

**Plate-forme AFIA / Grenoble, du 2 au 6 juillet 2007**

TROISIÈME ATELIER :

## Représentation et raisonnement sur le temps et l'espace (RTE 2007)



Maroua Bouzid  
Florence Le Ber  
Gérard Ligozat  
Odile Papini



## Avant-Propos

La représentation du temps et de l'espace ainsi que les modèles de raisonnements associés sont des thèmes largement étudiés en informatique, d'une manière générale et en intelligence artificielle en particulier. Ces thèmes sont de plus en plus importants, avec l'engouement que connaît l'informatique ubiquitaire, les grilles de calcul, l'Internet, où une très grande quantité d'informations et de services pouvant évoluer au cours du temps sont disponibles. Les techniques temporelles et/ou spatiales s'appliquent particulièrement sur les problèmes suivants :

- la gestion des grandes quantités de données, l'analyse et la fouille de ces données,
- la simulation et l'analyse de l'évolution temporelle des processus, l'évaluation de la sécurité et la sûreté ;
- la gestion dynamique des connaissances ;
- la gestion de l'espace, la prévention des risques naturels, la modélisation des systèmes dynamiques complexes, etc.

En complément aux contributions dans les conférences généralistes dans les domaines de l'intelligence artificielle, des systèmes d'informations, des bases de données, du traitement automatique de la langue naturelle, des systèmes temps réel et de l'informatique théorique, l'atelier « Représentation et raisonnement sur le temps et l'espace » a émergé comme l'atelier national qui réunit les chercheurs français s'intéressant aux dimensions temporelle et spatiale pour discuter des travaux en cours, projets et idées. L'atelier RTE, qui s'est tenu pour la première fois en 2005 à Nice dans le cadre de la plateforme AFIA, puis en 2006 à Nantes dans le cadre de la semaine de la connaissance, est ouvert à la présentation de travaux de chercheurs et doctorants portant sur l'un ou plusieurs des thèmes suivants : modélisation du temps, modélisation de l'espace, raisonnement spatial qualitatif, raisonnement temporel qualitatif, traitement du langage naturel, représentation de connaissances, applications, etc. Les présentations peuvent aborder ces concepts selon différentes problématiques.

Pour cette troisième manifestation qui se tient à Grenoble dans le cadre de la plateforme AFIA, nous avons recueilli 6 soumissions qui présentent des travaux dans les différents champs cités : modélisation des phénomènes itératifs, traitement des données temporelles, recherche d'itinéraires, raisonnement spatial qualitatif, relations spatiales, contraintes spatiales ou temporelles, avec des applications variées : traitement automatique de langue naturelle, reconnaissance d'images, etc. Ces présentations de travaux seront accompagnées d'une conférence invitée, présentée par Isabelle Bloch (ENST) sur la modélisation floue des relations spatiales et de leur utilisation pour combler le fossé sémantique et guider la reconnaissance de structures dans les images. De plus, cette année, l'atelier a la particularité de partager une session avec les journées nationales « Intelligence Artificielle Fondamentale », instituées à l'initiative du thème éponyme du GDR I3.

Nous souhaitons que les participants à cet atelier trouvent matière à idées et discussions dans l'ensemble des présentations et que ces discussions puissent se prolonger lors de futures manifestations. Finalement, nous remercions chaleureusement les membres du comité de programme de l'atelier ainsi que les membres du comité d'organisation de la plateforme AFIA 2007, sans lesquels cet atelier n'aurait pu se tenir.

Grenoble, le 2 juillet 2007

Maroua Bouzid, Florence Le Ber, Gérard Ligozat, Odile Papini

## **Comité de programme**

Maroua Bouzid (GREYC, Caen)  
Jean-François Condotta (CRIL, Lens)  
Florence Le Ber (CEVH, Strasbourg / LORIA)  
Gérard Ligozat (LIMSI, Orsay)  
Philippe Muller (IRIT, Toulouse)  
Odile Papini (LSIS, Marseille)

*RTE 2007*

## Table des matières

*Modélisation floue de relations spatiales : application à la reconnaissance de structures dans les images*

I. Bloch (LTCI, ENST) p. 7

*Vers une approche qualitative pour la reconnaissance des empreintes digitales*

N. Ben Aloui (LSIS, Univ. Toulon-Var), H. Glotin (LSIS, Univ. Toulon-Var),  
P. Herbrard (DCN, Toulon), O. Papini (LSIS, Univ. Méditerranée) p. 9

*Computing Data from Temporal Data*

N. Bidoit (LRI, Univ. Paris Sud) et F. Hantry (Rectorat de Paris) p. 19

*Eligibilité de contraintes pour la résolution de contraintes qualitatives temporelles et spatiales*

J.-F. Condotta (CRIL, Univ. Artois), G. Ligozat (LIMSI, Univ. Paris Sud),  
M. Saade (CRIL, Univ. Artois) p. 29

*Deux approches pour la détection d'itinéraires dans du texte*

P. Loustau, J. Lesbegueries (LIUPPA, Univ. Pau) p. 41

*Une approche cognitive de l'itération et sa modélisation*

Y. Mathet (Greyc, Univ. Caen) p. 53

*Vers le Web sémantique géospatial avec ONTOAST*

A. Dia Miron, J. Gensel, M. Villanova-Oliver, H. Martin (IMAG, Univ. Joseph Fourier) p. 63

*RTE 2007*

# **Modélisation floue de relations spatiales**

## **Application à la reconnaissance de structures dans les images**

Isabelle Bloch

ENST - CNRS UMR 5141 LTCI  
Département Traitement du Signal et des Images  
46 rue Barrault  
F-75634 Paris Cedex 13  
Isabelle.Bloch@enst.fr

La reconnaissance de structures dans les images peut être guidée par les relations spatiales entre ces structures, lorsque celles-ci sont données dans une base de connaissances ou un modèle. C'est typiquement le cas des structures internes du cerveau, pour lesquelles de nombreuses descriptions anatomiques existent et impliquent fortement les relations spatiales. Celles-ci sont de plus relativement stables, y compris dans les cas pathologiques, ce qui en fait un support plus solide pour la reconnaissance que les caractéristiques des objets eux-mêmes (forme, taille...).

Afin de prendre en compte le caractère souvent intrinsèquement vague de ces relations (à droite de, entre...), ainsi que leur variabilité potentielle (même si elle est faible), nous proposons des modèles mathématiques de quelques relations spatiales, à la fois topologiques et métriques, à partir d'outils de morphologie mathématiques associés à la théorie des ensembles flous.

Nous montrons comment ces relations peuvent être exploitées pour guider la reconnaissance de structures dans des images. Les modèles, souvent exprimés sous forme symbolique, sont codés d'une part dans des ontologies, et d'autre part sous forme de graphes et d'hypergraphes, dans lesquels les noeuds représentent les structures et les arcs les relations spatiales entre ces structures. Dans le cadre de l'imagerie médicale, nous nous appuyons sur des extraits de l'ontologie anatomique FMA, que nous enrichissons d'une ontologie de relations spatiales. Ce formalisme permet de raisonner à un niveau générique et symbolique. Pour passer à l'interprétation des images, se pose le problème classique du fossé sémantique. Nous contribuons à le combler en associant à l'ontologie de relations spatiales une sémantique, propre au domaine et au contexte de l'application, sous la forme de représentations floues. Les paramètres de ces représentations sont appris sur des bases d'exemples. Ainsi, à un concept symbolique est associé un ensemble flou qui en donne la sémantique et qui permet de faire le lien avec les percepts visuels extraits des images.

Nous illustrons ces idées dans le cadre de la reconnaissance de structures internes du cerveau dans des images IRM 3D, pour des cas normaux et pathologiques. La reconnaissance est réalisée de manière séquentielle, chaque structure étant segmentée

et reconnue en exploitant ses relations spatiales aux structures préalablement reconnues.

