ensemble E délimite d'une part les contraintes sur lesquelles est réalisée la recherche (fonction CohérenceAux, ligne 5). D'autre part, E permet de délimiter les contraintes sur lesquelles doivent être réalisées les opérations de triangulation pour maintenir le

RCQ $\diamond_X$-cohérent, traitement réalisé par la fonction $\diamond_X$–Cohérence. La recherche effective est réalisée par la fonction CohérenceAux. En premier lieu, cette fonction maintient $\mathcal{N}$ $\diamond_X$–Cohérent (ligne 2). Dans le cas où le RCQ est détecté non cohérent, un retour arrière est réalisé (lignes 3-4). Dans le cas contraire, la prochaine contrainte à traiter est sélectionnée (ligne 5). La fermeture par rapport à $\mathcal{C}$ de la relation définissant la contrainte sélectionnée ne doit pas être incluse dans la relation définissant initialement cette même contrainte. Nous verrons par la suite que cette condition est suffisante pour assurer l'arrêt de la recherche (ligne 6-7). La relation $\mathcal{N}[v, v']$ correspondant à la contrainte sélectionnée est scindée en sous-relations $r_1, \ldots, r_k$ appartenant à $\mathcal{C}$ (ligne 8). Ainsi, nous avons $\mathcal{N}[v, v'] = r_1 \cup \ldots \cup r_k$. Puis, en remplaçant alternativement la contrainte sélectionnée par chacune de ces sous-relations r_i, un sous-RCQ de $\mathcal{N}$ cohérent est recherché à l'aide de l'appel de CohérenceAux à la ligne 11. Ce sous-RCQ aura pour particularité d'avoir comme relation définissant la contrainte entre les variables v et v' une relation $r \subseteq r_i$. Notons que $r_i \in \mathcal{C}$ et est forcément une sous-relation de $\mathcal{N}_{Init}[v, v']$, ainsi nous pouvons affirmer que la contrainte entre v et v' ne sera plus sélectionnée (ligne 5) lors des appels suivants de CohérenceAux. Ceci garanti l'arrêt de notre algorithme. Dans le cas où chacune des sous-relations $r_1, \ldots, r_k$ ne conduit pas à la caractérisation d'un sous-RCQ cohérent, l'incohérence de $\mathcal{N}$ est détectée.

Function `Cohérence(`$\mathcal{N}$,X,$\mathcal{C}$`)` : Booléen

in : $\mathcal{N} = (V, C)$, un RCQ, X un ensemble d'ensembles de variables de V, $\mathcal{C}$ une classe de relations.

output : $true$ si $\mathcal{N}$ est cohérent, $false$ sinon.

```
1 begin
2     N_Init ← N;
3     N ← preTreatment(N);
4     if N = ⊥ then
5         return false ;
6     E ← {(v, v') : ∃X_i ∈ X, avec v, v' ∈ X_i};
7     return CohérenceAux(N) ;
8 end
```

Théorème 3 *Soit $\mathcal{N} = (V, C)$ un* RCQ *défini sur une classe de relations $\mathcal{C}$. $\mathcal{C}$ est la classe des relations $OrdHorn$ de l'algèbre des intervalles ou bien est telle que tout* RCQ *$\diamond$-cohérent défini sur $\mathcal{C}$ est globalement cohérent. Soit $T = (X, F)$ une décomposition arborescente de $\mathcal{N}$. L'algorithme* Cohérence$(\mathcal{N}, X, \mathcal{C})$ *résout le problème de la cohérence de $\mathcal{N}$.*

Preuve. À partir de la description précédente de l'algorithme nous pouvons affirmer que l'algorithme s'arrête et que de plus, lorsqu'une incohérence est détectée, $\mathcal{N}$ est bien incohérent. Concentrons nous sur le cas où l'algorithme Cohérence détecte la cohérence de $\mathcal{N}$ et montrons que dans ce cas là, $\mathcal{N}$ est bien un RCQ cohérent. Dans la suite, $\mathcal{N}_D$ dénotera le RCQ $\mathcal{N}$ initial et nous dénoterons par $\mathcal{N}_F$ le RCQ correspondant au RCQ $\mathcal{N}$ lors de l'arrêt de Cohérence. Nous pouvons affirmer que $\mathcal{N}_F$ est un sous-RCQ de

Function `CohérenceAux(`$\mathcal{N}$`)` : Booléen

in : $\mathcal{N} = (V, C)$, un RCQ.
output : $true$ si $\mathcal{N}$ est cohérent, $false$ sinon.

