Décompositions arborescentes intelligentes pour le raisonnement par contraintes qualitatives

Yoan Thomas 1* Michael Sioutis¹ Nofar Carmeli²

¹ LIRMM UMR 5506, University of Montpellier & CNRS, France ² Inria, LIRMM UMR 5506, University of Montpellier & CNRS, France {yoan.thomas,michael.sioutis,nofar.carmeli}@lirmm.fr

Résumé

De nombreuses approches ont été développées pour raisonner avec des réseaux de contraintes qualitatives (RCQ). Dans de nombreuses applications, une phase de triangulation du graphe de contraintes est nécessaire pour assurer la cohérence des méthodes. Si certaines approches visent à optimiser la structure du graphe, comme sa largeur arborescente, elles ne prennent pas en compte la sémantique des relations durant cette phase de triangulation. Pour tester cette idée, nous allons définir différentes mesures de la qualité de la triangulation d'un RCQ. Nous allons pouvoir choisir des triangulations qui minimisent ces mesures, et donner ces bonnes triangulations comme problèmes d'entrées à différents solveurs. Nous allons collecter les différents temps d'exécution pour les comparer, en les mettant notamment en perspective par rapport à la triangulation par défaut. Nous espérons pouvoir conclure sur l'intérêt de consacrer du temps à calculer une bonne triangulation et de concevoir des solveurs qui utilisent ces décompositions.

Abstract

Many approaches have been developed for reasoning with Qualitative Constraint Networks (QCN). In many applications, a graph triangulation phase is necessary to ensure method coherence. While some approaches aim to optimize the graph's structure, such as its tree width, they do not take into account the semantics of relations during this triangulation phase. To test this new idea, we will define various measures of the quality of a QCN's triangulation. We will be able to choose triangulations that minimize these measures and provide these good triangulations as input problems to different solvers. We will collect different execution times to compare them, particularly in relation to the default triangulation. We hope to be able to conclude on the benefits of dedicating time to calculating a good triangulation and designing solvers that use these decompositions.

Introduction

Le Raisonnement Qualitatif Spatial et Tempo-40 rel (RQST) est un domaine majeur de recherche en IA, et en particulier en Raisonnement et Représentation des Connaissances, qui utilise les concepts intuitifs humains du temps et de l'espace de manière abstraite, via un langage simple de contraintes qualitatives [9, 16]. Par exemple, une utilisation des expressions qualitatives dans le langage naturel pourrait consister à utiliser les termes "dedans", "avant" et "au nord" pour décrire la relation entre un objet et un autre objet ou avec lui-même, sans fournir directement d'information quantitative sur cet objet. Ainsi, le RQST fournit un cadre concis et peu coûteux pour le raisonnement spatio-temporel, ce qui a conduit à une grande variété de domaines d'application et de recherche, tels que la robotique cognitive [10], l'apprentissage profond [15], l'intelligence 55 ambiante, l'explication visuelle [28] et la création de sens [29], les questions-réponses sémantiques [28], la modélisation qualitative [23] et le minage de données spatio-temporelles [19, 13, 14].

L'Algèbre d'Intervalle (AI) [1] est l'une des approches 60 en Intelligence Artificielle pour représenter et raisonner avec des relations temporelles, des relations qui peuvent être encodées par des contraintes binaires entre deux variables/entités. Ainsi, ces informations qualitatives peuvent être modélisées par une variante à domaine infini du Problème de Satisfaction de Contraintes (CSP), appelée Réseau de Contraintes Qualitatives (RCQ). Un exemple est donné dans la Figure 1a.