## Références

- Atif J., Hudelot C., Fouquier G., Bloch & Angelini E. (2007). From Generic Knowledge to Specific Reasoning for Medical Image Interpretation using Graph-based Representations. *In International Joint Conference on Artificial Intelligence IJCAI'07, Hyderabad, India.*
- Atif J., Hudelot C., Nempont O., Richard N., Batrancourt B., Angelini E. & Bloch I. (2007). GRAFIP: A Framework for the Representation of Healthy and Pathological Cerebral Information. *In ISBI 2007, Washington DC, USA*, p. 205-208.
- Bloch I.(2005). Fuzzy Spatial Relationships for Image Processing and Interpretation: A Review. *Image and Vision Computing*, 23(2):89-110.
- Bloch I. (2006). Spatial Reasoning under Imprecision using Fuzzy Set Theory, Formal Logics and Mathematical Morphology. *International Journal of Approximate Reasoning*, 41:77-95.
- Colliot O., Camara O. & Bloch I. (2006). Integration of Fuzzy Spatial Relations in Deformable Models - Application to Brain MRI Segmentation. *Pattern Recognition*, 39:1401-1414.
- Hudelot C., Atif J. & Bloch I. (2006). Ontologie de relations spatiales floues pour l'interprétation d'images. *In Rencontres francophones sur la Logique Floue et ses Applications, LFA 2006, Toulouse, France*, p. 363-370.
- Hudelot C., Atif J. & Bloch I. (2007). An Ontology of Spatial Relations using Fuzzy Concrete Domains. *In AISB symposium on Spatial Reasoning and Communication, Newcastle, UK.*

# Vers une approche qualitative pour la reconnaissance des empreintes digitales

Nidhal Ben Aloui<sup>1,3</sup>, Hervé Glotin<sup>1</sup>, Patrick Hebrard<sup>3</sup>, Odile Papini<sup>2</sup>

<sup>1</sup> Laboratoire LSIS UMR 6168  
Université du Sud Toulon-Var B.P. 20 132 - 83 957 La Garde  
{benaloui,glotin}@univ-tln.fr

<sup>2</sup> Laboratoire LSIS UMR 6168, Université de Méditerranée,  
Campus de Luminy - 13 288 Marseille Cedex 09  
odile.papini@esil.univmed.fr

<sup>3</sup> Service Sécurité des Systèmes d'informations  
DCN - Combat Management Systems, Le Mourillon B.P. 403 - 83 055 Toulon  
{nidhal.ben-aloui,patrick.hebrard}@dcn.fr

**Résumé** : La reconnaissance des empreintes digitales est une technique importante dans le domaine de la biométrie. Les approches développées jusqu'à présent pour la reconnaissance des empreintes digitales reposent sur des méthodes quantitatives, cependant ces méthodes comportent un certain nombre de limitations.

Ce papier propose une nouvelle approche qualitative pour la représentation des empreintes basée sur l'algèbre des rectangles définie dans le cadre du raisonnement spatial en Intelligence Artificielle et propose des pistes pour une approche qualitative de la reconnaissance d'empreintes.

**Mots-clés** : Algèbre des rectangles, reconnaissance d'empreinte digitale, représentation qualitative, matching des minuties, biométrie.

## 1 Introduction

Dans le domaine de la sécurité informatique la biométrie a pour objet l'identification et l'authentification des personnes sur la base de caractéristiques biologiques, comportementales ou morphologiques.

Les techniques biométriques peuvent être classées en 3 catégories : les analyses biologiques comme le sang, l'ADN, ou la salive ; les analyses comportementales comme la dynamique de signature, la dynamique de frappe sur un clavier d'ordinateur, la parole, la démarche ; les technologies basées sur des analyses morphologiques comme les empreintes digitales, la forme de la main, la forme du visage, la forme de l'iris ou de la rétine (une rapide synthèse sur ces approches est proposée dans Poirel (2004)).

Les caractéristiques morphologiques présentent l'avantage d'être assez stables dans la vie d'un individu et sont moins sujettes aux variations que peuvent subir des caractéristiques comportementales. Même si elles ne sont pas aussi efficaces que les analyses

biologiques, elles ont, en revanche, l'intérêt de ne pas être invasives. Dans ce papier nous proposons de revisiter le codage et décodage de l'identité par empreintes digitales.

L'empreinte digitale (ED) est le modèle du relief cutané des doigts, ce relief se forme aléatoirement durant la période foetale. Son caractère aléatoire s'affranchit des risques de ressemblance entre individus ayant le même patrimoine génétique : la probabilité pour que deux personnes aient la même ED est d'environ 1/64 milliards. Les ED sont utilisées depuis l'antiquité comme signatures, mais c'est depuis la fin du XIX<sup>e</sup> siècle qu'elles sont utilisées pour l'identification de personnes, en particulier, grâce aux travaux réalisés en anthropométrie par A. Bertillon (1886). La numérisation de l'information ainsi que l'accroissement des capacités de calcul, ont permis l'automatisation de la reconnaissance d'empreintes qui se ramène à un problème de recherche et de reconnaissance de formes sur une image 2D.

Un grand nombre d'approches ont été définies (Maio *et al.*, 2003) pour la reconnaissance automatique des ED. Parmi elles, l'approche basée sur la corrélation qui relève du traitement d'images. Le principe est simple : deux images d'empreintes digitales sont superposées et la corrélation entre les pixels correspondants aux deux images est calculée pour différents alignements. Les ED sont caractérisées par des motifs particuliers, les minuties, dont les positions relatives dans le plan caractérisent l'identité. Certaines de ces approches ont été efficacement implantées et ont donné lieu à des produits commerciaux présents sur le marché. Cependant il persiste plusieurs limites : la variation "intra-class", c'est-à-dire l'instabilité d'une ED chez une même personne, l'usure des sillons, le déplacement, la rotation, la distortion et la pression du doigt lors de la capture, le bruit lié au capteur, la saleté, les erreurs d'extraction des traits liées aux techniques probabilistes des algorithmes employés. Tous ces facteurs rendent la tâche de reconnaissance difficile.

Nous proposons une nouvelle approche dite qualitative pour la représentation des ED en ne considérant plus de manière exacte la position planaire d'une minutie, mais plutôt un rectangle englobant cette position, traitant donc partiellement les problèmes de distortion et de déplacement du doigt. Ici une ED n'est plus représentée par un vecteur de coordonnées et de mesures numériques mais plutôt par un réseau de contraintes de l'algèbre des rectangles (Condotta, 2000), offrant ainsi la possibilité d'appliquer les techniques de l'Intelligence Artificielle pour résoudre le problème d'identification.

L'article est organisé comme suit. La section 2 présente la caractérisation d'une empreinte digitale par des traits saillants, appelés les minuties, puis présente les méthodes de traitement de l'image, d'extraction de minuties, de reconnaissance d'empreinte de manière quantitative, enfin, elle rappelle brièvement l'algèbre des rectangles. La section 3 propose une nouvelle approche qualitative de représentation des empreintes digitales. Les avantages et le coût de cette nouvelle représentation sont présentés avant de conclure, en section 4, sur des pistes de recherche pour une approche qualitative de la reconnaissance d'empreintes digitales.

## 2 Préliminaires

### 2.1 Les empreintes digitales

Une empreinte digitale est une marque laissée par l'extrémité des doigts sur des objets qui reflète le relief cutané des doigts.

Les systèmes biométriques reposant sur les empreintes digitales, dans le cadre des approches basées sur les minuties, effectuent l'identification selon les différentes étapes : capture de l'image de l'empreinte, pré-traitement de l'image, extraction des minuties, représentation des minuties, reconnaissance de l'empreinte à partir des minuties.

#### 2.1.1 Capture de l'image de l'empreinte

La capture de l'image d'une empreinte digitale permet d'obtenir une image numérisée de l'empreinte qui reflète les lignes en contact avec le capteur, appelées arêtes et les creux entre les lignes, appelées vallées. Cette étape de numérisation de l'image d'une empreinte est réalisée selon différentes techniques et utilise différents types de capteurs : optiques, ultrasoniques, de champ électrique, de capacité de température.

L'analyse de l'image d'une empreinte digitale permet de déterminer des points caractéristiques sur les lignes, appelés minuties. Les deux principaux types de minuties sont les terminaisons, c'est-à-dire, les points où une ligne se termine et les bifurcations, c'est-à-dire, les points où la ligne donne naissance à deux lignes distinctes. Bien qu'il existe d'autres types de minuties, ces deux types de minuties, présentés dans la figure 1 sont suffisants pour caractériser une empreinte digitale.

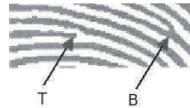


FIG. 1 – Les 2 types de minuties d'une empreinte digitale : (T) terminaison et (B) bifurcation

#### 2.1.2 Reconnaissance d'empreintes digitales à partir de minuties

La reconnaissance d'empreintes digitales à partir de minuties s'effectue selon différentes étapes (Maio *et al.*, 2003). Tout d'abord un pré-traitement de l'image est réalisé afin d'améliorer la qualité de l'image par l'élimination du bruit et la mise en évidence des minuties pour en faciliter l'extraction. Il se déroule selon différentes étapes, segmentation, binarisation et amincissement, illustrées dans la figure 2. La phase d'extraction permet de détecter les minuties, d'en donner le type, de déterminer les coordonnées et l'orientation dans le plan.