1 **begin**
2 $\mathcal{N} \leftarrow \overset{\diamond}{_X}$–Cohérence($\mathcal{N}, E$);
3 **if** $\mathcal{N} = \perp$ **then**
4 **return** *false* ;
5 Sélectionne $(v, v') \in E$ tel que $(\mathcal{N}[v, v'])^{\mathcal{C}} \not\subseteq \mathcal{N}_{Init}[v, v']$;
6 **if** *un tel couple n'existe pas* **then**
7 **return** *true* ;
8 Partager $\mathcal{N}[v, v']$ en sous-relations $r_1, \ldots, r_k \in \mathcal{C}$;
9 **foreach** $i \in 1, \ldots, k$ **do**
10 $\mathcal{N}[v, v'] \leftarrow r_i$; $\mathcal{N}[v', v] \leftarrow r_i^{-1}$;
11 **if** CohérenceAux($\mathcal{N}$) **then**
12 **return** *true* ;
13 **return** *false*
14 **end**

Function $\overset{\diamond}{_X}$`-Cohérence(`$\mathcal{N}$`)`

in : $\mathcal{N} = (V, C)$.
output : $\mathcal{N}$ est $\diamond$-cohérent par rapport à E.

1 **begin**
2 changement $\leftarrow$ true ;
3 **while** *changement* **do**
4 changement $\leftarrow$ false ;
5 **foreach** *triplet* $(v_i, v_j), (v_i, v_k), (v_k, v_j) \in E$ **do**
6 $rel_{ij} \leftarrow \mathcal{N}[v_i, v_j] \cap (\mathcal{N}[v_i, v_k] \diamond \mathcal{N}[v_k, v_j])$;
7 **if** $\mathcal{N}[v_i, v_j] \not\subseteq rel_{ij}$ **then**
8 changement $\leftarrow$ true ;
9 $\mathcal{N}[v_i, v_j] \leftarrow rel_{ij}$;
10 **return** $\mathcal{N}$
11 **end**

$\mathcal{N}_I$ puisque toutes les opérations de modification du RCQ consistent en des suppressions de relations de base présentes dans ses contraintes. Soit $\mathcal{N}' = (V, C')$ défini par $\mathcal{N}'[v, v'] = (\mathcal{N}_F[v, v'])^{\mathcal{C}}$, pour tout v, v'. Du fait de la ligne 5 de CohérenceAux, nous pouvons affirmer que $\mathcal{N}' \subseteq \mathcal{N}_D$. Considérons maintenant le RCQ $\overset{\diamond}{_X}(\mathcal{N}')$, du fait que $\mathcal{N}'$ est défini sur la classe $\mathcal{C}$ nous pouvons affirmer que $\overset{\diamond}{_X}(\mathcal{N}')$ est également un RCQ défini sur $\mathcal{C}$ (ceci provient du fait que $\mathcal{C}$ est un ensemble stable pour les opérations d'intersection, d'inverse et de composition faible). Par définition de $\mathcal{N}'$ nous avons $\mathcal{N}_F \subseteq \mathcal{N}'$, il suit que $\overset{\diamond}{_X}(\mathcal{N}_F) \subseteq \overset{\diamond}{_X}(\mathcal{N}')$. Du fait de la ligne 2 de CohérenceAux, nous pouvons également affirmer que $\mathcal{N}_F = \overset{\diamond}{_X}(\mathcal{N}_F)$. Ainsi, $\mathcal{N}_F \subseteq \overset{\diamond}{_X}(\mathcal{N}') \subseteq \mathcal{N}_D$. De plus, $\overset{\diamond}{_X}(\mathcal{N}')$ n'est pas trivialement incohérent car $\mathcal{N}_F$ n'est pas trivialement incohé-

rent. $\overset{\diamond}{_X}(\mathcal{N}')$ est donc un RCQ non trivialement incohérent, défini sur $\mathcal{C}$ et $\overset{\diamond}{_X}$-cohérent. D'après les théorèmes 1 et 2 nous pouvons en déduire que $\overset{\diamond}{_X}(\mathcal{N}')$ est cohérent. $\overset{\diamond}{_X}(\mathcal{N}')$ étant un sous-RCQ de $\mathcal{N}_D$ nous pouvons en conclure que $\mathcal{N}_D$ est cohérent. $\dashv$