Motivations et travaux connexes

Avec un RCQ, nous pouvons résoudre plusieurs pro-*Mémoire de fin d'étude : Yoan Thomas¹ est auteur principal. 70 blèmes, tels que savoir s'il est satisfaisable ou com-

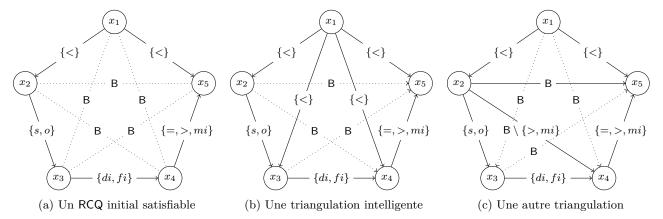


FIGURE 1 – Exemples de RCQ utilisant l'Algèbre d'Intervalle (Al) [1]; les symboles m, o, f, s correspondent aux relations de base meet, overlap, during, finish et start respectivement, avec bi dénotant la converse de b; la relation < entre X et Y se lit par exemple "X se déroule avant Y". Les deux RCQ de droites sont des fermetures chemin-cohérentes du RCQ initial à gauche, en utilisant deux triangulations différentes.

ment réparer efficacement ce RCQ s'il est insatisfaisable. Dans le cas général, le problème de satisfaisabilité est NP-dur [20, 25]. Cependant, il existe une sous-classe maximale polynomiale (tractable) contenant toutes les relations de base de Al pouvant être utilisée pour améliorer considérablement l'efficacité des calculs [20, 25]. 105 Des travaux récents améliorent l'efficacité et la scalabilité en pratique, notamment via l'utilisation de décompositions arborescentes [2, 6, 26, 12, 22]. Ces travaux 80 exploitent la propriété de patchwork des contraintes, qui assure intuitivement que la combinaison de deux 110 RCQ satisfaisables qui s'accordent complètement sur leurs variables en commun reste satisfaisable. De plus, des résultats théoriques sur la complexité paramétrée par la largeur arborescente [12, 3] vont dans ce sens.

À notre connaissance, il n'y a pas d'article publié $^{^{115}}$ qui utilise des décompositions arborescentes en prenant en compte ses implications sur les relations du RCQ pour raisonner avec les RCQ. Dans notre travail, nous 90 allons pouvoir choisir des décompositions arborescentes intelligentes qui prennent en compte la sémantique des relations; voir la Figure 1 (plus de détails suivront dans la suite).

Organisation

Le reste de cet article est organisé de la manière suivante : la section 2 introduit les notions autour des $\mathsf{RCQ}.$ La section 3 décrit notre approche et ce que nous $_{\scriptscriptstyle{130}}$ (RCQ), définis de la manière suivante : espérons obtenir comme résultats, tandis que la section 4 nous permet de conclure et d'envisager différentes 100 perspectives que l'étude pourrait amener.

Préliminaires techniques

Un langage de contraintes qualitatives binaires est basé sur un ensemble fini B de relations exhaustives et deux à deux disjointes, appelé l'ensemble des relations de base (atomes), qui est défini sur un domaine infini D [17]. Ces relations de base fournissent des informations sur la relation entre deux entités, avec une granularité donnée par le domaine D; une information indéfinie peut être spécifiée par l'union des relations de base possibles, et est représentée par l'ensemble les contenant. L'ensemble B contient la relation identité ld, et est clos par l'opération converse ($^{-1}$). L'ensemble des relations 2^B est muni des opérations ensemblistes d'union et d'intersection, de l'opération converse et de l'opération de faible composition, notée par \diamond [17]. La faible composition (\$\dightarrow\$) de deux relations de base $b, b' \in \mathsf{B}$ est la plus petite relation $r \in 2^{\mathsf{B}}$ qui inclut $b \circ b'$; formellement, $b \diamond b' = \{b'' \in \mathsf{B} : b'' \cap (b \circ b') \neq \emptyset\}$, où $b \circ b' = \{(x,y) \in D \times D : \exists z \in D \text{ tel que }$ $(x,z) \in b \text{ et } (z,y) \in b'$ }. Enfin, pour tout $r \in 2^{\mathsf{B}}$, $r^{-1} = \bigcup \{b^{-1} : b \in r\}$, et pour tout $r,r' \in 2^{\mathsf{B}}$, $r \diamond r' = \bigcup \{b \diamond b' : b \in r, b' \in r'\}.$

Pour l'Algèbre d'Intervalle (AI) [1] nous avons B = $\{=,<,>,m,mi,o,oi,d,di,s,si,f,fi\}$; voir la Figure 1, 125 et la Figure 3 en annexe.