Une empreinte digitale est ensuite définie comme étant un vecteur de minuties. Chaque minutie est caractérisée par trois composantes : son type (bifurcation ou terminaison), ses coordonnées dans le plan et un angle pour son orientation. Parmi les différentes méthodes de représentation des empreintes digitales, plusieurs méthodes quantitatives utilisent cette représentation pour la reconnaissance (Maio *et al.*, 2003).

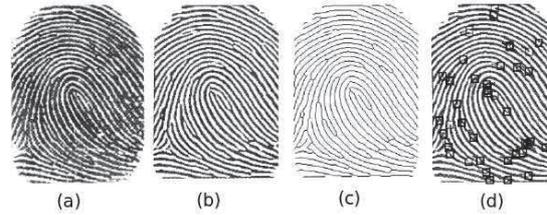


FIG. 2 – différentes phases de traitement d’une image d’empreinte digitale : (a) Image originale bruitée (b) Binarisation (c) Amincissement (d) Détection et extraction des minuties.

## 2.2 Algèbre des rectangles

Dans le domaine du raisonnement spatial en intelligence artificielle, parmi les approches qualitatives, l’algèbre des rectangles (Balbiani *et al.*, 1998) (Condotta, 2000) a été proposée pour raisonner sur des régions du plan. Il s’agit d’une extension de l’algèbre des intervalles. L’algèbre des intervalles a été introduite par Allen (Allen, 1983) pour le raisonnement temporel qualitatif. Il s’agit d’une algèbre basée sur 13 relations atomiques qui sont  $\mathcal{B}_{int} = \{b, m, o, s, d, a, bi, mi, oi, si, di, ai, eq\}$ . L’algèbre des rectangles considère comme entités spatiales des rectangles dont les côtés sont parallèles aux axes d’un repère orthogonal dans un espace euclidien de dimension 2. Les relations entre les rectangles sont définies à partir des relations de l’algèbre des intervalles d’Allen en considérant les projections des côtés des rectangles sur les axes du repère orthogonal. On note  $\mathcal{B}_{rec}$  l’ensemble des relations atomiques entre les rectangles et  $\mathcal{B}_{rec} = \{(A, B) \mid A, B \in \mathcal{B}_{int}\}$ . Soit  $x$  et  $y$  deux rectangles dont les projections sur les axes sont  $X_1$ , respectivement  $X_2$ , et  $Y_1$ , respectivement  $Y_2$ , les rectangles  $x$  et  $y$  satisfont la relation atomique  $(A, B)$  si et seulement si  $X_1$  et  $Y_1$  satisfont  $A$  et si  $X_2$  et  $Y_2$  satisfont  $B$ . L’ensemble des relations de l’algèbre des rectangles est noté  $2^{\mathcal{B}_{rec}}$  et chaque relation est une disjonction des relations atomiques. L’ensemble  $2^{\mathcal{B}_{rec}}$  est muni des opérations d’union ( $\cup$ ), d’intersection ( $\cap$ ), de composition ( $\circ$ ) et d’inverse ( $^{-1}$ ).

## 3 Représentation qualitative des empreintes digitales

Notre représentation qualitative repose sur une caractérisation d’empreinte digitale basée sur les minuties. Celle-ci consiste à considérer l’ensemble des minuties comme un ensemble de régions organisées spatialement dans le plan.

### 3.1 Représentation qualitative des minuties

#### 3.1.1 Types de minuties

Il existe pas moins de sept types de minuties. Cependant, les deux types de minuties les plus employées sont les terminaisons et les bifurcations qui sont suffisantes pour caractériser une empreinte digitale (Maio *et al.*, 2003).

### 3.1.2 Position des minuties

A partir de cette représentation nous proposons une nouvelle approche qui consiste à repérer ces minuties puis à les encadrer par des rectangles de taille identique. Les dimensions des rectangles peuvent être optimisées afin de maximiser le pouvoir discriminant des relations. Nous sommes actuellement en train de mener des expériences afin de déterminer les meilleures longueurs des côtés de ces rectangles. Le critère déterminant serait celui qui maximise les relations qualitatives discriminantes d'une ED.

Les minuties sont détectées après l'amincissement de l'image binaire de départ. Comme l'illustre la figure 3, les rectangles aux traits pleins représentent les terminaisons et les rectangles aux traits pointillés représentent les bifurcations.

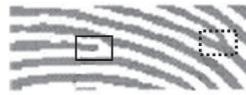


FIG. 3 – Positionnement d'un rectangle autour d'une bifurcation et d'une terminaison

Plusieurs implémentations efficaces de l'extraction des minuties sont disponibles. Nous avons retenu celle de A. Drygajlo et son équipe<sup>1</sup> qui fournit les coordonnées (x,y) des minuties dans le plan selon un repère orthogonal.

Après détection des minuties, nous encadrons les minuties par des rectangles comme suit. Pour une terminaison, la pointe de la terminaison coïncide avec le centre du rectangle cf. figure 4. Pour une bifurcation : l'intersection des arêtes coïncide avec le centre du rectangle cf. figure 5.



FIG. 4 – Centrage d'une terminaison.

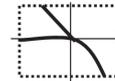


FIG. 5 – Centrage d'une bifurcation.

### 3.1.3 Orientation des minuties

Pour représenter l'orientation des minuties dans les rectangles englobants nous utilisons la représentation qualitative de l'orientation par les 8 relations proposées par (Freksa, 1992) et illustrées dans la figure 6. Par la suite, pour simplifier les notations, l'ensemble des 8 relations d'orientation est notée  $\mathcal{D} = \{sf, rf, rn, rb, sb, lb, ln, lf\}$ .

La figure 7 illustre l'orientation des minuties représentées dans les figures 4 et 5. On associe une orientation à une bifurcation en calculant la direction de l'ouverture du Y définie par les 8 orientations de la figure 6. On procède de la même manière pour une

<sup>1</sup>Ecole Polytechnique Fédérale de Lausanne, students Projects, équipe du Dr. Andrzej Drygajlo

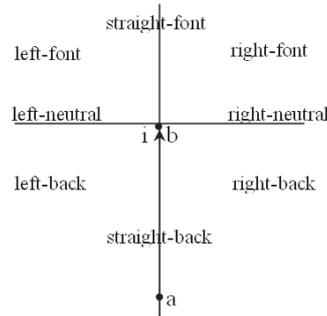


FIG. 6 – Représentation qualitative spatiale des relations d’orientations.

terminaison, sauf que nous considérons la direction du trait central de la terminaison cf. figure 4. Cette représentation de l’orientation est très intéressante car elle offre une

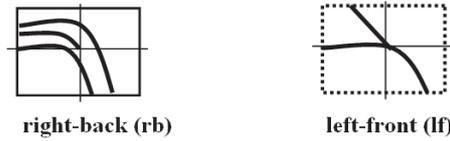


FIG. 7 – Orientation d’une minutie dans un rectangle.

souplesse d’incertitude supplémentaire sur l’orientation d’une minutie que la mesure d’angle des méthode quantitatives ne peut apportée.

### 3.2 Représentation qualitative d’une empreinte digitale

L’organisation dans le plan des minuties conduit à la représentation d’une empreinte digitale par un *réseau de contraintes qualitatives dans l’algèbre des rectangles* reflétant la position relative des rectangles encadrant les minuties dans le plan. Cette représentation est illustrée dans la figure 8. Une contrainte qualitative spécifie l’ensemble des relations de base admissibles entre deux rectangles (donc deux minuties), caractérisant ainsi leurs positions relatives.

Pour cela, nous considérons un sous-ensemble de l’algèbre des rectangles : en prenant l’hypothèse suivante : “*tous les rectangles sont de taille identique.*”. Soit  $h_1, h_2$  respectivement  $l_1, l_2$  les hauteurs respectivement les largeurs de deux rectangles  $R_1$  et  $R_2$ , selon notre hypothèse  $h_1 = h_2$  et  $l_1 = l_2$ .

Cette hypothèse nous conduit à considérer uniquement un sous-ensemble des 7 relations de base de l’algèbre d’Allen, noté  $S_{B_{int}}$  avec  $S_{B_{int}} \subseteq B_{int}$  et  $S_{B_{int}} = \{b, m, o, eq, oi, mi, a\}$ . Pour modéliser les positions relatives des projections des hauteurs et largeurs d’un rectangle sur les axes du repère orthogonal.