L'application de l'algorithme Cohérence nécessite au préalable le calcul d'une décomposition arborescente $T = (X, F)$ de $\mathcal{N}$. Notons qu'il n'est pas nécessaire de calculer une décomposition arborescente optimale (de largeur minimale) qui est un problème NP-difficile. Lors d'une étude expérimentale préliminaire, nous avons considéré des RCQ générés aléatoirement de manière similaire à celle utilisée dans [4]. À partir de décompositions de type PartRec calculées selon la méthode proposée dans [7], nous avons pu constater que le temps de calcul de ces décompositions est négligeable compte tenu du temps effectif mis lors de la recherche. D'autre part, l'utilisation de ces décompositions permet d'éliminer en moyenne un peu moins de 40% des triplets à considérer lors de l'application de la fermeture par faible composition et un peu plus de 20% des contraintes lors de la recherche pour les instances difficiles se trouvant à une densité de relations non triviales de 10.25. Ceci se traduisant par un gain de temps lors de la résolution de ces instances.

6 Conclusion et perspectives

Dans cet article, nous avons défini et étudié les décompositions de RCQ de type PartRec. Ces décompositions correspondent à celles utilisées par *Li et al*. [7] dans un processus de décomposition de RCQ proposé dans le cadre d'une traduction de RCQ en problèmes SAT. Nous avons montré que les décompositions de type PartRec peuvent s'exprimer de manière équivalente par des décompositions arborescentes. D'autre part, nous avons étudié le problème de la cohérence d'un RCQ en lien avec une décomposition arborescente, une classe traitable et la fermeture par faible composition. Nous avons montré que pour certaines classes de relations, en particulier pour la classe des relations ORD-Horn de l'algèbre des intervalles, l'obtention de la $\diamond$–cohérence sur les contraintes appartenant aux différents clusters de la décomposition arborescente est suffisante pour affirmer la cohérence du réseau de contraintes qualitatives. Ce résultat généralise des propriétés établis dans [7] puisque nous considérons d'une part des décompositions arborescentes et d'autre part des classes traitables incluant l'ensemble des relations singletons. Nous avons également proposé un algorithme permettant de décider du problème de la cohérence des RCQ en tirant profit d'une classe traitable et d'une décomposition arborescente. Une évaluation expérimentale de cet algorithme est en cours de réalisation. Elle consiste en l'application de cet algorithme sur des RCQ de l'Algèbre des Intervalles en utilisant la classe des relations ORD-Horn et des décompositions PartRec calculées d'une manière similaire à celle utilisée par *Li et al*. [7]. Il est également envisagé d'utiliser des décompositions arborescentes autres que les décompositions de type PartRec. Une autre perspective de recherches est la définition et l'étude d'algorithmes inspirés de l'algorithme BTD [6] afin de profiter des structures des décompositions arborescentes utilisées lors de la résolution des RCQ.

Références

[1] J-F. Allen. An interval-based representation of temporal knowledge. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI'81)*, pages 221–226, 1981.

[2] Peter Van Beek and Robin Cohen. Exact and approximate reasoning about temporal relations. *Computational Intelligence*, 6 :133–44, 1990.

[3] Christian Bliek and Djamila Sam-Haroud. Path consistency on triangulated constraint graphs. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 456–461, 1999.

[4] Jean-François Condotta and Christophe Lecoutre. A class of df-consistencies for qualitative constraint networks. In *Proceedings of 12th International Conference on Principles of Knowledge Representation and Reasoning (KR'10)*, pages 319–328, 2010.