Les réseaux de contraintes qualitatives

Une manière de représenter et de raisonner avec des informations qualitatives spatiales ou temporelles est d'utiliser des Réseaux de Contraintes Qualitatives

Définition 1 Un Réseau de Contraintes Qualitatives (RCQ) est un tuple (V, C) où :

 $-V = \{v_1, \ldots, v_n\}$ est un ensemble non vide de variables, chacune représentant un objet dans le domaine infini D.

135

145

150

155

 $que\ C(v,v)\ =\ \{\operatorname{Id}\}\ pour\ tout\ v\ \in\ V\ et$ $C(u,v) = C(v,u)^{-1}$ pour tout $u,v \in V$.

Un exemple de RCQ est montré dans la Figure 1; 140 pour plus de clarté, les opérations converses et les boucles Id (relation = de l'AI) ne sont ni montrées ni 190 suivante : mentionnées sur les figures.

Définition 2 Soit $\mathcal{N} = (V, C)$ un RCQ, alors :

- une solution de N est une fonction d'affectation $\sigma: V \to \mathsf{D}$ telle que $\forall v, v' \in V, \exists b \in C(v, v')$ tel que $(\sigma(v), \sigma(v')) \in b$:
- \mathcal{N} est satisfaisable ssi il admet une solution;
- \mathcal{N} est trivialement incohérent, dénoté par $\emptyset \in \mathcal{N}$, $ssi \exists v, v' \in V \ tel \ que \ C(v, v') = \emptyset;$
- Un sous-RCQ \mathcal{N}' de \mathcal{N} , dénoté par $\mathcal{N}' \subseteq \mathcal{N}$, $C'(v,v') \subseteq C(v,v')$;
- Le graphe de contraintes de \mathcal{N} , dénoté par $G_{\mathcal{N}}$, est un graphe (V, E) tel que $\forall u, v \in V$, $C(u,v) \neq \mathsf{B} \ ssi \ \{u,v\} \in E ;$
- \mathcal{N} est atomique ssi $\forall v, v' \in V$, $C(v, v') = \{b\}$,

Rappelons la notion de cohérence par rapport à la faible composition:

160 **Définition 3** Soit un RCQ $\mathcal{N} = (V, C)$ et un graphe $G=(V,E),~\mathcal{N}~est~^{\diamond}_{G}$ -cohérent (chemin-cohérent) ssi 210 $\forall \{v, v'\}, \{v, v''\}, \{v', v''\} \in E, \ C(v, v') \subseteq C(v, v'') \diamond$ C(v'', v').

Dans la suite, soit la faible composition ⋄, l'unique $_{165}$ ⊆-maximal $_{G}^{\diamond}$ -cohérent sous-RCQ de ${\mathcal N}$ est appelé la $_{215}$ fermeture de \mathcal{N} par la ${}^{\diamond}_{G}$ -cohérence et est noté ${}^{\diamond}_{G}(\mathcal{N})$. Nous supposerons que ${}^{\diamond}_{G}(\mathcal{N})$ existe toujours et qu'il est calculable avec une complexité en $\mathcal{O}(\Delta|\mathsf{B}||E|)$ pour un graphe G = (V, E), où Δ est le degré maximum du 170 graphe, et que la $^{\diamond}_{G}$ -cohérence décide de la satisfaisa- $_{220}$ bilité des RCQ atomiques; c'est le cas pour la plupart des langages qualitatifs connus [9].

Dans le cas où G est complet, on obtient la plus classique \diamond -cohérence [24] qui a une complexité en $\mathcal{O}(|E|^3)$. $_{\mbox{\scriptsize 175}}$ Dans le cas général, G est intéressant pour nous lorsque $_{\mbox{\scriptsize 225}}$ c'est un super graphe du RCQ, de telle sorte que toutes les informations du RCQ sont gardées.

Patchwork et graphes cordaux :

180 qui a été montrée dans le cadre des RCQ atomiques utilisant l'Algèbre d'Intervalle (AI), introduite dans [18].