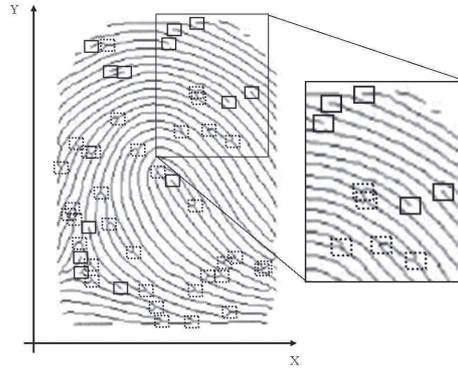


FIG. 8 – Recouvrement complet des minuties par des rectangles.

Cela nous conduit à considérer un sous-ensemble des relations de base de l'algèbre de rectangles, que nous notons  $S_{B_{rec}} = \{(A, B) \mid A \in S_{B_{int}}, B \in S_{B_{int}}\}$ . La figure 9 illustre l'ensemble des relations possibles entre deux rectangles selon l'axe des X. Nous

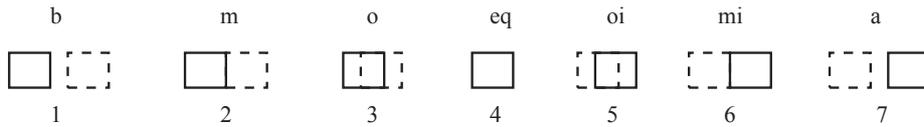


FIG. 9 – Représentation des 7 relations possibles entre 2 rectangles identiques (b :before, m :meets, o :overlaps, eq :equals, oi :overlaped-by, mi :meteed-by, a :after).

obtenons, de manière analogue, l'ensemble des relations possibles entre deux rectangles selon l'axe des Y. Un réseau de contraintes qualitatives, noté  $R_E$ , représentant une empreinte digitale est défini formellement comme suit :

### Définition 1

Un réseau de contraintes qualitatives  $R_E$  est un couple  $(V, C)$  où  $V$  est un ensemble fini de  $n$  variables  $v_0, \dots, v_{n-1}$  (avec  $n$  un entier strictement positif) et  $C$  est une application qui, à chaque couple  $(v_i, v_j)$  de variables de  $V$  associe un sous-ensemble  $C(v_i, v_j)$  de relations de base de l'algèbre des rectangles :  $C(v_i, v_j) \subseteq S_{B_{rec}}$ .

Pour simplifier les notations  $C(v_i, v_j)$  est noté  $C_{ij}$ . De plus  $C$  est tel que pour chaque  $i \in \{1, \dots, |V|\}$ ,  $C_{ii} = \{(eq, eq)\}$  et pour chaque  $i, j \in \{1, \dots, |V|\}$ ,  $C_{ij} = C_{ji}^{-1}$ .

### Exemple 1

Afin d'illustrer notre approche, nous montrons la représentation qualitative d'une partie de l'empreinte digitale (figure 10) constituée de 10 minuties. Sur cette figure, on remarque bien le choix de la numérotation, les minuties sont numérotées du haut vers

le bas et de gauche à droite. Les numéros pairs sont attribués aux bifurcations et les numéros impairs aux terminaisons.

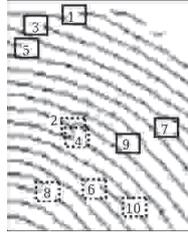


FIG. 10 – Représentation qualitative des minuties de la partie zoomée

Deux possibilités s’offrent pour représenter le réseau de contraintes qualitatives, soit une représentation matricielle, notée  $M_E$  (voir Tab. 1), soit une représentation sous forme de graphe de contraintes qualitatives, notée  $G_{R_E}$ .

### Définition 2

Un graphe de contraintes qualitatives de l’algèbre des rectangles, noté  $G_{R_E}$ , est un couple  $(N, R)$  où  $N$  est un ensemble de noeuds qui représentent les rectangles encadrant les minuties et  $R$  est un ensemble d’arcs représentant les relations spatiales entre ces rectangles. Chaque noeud est étiqueté par un couple  $(i, d)$  où  $i$  est un entier qui code le type de noeud : un entier impair pour les bifurcations et un entier pair pour les terminaisons et  $d \in \mathcal{D}$  représente l’orientation de la minutie, où  $\mathcal{D}$  est défini en 3.1.3. Chaque arc est étiqueté par le couple  $(v_i, v_j) \in S_{B_{rec}}$  qui représente une relation de base de l’algèbre des rectangles.

Nous illustrons ces représentations sur une partie de l’empreinte digitale, présentée dans la figure 10.

### Exemple 2

La table 1 illustre la représentation matricielle de la partie de l’empreinte digitale de la figure 10. Il s’agit d’une matrice  $10 \times 10$  pour les 10 noeuds. On remarque que la matrice est triangulaire, cela est dû au fait que toutes les relations sont symétriques.

La figure 11 illustre le graphe de contraintes qualitatives dans l’algèbre des rectangles correspondant à la partie de l’empreinte digitale de la figure 10. Chaque noeud est étiqueté par un entier et une relation d’orientation. Chaque arc est étiqueté par une relation de l’ensemble  $S_{B_{rec}}$ . Pour une meilleure lisibilité du graphe  $G_{R_E}$  nous n’avons pas représenté tous les arcs, mais seulement les arcs nécessaires à la représentation, les autres pouvant être déduits par transitivité.

## 3.3 Coût de la représentation qualitative d’une empreinte digitale

La représentation qualitative d’une empreinte digitale pose la question du coût d’une telle représentation. Le codage nécessite : 1 bit pour le type (2 types), 3 bits pour la

	1	2	3	4	5	6	7	8	9	10
1	-	(oi,a)	(a,oi)	(oi,a)	(a,a)	(o,a)	(b,a)	(a,a)	(b,a)	(b,a)
2		-	(a,b)	(o,oi)	(a,b)	(b,a)	(b,oi)	(a,a)	(b,a)	(b,a)
3			-	(b,a)	(oi,a)	(b,a)	(b,a)	(o,a)	(b,a)	(b,a)
4				-	(a,b)	(b,a)	(b,o)	(a,a)	(b,oi)	(b,a)
5					-	(b,a)	(b,a)	(b,a)	(b,a)	(b,a)
6						-	(b,b)	(a,oi)	(b,b)	(b,a)
7							-	(a,a)	(a,a)	(a,a)
8								-	(b,b)	(o,a)
9									-	(o,a)
10										-

TAB. 1 – Représentation matricielle des minuties de l'exemple fig. 10.

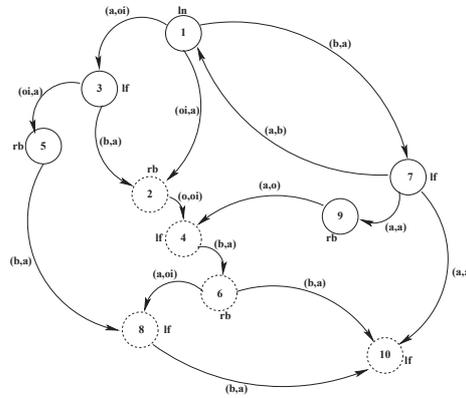


FIG. 11 – Graphe de contraintes qualitatives des minuties de l'exemple fig. 10

direction (8 directions), 3 bits pour coder la position en X (7 positions), 3 bits pour coder la position en Y (7 positions).

La somme des 10 bits obtenus représente le coût de représentation d'une minutie, noté  $\delta_i$ . Nous pouvons donner une fonction de coût :

### Définition 3

Soit  $\delta_i$  le coût de représentation d'une minutie et soit  $n > 0$  le nombre total de minuties, la fonction  $f$  de coût de représentation d'une empreinte digitale  $E$  est donnée par la formule  $f(E) = \sum_{i=0}^n \delta_i + \partial$  où  $\partial$  représente le coût pour coder les  $\frac{n(n-1)}{2}$  relations entre les  $n$  minuties. Avec  $\partial = \sum_{i=0}^{\frac{n(n-1)}{2}} \varphi_i$  où  $\varphi_i$  représente le coût nécessaire pour coder un couple de relations  $(v_i, v_j) \subseteq S_{B_{rec}}$ .

Des études statistiques montrent qu'il y a au plus 100 minuties dans une empreinte digitale. En général pour pouvoir distinguer deux empreintes digitales il faut un mini-

mum de 7 minuties et un maximum de 12 minuties. Cette représentation nous semble donc raisonnable pour un stockage sur carte à puce.

### Example 3

Pour  $n = 100$  minuties il y a 450 couples de relations à calculer (les  $\frac{n(n-1)}{2}$  minuties). Un couple de relation  $(v_i, v_j)$  est codé sur  $\varphi_i = 6$  bits (car il y a  $2^6 > 49$  relations), donc 450 couples sont codés sur  $\partial = 2700$  bits. Le coût de représentation d'une minutie est de  $\delta_i = 4$  bits, donc 100 minuties seront codées sur 400 bits. Ce qui donne un total de  $f(E) = 2700 + 400 = 3100$  bits, soit 387,5 octets, ce qui est raisonnable pour un stockage sur une carte à puce.