[5] Z. Gantner, M. Westphal, and S. Wolfl. GQR a fast reasoner for binary qualitative constraint calculi. In *Proceedings of AAAI'08 Workshop on Spatial and Temporal Reasoning*, 2008.

[6] Philippe Jégou and Cyril Terrioux. Hybrid backtracking bounded by tree-decomposition of constraint networks. *Artificial Intelligence*, 146(1) :43–75, 2003.

[7] Jason Jingshi Li, Jinbo Huang, and Jochen Renz. A divide-and-conquer approach for solving interval algebra networks. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 572–577, 2009.

[8] Gérard Ligozat. A New Proof of Tractability for ORD-Horn Relations. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, volume 1, pages 395–401, 1996.

[9] Gérard F. Ligozat. On Generalized Interval Calculi. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI'91)*, pages 234–240, Menlo Park/Cambridge, July 1991. American Association for Artificial Intelligence, AAAI Press/The MIT Press.

[10] Robert A. Morris, William D. Shoaff, and Lina Al-Khatib. Path consistency in a network of non-convex intervals. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI'93)*, 1993.

[11] B. Nebel. Solving hard qualitative temporal reasoning problems : Evaluating the efficienty of using the ORD-Horn class. In *Proceedings of ECAI'96*, pages 38–42, 1996.

[12] Bernhard Nebel and 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.

[13] Duc Nghia Pham, John Thornton, and Abdul Sattar. Modelling and solving temporal reasoning as propositional satisfiability. *Artif. Intell.*, 172(15) :1752–1782, 2008.

[14] A.K. Pujari, G.V. Kumari, and A. Sattar. Indu : An interval and duration network. In *Proceedings of Australian Joint Conference on Artificial Intelligence*, pages 291–303, 1999.

[15] D.A. Randell, Z. Cui, and A.G. Cohn. A spatial logic based on regions and connection. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, pages 165–176, 1992.

[16] J. Renz. Maximal tractable fragments of the region connection calculus : a complete analysis. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 448–455, 1999.

[17] Jochen Renz and Gérard Ligozat. Weak composition for qualitative spatial and temporal reasoning. In *Proceedings of the Eleventh International Conference on Principles and Practice of Constraint Programming (CP'05)*, 2005.

[18] Neil Robertson and Paul D. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3) :309–322, 1986.

Annexe

Preuve (Proposition 2).

(1) Considérons le cas où $X_i \notin \mathsf{leaves}(T)$ (pour le cas où $X_i \in \mathsf{leaves}(T)$ la propriété est trivialement vraie puisque $\overline{X_i} = X_i$ et $X_i \neq \emptyset$). Soient $X_i, X_j, X_k \in X$ tels que X_i est le père de X_j et X_k. Supposons par l'absurde que $X_j \cap X_k = \emptyset$. D'après la définition 3, nous avons $\mathcal{N}_{X_i} = \mathcal{N}_{X_j} \cup \mathcal{N}_{X_k}$. Ainsi, pour tout $v \in X_j$ et $v' \in X_k$, nous pouvons affirmer que $\mathcal{N}[v, v'] = \Psi$ et que (v, v') n'est pas une arête de $\mathsf{G}(\mathcal{N})$. De plus, nous avons précédemment effectuer l'hypothèse que pour tout RCQ $\mathcal{N}$ considéré $\mathsf{G}(\mathcal{N})$ est connecté. Ainsi, pour tout $v \in X_j$ et $v' \in X_k$, il existe un chemin dans $\mathsf{G}(\mathcal{N})$ entre v et v' et ce chemin passe forcément par deux arêtes (v, v'') et (v, v''') avec $v'' \in V \setminus \{X_j \cup X_k\}$ et $v''' \in V \setminus \{X_j \cup X_k\}$. Puisque $X_j \cup X_k = X_i$ nous avons $v'' \notin X_i$. Soit X_l le plus proche ascendant de X_i tel que $v'' \in X_l$. Soit X_m et X_o les deux fils de X_l avec $X_m \in \mathsf{asc}(X_i)$. Nous avons $v'' \notin X_m$ et $v \in X_m$. De plus, $v'' \in X_o$ puisque $X_o \cup X_m = X_l$. De plus, nous savons que $\mathcal{N}_{X_l} = \mathcal{N}_{X_m} \cup \mathcal{N}_{X_o}$. Il s'ensuit que $\mathcal{N}[v, v''] = \Psi$. Or, (v, v'') est une arête de $\mathsf{G}(\mathcal{N})$. Il y a une contradiction. Nous pouvons conclure $X_j \cap X_k \neq \emptyset$.