Définition 4 Un langage de contrainte a la propriété de patchwork si, pour tous les réseaux de contraintes satisfaisables $\mathcal{N} = (V, C)$ et $\mathcal{N}' = (V', C')$ définis -C est une application $V \times V \rightarrow 2^{\mathsf{B}}$ telle 185 avec ce langage, tels que $\forall u, v \in V \cap V'$, nous avons C(u,v) = C'(u,v), alors le réseau de contraintes $\mathcal{N} \cup$ \mathcal{N}' est satisfaisable.

> Huang a généralisé l'utilisation de patchwork pour les RCQ non-atomiques [11], donnant la proposition

> **Proposition 1** L'Al a la propriété de patchwork pour les RCQ chemin-cohérents non trivialement incohérents comprenant les relations de la sous-classe polynomiale (tractable) maximale \mathcal{H}_{Al} [21].

> Intuitivement, la propriété de patchwork assure que la combinaison de deux RCQ satisfaisables qui s'accordent complètement sur leurs variables en commun continue d'être satisfaisable.

D'après les travaux dans [2], nous avons un résultat est un RCQ (V,C') tel que $\forall v,v' \in V$, plus fort encore sur la chemin-cohérence, le patchwork et la sous-classe traitable maximale de relations :

> **Proposition 2** ([2]) Soit un RCQ $\mathcal{N} = (V, C)$ de l'Al non trivialement incohérent, avec des relations définies dans \mathcal{H}_{AI} , et pour G = (V, E) un super graphe cordal $de\ G_{\mathcal{N}},\ si\ \forall \{i,j\},\{i,k\},\{j,k\}\in E,\ C(i,j)\subseteq C(i,k)\diamond$ C(k,j), alors \mathcal{N} est satisfaisable.

> La dernière proposition généralise les résultats précédemment introduits, et fait de la chemin-cohérence l'outil principal pour vérifier la satisfaisabilité d'un RCQ. C'est pour cette raison que trouver un super graphe cordal d'un RCQ est primordial.

> Rappelons la définition d'un graphe cordal. Soit un graphe G = (V, E) un graphe simple, G est cordal (ou triangulé) si tout cycle de longueur au moins 4 à une corde, qui est une arête connectant deux nœuds non adjacents du cycle.

> Par exemple, le graphe de contrainte de la Figure 1 n'est pas cordal, mais il le devient en ajoutant les arêtes $\{x_1, x_3\}$ et $\{x_1, x_4\}$. Aussi, on constate que la chemincohérence sur les arêtes $\{x_1, x_3\}$ et $\{x_1, x_4\}$ nous donnent chacune la relation $\{<\}$ (avec respectivement les compositions $C(x_1, x_2) \diamond C(x_2, x_3)$ et $C(x_1, x_5) \diamond$ $C(x_5, x_4)$, contrairement à l'arête $\{x_2, x_5\}$ qui ne donne aucune information sur la relation entre x_2 et x_5 . Ainsi, la triangulation $\{\{x_1, x_3\}, \{x_1, x_4\}\}$ semble être une meilleure triangulation que $\{\{x_2, x_5\}, \{x_2, x_4\}\}$ car la seconde apporte moins d'information après la fermeture par chemin-cohérence que la première.

De plus, nous avons la proposition suivante :

Donnons une définition de la propriété de patchwork 230 **Proposition 3** ([8]) Le graphe G = (V, E) est cordal si et seulement s'il admet une décomposition arborescente où tous ses sacs sont des cliques.

Une définition formelle d'une décomposition arborescente est fournie en annexe.

Ainsi, nous avons une équivalence entre trouver une "bonne" triangulation et trouver une "bonne" décomposition arborescente, et donc nous pourrons utiliser des algorithmes utilisent l'une de ces notions. De plus, nous savons que le processus de triangulation d'un graphe 240 donné G=(V,E) peut être réalisé en temps linéaire en la taille du graphe [7]. Trouver une triangulation avec un nombre minimum d'arêtes ajoutées est NP-dur [30], cependant, nous pourrons énumérer les triangulations minimales (dans laquelle on ne peut pas enlever d'arête sans briser la cordalité) avec un délai entre deux solutions successives en temps polynomial [5, 4].