## 4 Conclusion

Dans cet article nous proposons une nouvelle représentation qualitative utilisant l'algèbre des rectangles pour représenter un ensemble de minuties. L'empreinte digitale est ainsi représentée par un réseau de contraintes qualitatives stocké sous forme de graphe de contraintes qualitatives.

Notre approche présente plusieurs avantages : elle permet de décrire à la fois avec souplesse et précision les positions et orientations des minuties ainsi que leurs relations dans le plan. Elle permet de s'affranchir partiellement des problèmes de variation "intra-classe", de distorsion, rotation et déplacement.

Le faible coût de cette représentation, (387,5 octets) nous encourage à poursuivre nos recherches pour proposer une approche qualitative de reconnaissance d'empreintes digitales en nous basant sur des travaux réalisés dans le domaine de l'Intelligence Artificielle (Ber *et al.*, 2006).

Nous envisageons plusieurs approches. La première consisterait à ordonner les minuties, bifurcations puis terminaisons, à reconnaître une minutie, à partir de laquelle les autres seraient reconnues. La deuxième consisterait à diviser le plan de l'image de l'empreinte digitale en strates et à rechercher la plus forte correspondance au sein d'une même strate, ou à mieux exploiter dans un cadre qualitatif l'information des positions des minuties au centre morphologique. Par ailleurs, nous pensons à des approches reposant sur des méthodes de recherche d'isomorphismes de graphes et d'appariements de graphes.

## Références

- ALLEN J. (1983). Maintaining knowledge about temporal intervals. *Communication of the ACM*, **26**, 832–843.
- BALBIANI P., CONDOTTA J. F. & DEL CERRO L. F. (1998). A model for reasoning about bidimensional temporal relations. *KR '98 Proc. of the 6th Int. Conf.*, p. 124–130.
- BER F. L., LIGOZAT G. & PAPINI O. (2006). *Raisonnements sur l'espace et le temps*. Lavoisier : Hermes.
- BERTILLON A. (1886). De l'identification par les signalements anthropométriques. *AAC*.
- CONDOTTA J. (2000). *Problèmes de satisfaction de contraintes : algorithmes et complexité*. PhD thesis, Université Paul Sabatier (Toulouse III).
- FREKSA C. (1992). Using orientation information for qualitative spatial reasoning. In *Spatio-Temporal Reasoning*, p. 162–178.
- MAIO D., MALTONI D., JAIN A. K. & PRABHAKAR S. (2003). *Handbook of Fingerprint Recognition*. New York : Springer Verlag.
- POIREL C. (2004). *État de l'art sur la biométrie*. Rapport interne, Université du Sud Toulon VAR.

# Computing Temporal Data from Temporal Data

Nicole Bidoit<sup>1</sup>, François Hantry<sup>2</sup>

<sup>1</sup> Univ. Paris-Sud, UMR 8623, Orsay F-91405  
CNRS, Orsay F-91405  
bidoit@lri.fr

<sup>2</sup> Rectorat de Paris

**Abstract** : The paper introduces and investigates relational temporal machine (RTM) as a general abstract model for temporal querying. The RTM devices subsume most temporal query languages that have emerged in the literature. A first contribution of the paper is to provide two simplified forms for our machines, namely extended one-tape RTMs and one-tape RTMs. Another contribution is to establish connections between RTMs and the T-WHILE and TS-WHILE extensions of FO based on complexity criteria.

**Mots-clés** : Temporal databases, query languages, generic machines

## 1 Introduction

Emerging applications such as the web, bioinformatics, medical applications, multimedia applications, animation, simulation, personal information management, data flow processing, streaming, are intensively managing temporal or ordered information. Thus in many computer science area, modeling and manipulating temporal and ordered information turns out to be a central issue.

Suitable database models and query languages (9; timecenter) have been developed to store and support temporal information management. The languages mainly target snapshot queries (what are the data available at some time point?), history extractions (what is the evolution of some data?) and combinations.

Most temporal query languages are based on extensions of first-order logic (FO) or SQL. Two central classes of formal temporal languages have been considered so far: (1) Linear temporal logic TL (10; 9) and its extensions ETL (15),  $\mu$ TL (14), T-WHILE and T-FIXPOINT (1; 6) allow one to query temporal databases represented as finite sequences of database states; (2) Time-stamped first-order logic TS-FO and its extension TS-WHILE (1) allows to query temporal databases represented by time-stamped relations.

The aim of this paper is to introduce and investigate relational temporal machines (RTM) as a general-purpose computation model for temporal querying. The first issue is to provide an abstract model of temporal querying. Indeed, the well-known formal temporal languages TL and TS-FO have very strong limitations: they provide the ability to define mappings from temporal instances to relations (tables) and lack

the ability to map temporal instances to temporal instances, in a general manner. The T-WHILE extension of FO shares the same limitation although the language allows for computing auxiliary temporal instances in a restricted manner<sup>1</sup>. Indeed, RTMS have the ability to map temporal instances to temporal instances and thus these devices subsume most temporal query languages. The second issue is to demonstrate that relational temporal machines are of special interest for studying, understanding and extending some of the features of known temporal languages. One of the contribution of our paper is to establish connections between RTMS and the T-WHILE and TS-WHILE extensions of FO.

RTMS are based on Turing machines and have been inspired by relational machines (RM) (2) known as a very powerful formal tool for studying relational queries and their complexity. The RTM model is a rather simple and intuitive extension of the RM, given that we assume temporal instances to be finite sequences of database states. An RTM machine uses tapes to store the input temporal instance (input tape) and the auxiliary temporal instances (auxiliary tapes). Roughly, the contents of an RTM cell is a (set of) relation(s) of fixed arity rather than a bit in a Turing machine cell. The interactions or transitions of an RTM can modify the auxiliary tapes either by changing the contents of (relations stored in) a cell through a first order (FO) query, or by adding/removing cells. The input is accepted if the machine halts in an accepting state. The output is the temporal instance stored in one of its auxiliary tape.

The first results presented in the paper are dedicated to analysing the usefulness of different components of our devices: number of registers and number of auxiliary tapes. We then investigate some complexity classes of temporal computation by RTMS. Our complexity measures are with respect to length of the input temporal instance (tape). We focus on space complexity: we intend to measure the size (length) of the auxiliary memory needed to compute some temporal query. The paper first investigates linear-space and polynomial-space RTM. Although T-WHILE and TS-WHILE are standard temporal query language (they output static tables), they make use of temporal auxiliary instances during their computation. Thus it makes sense to compare T-WHILE and TS-WHILE with RTM. Nicely enough, it turns out that T-WHILE (resp. TS-WHILE) is complete in the class of linear-space (resp. polynomial-space) RTMS.

Constant-space RTMS are devices using one auxiliary store. A store is a one-cell tape. Such devices are called relational register machine (RRM), because registers are needed in that case. Investigating RRM leads us to study new languages based on T-WHILE. Our main contribution provides a temporal query language, based on T-WHILE, which is complete w.r.t. RRMS.

This paper is a short version of (5). Due to space limitation, the proof of the results are not presented here. Sketches of proof can be found in (5).

---

<sup>1</sup>Auxiliary temporal instances in T-WHILE programs need to have a number of states equal to that of the input temporal instance

## 2 Preliminaries

We informally review some notions and notations related to relational databases (db) and query languages. We assume the reader familiar with both first-order logic FO and with the usual definitions of *relation schema*, *database schema* and *instances*. In the whole paper, we assume a unique domain. In the context of query specification, the db schema  $\mathcal{R}$  (resp.  $\mathcal{S}$ ) is a set of relation schemas and is used, in general, as the input db schema (resp. auxiliary db schema).

An implicit temporal instance  $\mathcal{I}$  over the db schema  $\mathcal{R}$  is a finite sequence  $I_1, \dots, I_n$  of finite instances over  $\mathcal{R}$ . The size of  $\mathcal{I}$ , denoted by  $|\mathcal{I}|$  is the size of the sequence  $I_1, \dots, I_n$ , that is  $n$ . For each  $i \in [1..n]$ ,  $I_i$  is called the state of  $\mathcal{I}$  at time point  $i$ . The instance  $I_i(R)$  of the relation schema  $R$  at time point  $i$  is also denoted by  $\mathcal{I}[i](R)$ . The active domain of  $\mathcal{I}$ , denoted by  $adom(\mathcal{I})$ , is set of domain elements appearing in  $\mathcal{I}$ . In the paper,  $\vec{x}$  represents a tuple of (data) variables whose arity is clear from the context, and  $\nu$  is a valuation of  $\vec{x}$  ranging over the active domain. We assume w.l.o.g. that the db schema  $\mathcal{R}$  includes two 0-ary relations (propositions) *First* and *Last* such that  $\mathcal{I}[i](First)=true$  iff  $i=1$ , and  $\mathcal{I}[i](Last)=true$  iff  $i=n$ . We denote  $\emptyset_{\mathcal{R}}^n$  the temporal instances over  $\mathcal{R}$  of size  $n$  whose states are all empty, and in particular,  $\emptyset_{\mathcal{R}}^0$  the temporal instance of size 0.