(2) Soient $X_i, X_j, X_k \in X$ avec $X_k = \mathsf{lca}(\{X_i, X_j\})$. Supposons que $X_k \neq X_i$ et $X_k \neq X_j$. Nous avons forcément $X_k \notin \mathsf{leaves}(T)$, dénotons par X_l et X_m les deux fils de X_k. Nous avons $X_i \in \mathsf{desc}(X_l)$ et $X_j \in \mathsf{desc}(X_m)$ ou bien, $X_i \in \mathsf{desc}(X_m)$ et $X_j \in \mathsf{desc}(X_l)$. Ainsi d'après la proposition 1 (1), nous avons $X_i \subseteq X_l$ et $X_j \subseteq X_m$ ou $X_i \subseteq X_m$ et $X_j \subseteq X_l$. Il suit que $X_i \cap X_j \subseteq X_m \cap X_l$. Nous pouvons en conclure que $X_i \cap X_j \subseteq \overline{X_k}$.

(3) Soit $X_i \in X$. Considérons tout d'abord le cas où $X_i \in \mathsf{leaves}(X)$. Par définition, $\overline{X_i} = X_i$, ainsi la propriété est satisfaite. Supposons maintenant que $X_i \notin \mathsf{leaves}(X)$ et soient X_k et X_l les deux fils de X_i. Nous avons $X_k \cap X_l = \overline{X_i} \neq \emptyset$. Soit $v \in \overline{X_i}$, nous avons $v \in X_i$. D'après la proposition 1 (4), il existe $X_j \in \mathsf{leaves}(T) \cap \mathsf{desc}(X_i)$, tel que $v \in X_j$. D'après la propriété (2) de la définition 3, nous avons $X_k \cap X_l \subseteq X_j$ puisque $(X_k \cap X_l) \cap X_j \neq \emptyset$.

$\dashv$

Preuve (Proposition 3).

– Nous savons que $\overline{X_i} \neq \emptyset$ (proposition 2 (1)), il existe donc une variable $v \in \overline{X_i}$. Par définition de $\overline{X_i}$, nous savons que $v \in X_j$. D'après la proposition 1 (4), il existe $X_k \in \mathsf{leaves}(T) \cap \mathsf{desc}(X_j)$ tel que $v \in X_k$. D'après la propriété (3) de la définition 3, nous pouvons affirmer que $\overline{X_i} \subseteq X_k$. Ainsi, $X_k \in \widetilde{X_i X_j}$ et nous pouvons en conclure que $\widetilde{X_i X_j} \neq \emptyset$.

– Soit $X_l = \mathsf{lca}(\widetilde{X_i X_j})$. Il existe $X_k, X_m \in \widetilde{X_i X_j}$ tels que $X_l = \mathsf{lca}(\{X_k, X_m\})$. Dans le cas où $X_l = X_k$ ou $X_l = X_m$ la propriété est trivialement vraie. Dans le

cas où, $X_l \neq X_k$ et $X_l \neq X_m$ nous avons d'après la proposition 2 (2), $X_k \cap X_m \subseteq \overline{X_l}$. Comme $\overline{X_i} \subseteq \overline{X_k}$ et $\overline{X_i} \subseteq \overline{X_m}$ nous pouvons en déduire que $\overline{X_i} \subseteq \overline{X_l}$. De plus, comme $X_l = \mathsf{lca}(\{X_k, X_m\})$, $X_k \in \mathsf{desc}(X_j)$ et $X_m \in \mathsf{desc}(X_j)$, nous avons $X_l \in \mathsf{desc}(X_j)$. Nous pouvons en conclure que $X_l \in \widetilde{X_i X_j}$.