3 Contributions prévues

265

270

275

Notre problématique consiste à déterminer si une décomposition intelligente permet d'améliorer le raisonnement par contrainte qualitative. Pour ce faire, nous fournirons à plusieurs solveurs, basés sur des rai-285 prendre la moyenne du temps d'exécution de cinq trisonneurs natifs ou sur des encodages SAT, différentes triangulations d'une même instance et constaterons empiriquement les types de triangulations qui améliorent 255 l'efficacité des algorithmes.

triangulation du graphe de contraintes de \mathcal{N} et nous leur donnerons une valeur basée sur une mesure, une application qui prend entrée le RCQ \mathcal{N} et un graphe cordal G. Ces mesures pourraient impliquer, pour un graphe cordal G = (V, E), par exemple :

- le nombre d'arêtes ajoutées dans le graphe de contrainte de \mathcal{N} (fill-in), viz., $|E| - |E(G_{\mathcal{N}})|$,
- le nombre total de relations de base dans le RCQ après $^{\diamond}_{G}$ -cohérence (base rel), viz., $\sum_{\{u,v\}\in E}|^{\diamond}_{G}(\mathcal{N})(u,v)|$, ou encore
- la largeur arborescente du graphe, viz., $\omega(G) 1$. Il existe également d'autres mesures, moins évidentes, qui tirent profit de la $^{\diamond}_{G}$ -cohérence, telles que :
 - le nombre de relations traitables (par exemple, de $\mathcal{H}_{\mathsf{AI}}$ pour l'AI) dans ${}_{G}^{\diamond}(\mathcal{N})$,
 - la "distance" entre ${}^{\diamond}_{G}(\mathcal{N})$ et un \subseteq -maximal sous-RCQ tractable de ${}_{G}^{\diamond}(\mathcal{N})$, ou encore
 - la somme (ou la variation) du nombre de modèles 305 locaux pour chaque arête [27].

Avec ces mesures, nous pourrons aussi essayer de développer des résultats théoriques, comme de nouveaux résultats de complexité paramétrée par nos mesures.

Nous choisirons les triangulations présentant des mesures faibles et les fournirons aux différents solveurs afin de récupérer leurs temps d'exécution. Avec suf-310 Remerciements Ce travail a été en partie financé fisamment de données, nous pourrons comparer les différentes mesures sur différents solveurs et différents types d'instances. Pour chaque mesure, nous pensons

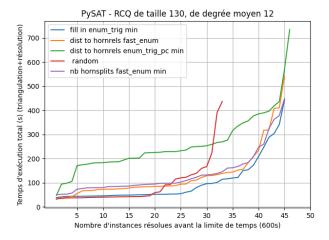


FIGURE 2 – Résultats prometteurs, différente combinaison de mesures et méthodes d'obtention des triangulations sont proposées

angulations avec les mesures les plus faibles. Nous les comparerons notamment avec des triangulations aléatoires et le graphe complet comme point de référence.

De plus, nous allons générer nos instances en créant Soit un RCQ $\mathcal{N} = (V, C)$, nous générons une liste de 290 des graphes aléatoires de tailles variables. Nous pourrons rendre le nouveau RCQ satisfaisable, ou le rendre plus "dur" en utilisant un jeu de relations connu pour être plus difficile pour les solveurs.

> Dans la Figure 2, montrant des résultats prometteurs, 295 nous constatons que nos méthodes font mieux que les triangulations aléatoires dans la plupart des cas.

4 Conclusion

Cette étude pourra donner des pistes d'amélioration pour raisonner avec les Réseaux de Contraintes Qualitatives (RCQ), en définissant différentes mesures de qualité pour les triangulations de leurs graphes de contraintes et observer empiriquement l'impact de ces mesures sur l'efficacité des solveurs utilisés. Les résultats obtenus pourraient permettre d'esquisser une définition empirique d'une "bonne" triangulation et de déterminer si le temps consacré au calcul de cette triangulation serait justifié. Ces réponses pourront, ou non, motiver la création de nouveaux solveurs utilisant ces décompositions arborescentes intelligentes.

par l'initiative TransLIRMM du LIRMM UMR 5506, qui soutient des stages de Master 2 relevant d'un axe scientifique transverse.