The temporal query language **T-WHILE** (1) is an extension of the well-known **WHILE** query language for static databases (8). It is built using imperative mechanisms like assignments, loops plus temporal moves. Variants of this language have been investigated in (6).

A program over  $\mathcal{R}$  is specified by a sequence of *declarations* specifying the auxiliary db schema  $\mathcal{S}$ , followed by a sequence of *instructions*. An auxiliary schema  $S \in \mathcal{S}$  can either be declared **shared** or **private**. Intuitively, instances of auxiliary shared schema are identical in all states of a temporal instance, that is if  $\mathcal{J}$  is a temporal instance over  $\mathcal{S}$  and  $S$  is shared then  $\mathcal{J}[i](S)=\mathcal{J}[j](S)$  for all  $i, j \in [1..|\mathcal{J}|]$ . No such restriction is enforced for private schemas. The instructions of a program are:

[temporal moves] **left** and **right**.

[assignment]  $S := \varphi(\vec{x})$ , where  $S \in \mathcal{S}$  and  $\varphi$  is a FO formula over  $\mathcal{R} \cup \mathcal{S}$  with free variables  $\vec{x}$ .

[while loop] **while** *Cond* **do** *Body* **end**, where *Cond* is a closed FO formula over  $\mathcal{R} \cup \mathcal{S}$  and *Body* a sequence of instructions. While loops can be nested.

**EXAMPLE 2.1** The program  $\mathcal{P}$  below uses an auxiliary shared proposition  $B$ .

**while**  $\neg Last$  **do** { **right** ;  $B := \neg B$  } **end** ;

The value of  $B$  returned by  $\mathcal{P}$  is *true* iff the size  $n$  of the temporal input  $\mathcal{I}$  is even.

Intuitively, the evaluation of a program  $\mathcal{P}$  over a temporal instance  $\mathcal{I}$  proceeds as follows. It uses a temporal cursor, denoted by *ctp*, which is assigned initially to 1. A temporal instance  $\mathcal{J}$  over the auxiliary db schema  $\mathcal{S}$  is initialized with the empty temporal instance  $\emptyset_{\mathcal{S}}^n$ . Instructions of  $\mathcal{P}$  are executed sequentially. Each instruction has some effect either on the instance  $\mathcal{J}$  or on *ctp* as explained below:

- **left** (resp. **right**) decreases (resp. increases) *ctp* by 1. If  $ctp=1$  (resp.  $ctp=n$ ) before the execution of **left** (resp. **right**) then *ctp* remains unchanged.

- $S: =\varphi(\vec{x})$  changes the instance  $\mathcal{J}$  only over  $S$ . Let  $ans$  be the answer of the FO query  $\varphi(\vec{x})$  evaluated over the static instance  $(\mathcal{I}+\mathcal{J})[ctp]$  (i.e. over the snapshot of  $\mathcal{I}+\mathcal{J}$  at the current time point  $ctp$ ).

- If  $S$  is private then  $\mathcal{J}$  changes over  $S$  only at  $ctp$ , that is  $\mathcal{J}[ctp](S)=ans$ ;

- if  $S$  is shared then  $\mathcal{J}$  changes over  $S$  everywhere, that is  $\mathcal{J}[i](S)=ans$  for all  $i \in [1..n]$ .

- **while**  $Cond$  **do**  $Body$  **end** executes all instructions of  $Body$  until either  $Cond$  becomes false, or until a repetition of the effect of the execution of  $Body$  is obtained. The effect of one loop iteration is measured by observing configurations. A configuration, at some point of the evaluation of a program, is the pair  $(\mathcal{J}, ctp)$  where  $\mathcal{J}$  is the current instance of the auxiliary db schema  $S$  and  $ctp$  is the current time point. Thus a while loop stops as soon as the configuration obtained after an iteration is the same as the one at its beginning. The reader should pay attention to the fact that a repetition of the auxiliary instances over  $S$  is not sufficient to exit the loop: if  $ctp$  changes then the evaluation of the body of the loop is reactivated. Of course, if neither the condition  $Cond$  of the loop becomes false nor the sequence of configurations reaches a repetition, then the semantics of both the while loop and the program it belongs to are undefined.

Finally, a T-WHILE query over  $\mathcal{R}$  is specified by a T-WHILE program  $\mathcal{P}$  over  $\mathcal{R}$  and a distinguish shared schema  $S_{rep}$  whose instance, at the end of the evaluation of  $\mathcal{P}$ , is intended to collect the answer of the query. Let us emphasize that T-WHILE is a query language that maps temporal instances to static relations although it computes auxiliary temporal instances. As a matter of fact, the size of these auxiliary temporal instances is exactly the size of the input temporal instance.

The temporal language TS-FO is defined while considering a temporal instance over  $\mathcal{R}$  as a standard instance over the time-stamped db schema  $\mathcal{R}^{est}$  where  $\mathcal{R}^{est}$  is obtained from  $\mathcal{R}$  by adding one attribute  $T$  of type "time" to each relation schema  $R \in \mathcal{R}$ . The domain of the attribute  $T$  is the set of positive integers. It is well known that any temporal instance  $\mathcal{I}$  over  $\mathcal{R}$  can be transformed into a time-stamped instance over  $\mathcal{R}^{est}$ , and vice versa. For the sake of our presentation, the set of FO formulas over the time-stamped schema  $\mathcal{R}^{est}$  is denoted  $FO^*$  in order to distinguish these formulas from FO formulas over  $\mathcal{R}$ .

The language TS-FO is defined as a restriction of  $FO^*$ : free variables in a TS-FO formula are required to be "data" variables.

**EXAMPLE 2.2** Let  $R$  be a unary schema. The formula  $\varphi(t_1, t_2)$  below is a  $FO^*$  formula using one data variable  $x$  and two free variables  $t_1$  and  $t_2$  of type "time":

$$not(t_1=t_2) \wedge (\forall x R^{est}(x, t_1) \leftrightarrow R^{est}(x, t_2)).$$

The evaluation of such a formula over a time-stamped instance returns pairs of integers (time points).

$\exists t_1 \exists t_2 \varphi(t_1, t_2)$  is a TS-FO query, called **twin**, which tests whether there exists two time points holding the same instance over  $R$ .

The temporal language TS-WHILE also assumes that the temporal instances over  $\mathcal{R}$  are given as time-stamped instances over  $\mathcal{R}^{est}$ . Then the language TS-WHILE over  $\mathcal{R}$  is simply defined as the language WHILE over  $\mathcal{R}^{est}$ . Thus a TS-WHILE program is build

using auxiliary relation schemas, assignments of the form  $S := \varphi$  where  $\varphi \in \text{FO}^*$ , and while loops of the form **while** *Cond* **do** *Body* **end** where *Cond* is a boolean  $\text{FO}^*$  formula and *Body* is a sequence of instructions. The fact that the right part  $\varphi$  of an assignment is a  $\text{FO}^*$  formula entails that the auxiliary schema  $S$  may have multiple attributes of type "time". Of course in this framework, there is no such things as shared or private auxiliary schemas and instructions do not include temporal moves.

### 3 Relational temporal machines

In this section, we introduce *relational temporal machine* (RTM) as an abstract model for general temporal db querying. Recall that here we intend to capture temporal queries where both the input and the output are temporal db instances as explained in the introduction. The model is a Turing-like machine which merges ideas from relational machines (2) and automata (13). The input tape contains the input temporal instance and is read only. A finite number of working tapes are available for storing intermediate computation and the output. Of course, for both input and working tapes, each cell of the tape is a relational instance. A relational db schema is associated to each tape for typing the contents of the cells. Over each tape, a cursor is used to scan the corresponding temporal instance and gives access to the contents of a cell, i.e. a relational instance. A cursor is allowed to move in both directions (left and right). A finite number of registers is introduced in order to store time points over the input tape; the only operation on registers (apart from assignment) is a comparison with the input cursor. As usual, transitions specify the changes of state, on registers and tape contents as well as the moves of the cursors. Transition may use a query in order to modify the contents of a cell (pointed by a cursor). Queries are  $\text{FO}$  formulas specified over the input and auxiliary schemas. They are evaluated on the relational instance composed by the contents of the cells pointed by the input and auxiliary cursors.