$\dashv$

Preuve (Proposition 4). Les propriétés (1) et (2) de la définition 2 découlent du fait que pour tout $X_i \in \mathsf{leaves}(X)$, $\overline{X_i} = X_i$ et de la proposition 1. Montrons maintenant que la propriété (3) de la définition 2 est satisfaite par $\overline{T}$. Soient $\overline{X_i}, \overline{X_j}, \overline{X_k} \in \overline{X}$ avec $\overline{X_j}$ sur l'unique chemin entre $\overline{X_i}$ et $\overline{X_k}$ dans $\overline{T}$. Montrons que si $v \in \overline{X_i}$ et $v \in \overline{X_k}$ alors $v \in \overline{X_j}$. Si la longueur du chemin entre $\overline{X_i}$ et $\overline{X_k}$ est 0 la propriété est trivialement vraie. Supposons que la propriété soit vraie pour tout chemin de longueur $l \geq 0$ et montrons par induction qu'elle est vraie pour tout chemin entre $\overline{X_i}$ et $\overline{X_k}$ de longueur $l + 1$. Supposons dans un premier temps que $X_i \notin \mathsf{desc}(X_k)$ et $X_k \notin \mathsf{desc}(X_i)$ dans T. En examinant la définition 4 nous remarquons que dans $\overline{T}$ un chemin entre $\overline{X_i}$ et $\overline{X_k}$ passe forcément par $\overline{X_l}$ avec $X_l = \mathsf{lca}(\{X_i, X_k\})$. Nous avons d'après la proposition 2 (2), $X_i \cap X_k \subseteq \overline{X_l}$. Par conséquent, $v \in \overline{X_l}$. En considérant les deux chemins $\overline{X_i}, \ldots, \overline{X_l}$ et $\overline{X_l}, \ldots, \overline{X_j}$ qui ont une longueur inférieure ou égale à l, et en utilisant l'hypothèse d'induction nous pouvons affirmer que tous les noeuds du chemin $\overline{X_i}, \ldots, \overline{X_l}$ et ceux du chemin $\overline{X_l}, \ldots, \overline{X_k}$ contiennent v puisque $\overline{X_i}, \overline{X_l}$ et $\overline{X_k}$ contiennent v. Supposons maintenant que $X_i \in \mathsf{desc}(X_k)$ dans T (le cas $X_k \in \mathsf{desc}(X_i)$ peut être traité de manière similaire). Le cas $X_i = X_k$ est trivial, supposons que $X_i \neq X_k$. Nous avons $X_i \in \mathsf{desc}(X_l)$ avec X_l un des deux fils de X_k dans T. En examinant la définition 4 nous remarquons que dans $\overline{T}$, un chemin entre $\overline{X_k}$ et $\overline{X_i}$ passe forcément par $\overline{X_m}$ avec $X_m = \mathsf{lca}(\widetilde{X_k X_l})$. D'après la proposition 3 (2), nous savons que $\overline{X_k} \subseteq \overline{X_m}$. Il en résulte que $v \in \overline{X_m}$. En considérant les deux chemins $\overline{X_i}, \ldots, \overline{X_m}$ et $\overline{X_m}, \ldots, \overline{X_k}$ qui ont une longueur inférieure ou égale à l, et en utilisant l'hypothèse d'induction nous pouvons affirmer que tous les noeuds du chemin $\overline{X_i}, \ldots, \overline{X_m}$ et ceux du chemin $\overline{X_m}, \ldots, \overline{X_k}$ contiennent v puisque $\overline{X_i}, \overline{X_m}$ et $\overline{X_l}$ contiennent v.
Le fait que pour tout $\overline{X_i} \in \overline{X}$, il existe $X_j \in \mathsf{leaves}(T)$ tel que $\overline{X_i} \subseteq X_j$ découle, de manière directe de la proposition 2 (3). $\dashv$

Preuve (Proposition 1). Nous supposons sans perte de généralité que T possède une racine. Soient un noeud $X_i \in X$ et $T_{X_i} = (X_{X_i}, F_{X_i})$ le sous-arbre de T de racine X_i. Étant donnée une instanciation partielle s sur V' avec $V' \cap X'_i \subseteq X_i$ pour tout $X'_i \in X_{X_i}$ et telle que pour tout $X_j \in X$ avec $X_j \subseteq V'$, s_{X_j} est solution de $\mathcal{N}_{X_j}$, nous allons montrer la propriété suivante. s peut-être étendue en une instanciation partielle s' sur $V'' = V' \cup \bigcup\{X'_i \in X_{X_i}\}$ telle que que pour tout $X_j \in X$ avec $X_j \subseteq V''$, s'_{X_j} est solution de $\mathcal{N}_{X_j}$. Nous allons effectuer une preuve par induction sur la taille de X_{X_i}.