Références

320

340

350

365

- ³¹⁵ [1] James F Allen: Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26:832–843, 1983.
 - [2] Nouhad Amaneddine, Jean-François Condotta et Michael Sioutis: Efficient approach to solve the minimal labeling problem of temporal and spatial qualitative constraints. *In IJCAI*, 2013.
 - [3] Manuel Bodirsky et Stefan Wölfl: Rcc8 is polynomial on networks of bounded treewidth. In^{375} IJCAI, 2011.
- [4] Caroline Brosse, Vincent Limouzy et Arnaud Mary: Polynomial Delay Algorithm for Minimal Chordal Completions. In Mikołaj Bojańczyk, Emanuela Merelli et David P. Woodruff, 380 éditeurs: 49th International Colloquium on Automata, Languages, and Programming (ICALP 2022), volume 229 de Leibniz International Proceedings in Informatics (LIPIcs), pages 33:1–33:16, Dagstuhl, Germany, 2022. Schloss Dagstuhl Leibniz-Zentrum für Informatik.
- [5] Nofar CARMELI, Batya KENIG et Benny KIMEL-FELD: Efficiently enumerating minimal triangulations. In PODS, 2017.
 - [6] Jean-François Condotta et Dominique D'Al-MEIDA: Consistency of qualitative constraint 390 networks from tree decompositions. *In TIME*, 2011.
 - [7] Reinhard DIESTEL: Graph Theory, 4th Edition, volume 173 de Graduate texts in mathematics. Springer, 2012.
- ¹⁴⁵ [8] Gabriel Andrew DIRAC: On rigid circuit graphs. In Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg, volume 25, pages 71–76, 1961.
 - [9] Frank Dylla, Jae Hee Lee, Till Mossakowski, Thomas Schneider, André Van Delden, Jasper 400 Van De Ven et Diedrich Wolter: A survey of qualitative spatial and temporal calculi: algebraic and computational properties. *ACM Computing* Surveys, 50:1–39, 2017.
- 355 [10] Frank DYLLA et Jan Oliver WALLGRÜN: Quali-405 tative spatial reasoning with conceptual neighborhoods for agent control. *Journal of Intelligent* and Robotic Systems, 48:55-78, 2007.
- [11] Jinbo Huang: Compactness and its implications for qualitative spatial and temporal reasoning. In KR, 2012.
 - [12] Jinbo Huang, Jason Jingshi Li et Jochen Renz: Decomposition and tractability in qualitative spatial and temporal reasoning. Artificial Intelligence, 195:140–164, 2013.

- [13] Orestis Kostakis et Panagotis Papapetrou: On searching and indexing sequences of temporal intervals. *Data mining and knowledge discovery*, 31:809–850, 2017.
- 370 [14] Orestis Kostakis, Nikolaj Tatti et Aristides Gio-Nis: Discovering recurring activity in temporal networks. *Data Mining and Knowledge Discovery*, 31:1840–1871, 2017.
 - [15] Nikhil Krishnaswamy, Scott Friedman et James Pustejovsky: Combining deep learning and qualitative spatial reasoning to learn complex structures from sparse examples with noise. *In AAAI*, 2019.
 - [16] Gérard LIGOZAT: Qualitative spatial and temporal reasoning. John Wiley & Sons, 2013.
 - [17] Gérard LIGOZAT et Jochen RENZ: What is a qualitative calculus? A general framework. In PRICAI, 2004.
 - [18] Carsten Lutz et Maja Milicic: A tableau algorithm for DLs with concrete domains and GCIs. *In Description Logics*, 2005.
 - [19] Robert Moskovitch et Yuval Shahar: Fast time intervals mining using the transitivity of temporal relations. *Knowledge and Information Systems*, 42:21–48, 2015.
 - [20] Bernhard Nebel: Solving hard qualitative temporal reasoning problems: Evaluating the efficiency of using the ORD-Horn class. *Constraints*, 1:175–190, 1997.
 - [21] Bernhard Nebel et Hans-Jürgen Bürckert: Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. *Journal of the ACM*, 42:43–66, 1995.
 - [22] Charalampos Nikolaou et Manolis Koubara-Kis: Fast consistency checking of very large realworld rcc-8 constraint networks using graph partitioning. *In AAAI*, 2014.
 - [23] David A RANDELL, Zhan Cui et Anthony G Cohn: A spatial logic based on regions and connection. *In KR*, 1992.
 - [24] Jochen Renz et Gérard Ligozat : Weak composition for qualitative spatial and temporal reasoning. In CP, 2005.
 - [25] Jochen Renz et Bernhard Nebel: Efficient methods for qualitative spatial reasoning. *Journal of Artificial Intelligence Research*, 15:289–318, 2001.
 - [26] Michael Sioutis et Jean-François Condotta: Tackling large qualitative spatial networks of scale-free-like structure. *In SETN*, 2014.