Next, we assume that  $\mathcal{R}$  is the input schema and that the input tape stores an instance  $\mathcal{I}$  over  $\mathcal{R}$ . Formally, over  $\mathcal{R}$ , an  $\text{RTM}_k^m \mathcal{M}$  is specified by<sup>2</sup>:

- a finite number  $m$  of db auxiliary schemas  $\mathcal{S}_1 \dots \mathcal{S}_m$  pairwise disjoint, one for each of the  $m$  working tapes; an output tape given by  $out \in [1..m]$ ,
- a finite set of states  $\mathcal{E}$  among which *start* is the initial state; a subset  $\mathcal{E}_f$  of final states,
- a finite set  $\mathcal{T}$  of transitions of the form  $(q, P, s) \rightarrow (ns, a)$  where:
  - $q$  is a boolean  $\text{FO}$  query over  $\mathcal{R} \cup_{i=1..m} \mathcal{S}_i$ ,
  - $P \subseteq [1..k]$  is a subset of the  $k$  register indices,
  - $s$  and  $ns$  are states,
  - $a$  is an action among *left*( $i$ ), *right*( $i$ ), *erase*( $i$ ), *create*( $i$ ), where  $i \in [0..m]$ , *update*( $S, \varphi$ ) with  $S \in \mathcal{S}_i$ ,  $i \in [1..m]$  and  $\varphi$  a  $\text{FO}$  query over  $\mathcal{R} \cup_{i=1..m} \mathcal{S}_i$ , and finally *reg*( $j$ ) with  $j \in [1..k]$ .

A configuration of an  $\text{RTM}_k^m \mathcal{M}$  on input  $\mathcal{I}$  is specified by a tuple  $[[c_0, c_1 \dots c_m], s, \theta, [\mathcal{I}_1 \dots \mathcal{I}_m]]$  where  $c_i$  are the input and auxiliary cursor values,  $\theta$  is a register assignment

<sup>2</sup> $m$  stands for the number of tapes and  $k$  for the number of registers.

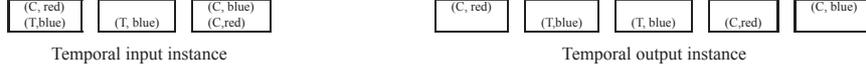


Figure 1: An example of a color based splitting transformation

and  $\mathcal{I}_i$  is a temporal instance over  $\mathcal{S}_i$ . It is required that  $c_0 \in [0..|\mathcal{I}|]$  and  $c_i \in [0..|\mathcal{I}_i|]$ , for  $i \in [1..m]$ . The initial configuration is  $[[0..0], start, \theta_0, [\emptyset_{\mathcal{S}_1}^0 \dots \emptyset_{\mathcal{S}_m}^0]]$  where  $\theta_0(j)=0$ ,  $\forall j \in [1..k]$ . A configuration whose state  $s$  is in  $\mathcal{E}_f$  is accepting and then the  $\text{RTM}_k^m \mathcal{M}$  outputs the temporal instance  $\mathcal{I}_{out}$ .

Given a configuration  $[[c_0, c_1 \dots c_m], s, \theta, [\mathcal{I}_1 \dots \mathcal{I}_m]]$ , the transition  $(q, P, s) \rightarrow (ns, a)$  applies iff

- $P = \{l \mid l \in [1..k], \theta(l) = c_0\}$ , and
- $q(J) = true$  where  $J$  is the relational instance over  $\mathcal{R} \cup_{i=1..m} \mathcal{S}_i$  defined by  $J[\mathcal{R}] = \mathcal{I}[c_0]$  and  $J[\mathcal{S}_i] = \mathcal{I}_i[c_i]$ .

It leads to the new configuration  $[[c'_0, c'_1 \dots c'_m], s, \theta_n, [\mathcal{J}_1 \dots \mathcal{J}_m]]$  where (we only mention changes):

- if  $a$  is *right*( $i$ ) (resp. *left*( $i$ )) then<sup>3</sup> if  $|\mathcal{I}_i| \neq 0$  then  $c'_i = \max\{c_i + 1, |\mathcal{I}_i|\}$  (resp.  $c'_i = \min\{c_i - 1, |\mathcal{I}_i|\}$ ),
- if  $a$  is *erase*( $i$ ) then  $\mathcal{J}_i$  is obtained by removing the last cell of the tape  $\mathcal{I}_i$ ; as side effects, if  $c_i = |\mathcal{I}_i|$  then  $c'_i = c_i - 1 = |\mathcal{J}_i|$ , and if  $\theta(j) = |\mathcal{I}_i|$  then  $\theta_n(j) = \theta(j) - 1 = |\mathcal{J}_i|$ ,
- if  $a$  is *create*( $i$ ) then  $\mathcal{J}_i$  is obtained by extending the tape  $\mathcal{I}_i$  with a new empty cell,
- if  $a$  is *update*( $S, \varphi$ ) with  $S \in \mathcal{S}_i$ , if  $c_i \neq 0$ , then  $\mathcal{J}_i[c_i](S) = \varphi(J)$ , where  $J$  is defined as above (nothing else changes),
- if  $a$  is *reg*( $j$ ) then  $\theta_n(j) = c_0$ ,

Before we start the analysis of our model, let us illustrate the definition of  $\text{RTM}$  by presenting an example. Assume that the temporal database stores a sequence of picture descriptors. Each state is a picture and each picture contains circles and triangles which can be either blue or red. The  $\text{RTM}$  given in Figure 2 takes as input a sequence of pictures and split each picture in order to separate red forms from blue ones: red forms are placed in a picture that precedes the blue ones which are placed in another picture. Thus, in general, the result of this "color based splitting" is a temporal instance whose size is greater or equal to the input size. Figure 1 gives an input sequence of pictures and the expected output. Figure 2 represents the  $\text{RTM}$  implementing the color based splitting.

A transaction  $(q, P, s) \rightarrow (ns, a)$  is drawn like this ( $P$  is omitted because the example does not require it):

$\textcircled{s} \xrightarrow{q / \_} \textcircled{ns}$ . When the boolean query  $q$  is *true*, the arrow is labelled by  $\_ / a$ , and when there no action, it is labelled by  $q / \_$ .

**Determinism:** Next, we will concentrate on deterministic  $\text{RTM}_k^m$ , ensuring that in each configuration at most one transition applies. The source of non determinism of an  $\text{RTM}$

<sup>3</sup>for the case  $i=0$ , one should take  $\mathcal{I}_0 = \mathcal{I}$

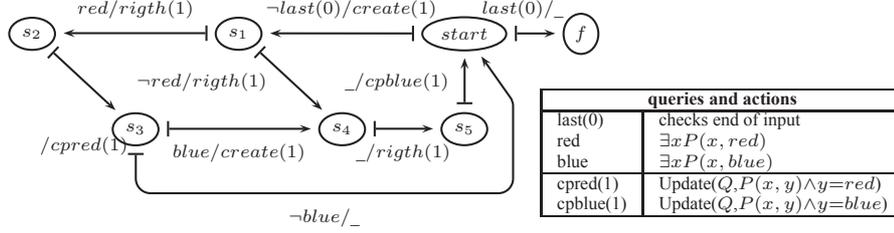


Figure 2: The RTM for color based splitting.

$\mathcal{M}$  comes from the boolean queries  $q$  in a transition  $(q, P, s) \rightarrow (ns, a)$ . One way to enforce determinism is to require that given a state  $s$  the set of transitions contains at most two kinds of transitions, the ones with premises of the form  $(q, P, s)$  and the others with premises of the form  $(\neg q, P, s)$ . In other words,  $s$  determines  $q$ . Next,  $\text{RTM}_k^m$  denotes the class of temporal queries computable by deterministic  $\text{RTM}_k^m$ .

**Registers** do not add any expressiveness to our model. They are introduced in order to simplify the specification of RTMs and also because they are useful (in a strong sense) for restricted version of our model (see section 4).

**Auxiliary tapes:** The second part of the next result says that it is sufficient to consider RTM with only two auxiliary tapes and moreover that one of these two tapes is a store. A store is a one-cell tape. Such devices are called *extended one-tape RTMs* and denoted by  $\text{RTM}_k^{1+}$ .