- Cas de base : $|X_{X_i}| = 1$. Nous avons $X_{X_i} = \{X_i\}$. Puisque $\mathcal{N}$ est $\diamond_X$-cohérent, nous avons le RCQ $\mathcal{N}_{X_i}$ qui est un RCQ $\diamond$-cohérent et donc globalement cohérent. $s_{V' \cap X_i}$ est une solution partielle de $\mathcal{N}_{X_i}$ pouvant être étendue en une solution s'' de $\mathcal{N}_{X_i}$. Nous définissons par s' l'instanciation partielle sur $V' \cup X_i$ de la manière suivante : si $v \in V'$ alors $s'(v) = s(v)$ sinon $s'(v) = s''(v)$. Nous avons s'_{X_i} qui est une solution de $\mathcal{N}_{X_i}$ et plus généralement s'_{X_k} est solution de $\mathcal{N}_{X_k}$ pour tout $X_k \in X$ et $X_k \subseteq V' \cup X_i$.
- Cas d'induction : $|X_{X_i}| > 1$. Nous supposons la propriété vraie pour tout sous-

arbre $T_{X_j} = (X_{X_j}, F_{X_j})$ avec $|X_{X_j}| < |X_{X_i}|$. Comme dans le cas précédent nous étendons s en une instanciation partielle s' sur $V' \cup X_i$ telle que s'_{X_k} est solution de $\mathcal{N}_{X_k}$ pour tout $X_k \in X$ et $X_k \subseteq V' \cup X_i$. Par hypothèse d'induction cette instanciation partielle s' peut-être étendue à toutes les variables appartenant aux descendants de X_i. En effet, considérons X_l un fils de X_i. Nous remarquons tout d'abord qu'en dénotant par $T_{X_l} = (X_{X_l}, F_{X_l})$ le sous-arbre de T de racine X_l, $|X_{X_l}| < |X_{X_i}|$. De plus comme T est une décomposition arborescente, nous avons pour tout $X_m \in X_{X_l}$, $X_m \cap (V' \cup X_i) \subseteq X_l$ (d'après la propriété (3) de la définition 2). Ainsi, l'hypothèse d'induction peut s'appliquer sur $T_{X_l} = (X_{X_l}, F_{X_l})$.

En appliquant la propriété précédente sur X_r racine de T, nous savons qu'il existe une instanciation partielle s sur l'ensemble des variables $\bigcup\{X_i : X_i \in X\}$ telle que s_{X_i} est solution du RCQ $\mathcal{N}_{X_i}$ pour tout $X_i \in X$. Notons tout d'abord que $V = \bigcup\{X_i : X_i \in X\}$ d'après la propriété (1) de la définition 2. Ainsi, s est une instanciation sur V. De plus nous pouvons montrer que s est solution de $\mathcal{N}$. En effet, soient $v, v' \in V$, si $\mathcal{N}[v, v'] = \Psi$ nous avons $s(v)$ et $s(v')$ qui satisfont clairement la contrainte $\mathcal{N}[v, v']$. Maintenant supposons que $\mathcal{N}[v, v'] \neq \Psi$. D'après la propriété (2) de la définition 2, il existe $X_i \in X$ tel que $v \in X_i$ et $v' \in X_i$. Nous savons que $s(v)$ et $s(v')$ satisfont $\mathcal{N}_{X_i}[v, v']$. Comme, $\mathcal{N}_{X_i}[v, v'] = \mathcal{N}[v, v']$ nous en déduisons que $s(v)$ et $s(v')$ satisfont $\mathcal{N}[v, v']$. Nous pouvons conclure que s est une solution de $\mathcal{N}$ $\dashv$