- ⁴¹⁵ [27] Michael Sioutis et Diedrich Wolter: Dynamic branching in qualitative constraint-based reasoning via counting local models. *Information and Computation*, 281:104787, 2021.
- [28] Jakob Suchan et Mehul Bhatt: Semantic question-answering with video and eye-tracking data: Ai foundations for human visual perception driven cognitive film studies. *In IJCAI*, 2016.
- [29] Jakob Suchan, Mehul Bhatt, Przemysław Wałega et Carl Schultz: Visual explanation by high-level abduction: On answer-set programming driven reasoning about moving objects. *In AAAI*, 2018.
 - [30] Mihalis Yannakakis: Computing the minimum fill-in is NP-complete. SIAM Journal on Algebraic Discrete Methods, 2:77–79, 1981.

Annexe

430

445

L'algèbre d'intervalle (AI) [1] considère les intervalles de temps (en tant qu'entités temporelles) et l'ensemble des relations de base $\mathsf{B} = \{eq, \, p, \, pi, \, m, \, mi, \, o, \, oi, \, s, \, si, \, d, \, di, \, f, \, fi\}$ pour coder les connaissances sur les relations temporelles entre les intervalles sur la ligne réelle, comme décrit dans la Figure 3.

$$\begin{array}{cccc} & \underset{x}{\operatorname{precedes}} & \underset{y}{p} & & \underset{x}{\operatorname{meets}} & \underset{y}{m} & & \underset{x}{\operatorname{overlaps}} & o & \underset{x}{\operatorname{starts}} & s \\ & & \underset{x}{\underbrace{\operatorname{during}}} & d & & \underset{y}{\operatorname{finishes}} & f & \underset{x}{\underbrace{\operatorname{equals}}} & eq \\ & & & & & \\ \end{array}$$

FIGURE 3 – Une représentation des 13 relations de base b de AI, chacune reliant deux intervalles potentiels x et y comme dans x b y; l'inverse de b, c'est-à-dire b^{-1} , peut être noté bi et est omis dans la figure.

La décomposition arborescente d'un graphe est définie comme tel :

- 440 **Définition 5** Une décomposition arborescente d'un graphe G = (V, E) est un tuple $\mathcal{T} = (X, T)$, tel que \mathcal{T} est un arbre, et $X = \{X_1, \ldots, X_r\}$ est une collection de sous-ensembles de sommets de V, que l'on appelle sac, qui respecte ces propriétés :
 - 1. Pour tout $v \in V$ il y a au moins un nœud X_i de \mathcal{T} tel que $v \in X_i$.
 - 2. Pour tout $(u, v) \in E$ il existe un nœud X_i de \mathcal{T} tel que $u \in X_i$ et $v \in X_i$.
- 3. Soit X_i, X_j, X_k trois nœuds de \mathcal{T} tels que X_j est sur le chemin entre X_i et X_k . Alors, si $v \in V$ appartient à X_i et à X_k il doit nécessairement appartenir à X_j .