**PROPOSITION 3.1**  $\text{RTM}_k^m = \text{RTM}_0^{m+1}$  and  $\text{RTM}_k^m = \text{RTM}_k^{1+}$ .

The reader may have noticed that the temporal auxiliary tape (resp. the store) of an extended one-tape RTM plays a similar role as the private (resp. shared) auxiliary schemas of a T-WHILE program. This remark will serve to study complexity issues.

The next result is stronger and may seem surprising at a first glance although it turns out to be a generalization of the normalisation of RM developed in (2). It tells us that, indeed, any  $\text{RTM}_k^m \mathcal{M}$  can be simulated by an  $\text{RTM}_k^1$ .

**THEOREM 3.2**  $\text{RTM}_k^m = \text{RTM}_k^1$  and thus  $\text{RTM}_k^m = \text{RTM}_1^0$

The normalization of RTM is an interesting result by itself. It is a witness, once more, that the technics provided in (2) are quite powerful. However, one-tape RTMs have a drawback: they blur the distinction between the cost, in space, of the computation in terms of the active domain (more precisely in term of the equivalence classes) and the cost, in space, of the computation in terms of the size (number of states) of the input. This is the reason why, in the next section complexity issues are discussed based on the extended one-tape devices rather than on the one-tape machines.

## 4 Space complexity: RTM versus temporal languages

Now, we are going to investigate the connection, in terms of complexity and expressiveness, between our general temporal computation model RTM and some languages, namely T-WHILE and TS-WHILE, as well as some restriction of these languages. As already mentioned, we focus on space complexity and the main issue addressed in this section is to measure temporal computation complexity in terms of the auxiliary space required. Recall that the size of a temporal instance is the length of its sequence of database states. We are not interested here in the size of the active domain as usually assumed.

Next, we only consider extended one-tape RTMs, the ones using one temporal tape plus a store. The set of RTMs whose runs make use of an auxiliary temporal tape of size linear (resp. polynomial) in the size of the input temporal instance is denoted LinSpace-RTM (resp. PSpace-RTM).

The first result of this section establishes a strong correspondence between LinSpace-RTMs and T-WHILE queries in the one hand, and between PSpace-RTMs and TS-WHILE queries in the other hand. The reader should be aware that this result holds iff the output tape of the extended one-tape RTM is restricted to be the store.

**THEOREM 4.1** *The following holds (under the above restriction):*

- LinSpace-RTM = T-WHILE, and
- PSpace-RTM = TS-WHILE.

We focus now on RTM whose space complexity is constant. Indeed constant space RTMs are the devices using one auxiliary store and no auxiliary temporal tape. Such devices are next called relational register machine ( $\text{RRM}_k$ ), because registers are necessary components in that case<sup>4</sup>. We also investigate relational pebble machine  $\text{RPM}_k$  which are straightforward restrictions of  $\text{RRM}_k$  that forces a stack discipline over the  $k$  registers.

**PROPOSITION 4.2** *Propositional RPM expresses exactly the regular languages over finite alphabet and thus  $\text{RPM} \subset \text{RRM}$ .*

We are now interested by languages that express queries RPM (resp RRM) computable. Below, the language T-WHILE<sub>shared</sub> is the restriction of T-WHILE to the case where only shared auxiliary relations can be used by the programs.

**THEOREM 4.3** • TS-FO  $\not\subseteq$  RPM, • T-WHILE<sub>shared</sub>  $\subseteq$  RPM (*claim: the inclusion is strict*)

Recall that the languages TS-FO and T-WHILE<sub>shared</sub> are not comparable: TS-FO can express the query  $\forall\text{-twin}$  ( $\forall t \exists t' [t \neq t' \wedge \forall x (R(x, t) \leftrightarrow R(x, t'))]$ ) but T-WHILE<sub>shared</sub> cannot; T-WHILE<sub>shared</sub> can express transitive closures which cannot be expressed with TS-FO.

---

<sup>4</sup>Proposition 3.1 stating that registers do not add expressiveness to the model, does not apply when considering  $\text{RRM}_k$  because registers can no more be simulated using a temporal tape

The last part of this section is devoted to the presentation of the language  $\text{T-WHILE}_{\text{shared}}^{\text{bind}}$  which is complete with respect to  $\text{RRM}$ . The language  $\text{T-WHILE}_{\text{shared}}^{\text{bind}}$  is an extension of  $\text{T-WHILE}_{\text{shared}}$  with temporal variables (t-variables) which are obviously meant to mimic the registers of the  $\text{RRM}$  devices. Thus besides shared auxiliary schemas, the declaration of a  $\text{T-WHILE}_{\text{shared}}^{\text{bind}}$  program may include a finite list of t-variables  $t_1 \dots t_k$  which are all initialized to 1. The formulas  $\varphi$  in the right part of assignments  $S : = \varphi$  belong to  $\text{FO}^{\text{bind}}$ . The language  $\text{FO}^{\text{bind}}$  over  $\mathcal{R} \cup \mathcal{S}$  and  $t_1 \dots t_k$  allows one for new atomic formulas of the form  $t_i = t_j$  but restrict quantification to apply over "data" variables only. Finally, instructions of the form  $\downarrow t_i$  are added in order to bind t-variables to the current time point. The semantics of a  $\text{T-WHILE}_{\text{shared}}^{\text{bind}}$  program directly follows from that of  $\text{T-WHILE}_{\text{shared}}$ . While loop execution and termination are based on configurations: a configuration is now a triple  $(\mathcal{J}, \text{ctp}, g)$  where  $g$  is a t-variable assignement.

**EXAMPLE 4.1** The following piece of program places the current time point  $\text{ctp}$  at the time point contained in the t-variable  $t$ . It uses two t-variables  $t$  and  $t'$ .

**while** not(first) **do** left **end**;  $\downarrow t'$ ; **while** not ( $t' = t$ ) **do** ( right;  $\downarrow t'$  ) **end**.

The next results states that the language  $\text{T-WHILE}_{\text{shared}}^{\text{bind}}$  is complete wrt  $\text{RRM}$ .

**THEOREM 4.4**  $\text{T-WHILE}_{\text{shared}}^{\text{bind}} = \text{RRM}$ .

In (5) we provide a second language, called  $\text{@TL}$ , which is equivalent to  $\text{T-WHILE}_{\text{shared}}^{\text{bind}}$ . Its definition is based on  $\text{TL}$  and borrows some idea from hybrid modal logic (4).

## 5 Discussion

In this paper, we have introduced a general computation model for temporal database queries. This model is the first that captures queries whose output is a temporal instance. The results of the paper show that  $\text{RTM}$  provides an interesting formal tool for analysing standard languages and we claim that this tool opens several further research directions.

The  $\text{RTM}$  computation model provides the basis for identifying some critical features of temporal computation which are often implicit when just considering concrete languages. Investigating the features/parameters of  $\text{RTM}$  seem a promising research direction and may well benefit to application domains like streaming. Continuous-query processing over data streams (3) is a very challenging research topic because of its large range of applications, from stock ticks to sensor applications. Formalizing continuous query and streamable queries (12; 11) is under study but a general model still needs to be proposed. For instance, none of the proposed formal models is yet able to take into account manipulations over multiple streams like fusion or in general operations that output a stream from streams which is a critical issue in the framework of distributed sensors applications.

Another important issue is the definition of concrete temporal languages satisfying reasonable complexity criteria. In (7), we propose a class of languages called  $\text{SCTL}$ . These languages are based on a paradigm related to windows in the context of streaming. They are based on a two phase process: first, the input temporal instance is sliced

to produce a sequence of subsequences of the input (a sequence of slices); slices may overlap or not, they may cover or not the input; secondly, temporal queries are evaluated over each slice although with access to the whole input. An open and exiting issue is the definition of a P-Space temporal language a la T-WHILE.

## References

- [1] ABITEBOUL S., HERR L. & DEN BUSSCHE J. V. (1995). Temporal connectives versus explicit timestamps in temporal query languages. In *Proc. of the VLDB International Workshop on Temporal Databases*, p. 43–57.
- [2] ABITEBOUL S. & VIANU V. (1995). Computing with first-order logic. In *Selected papers of the 23rd annual ACM symposium on Theory of computing*, p. 309–335.
- [3] ARASU A., BABCOCK B., BABU S., MCALISTER J. & WIDOM J. (2004). Characterizing memory requirements for queries over continuous data streams. *ACM Trans. Database Syst.*, **29**(1), 162–194.
- [4] ARECES C., BLACKBURN P. & MARX M. (2001). Hybrid logics: Characterization, interpolation and complexity. *J. of Symbolic Logic*, **66**(3), 977–1010.
- [5] BIDOIT N. & HANTRY F. (2007). Relational temporal machine. In *Proceedings of the 14th International Symposium on Temporal Representation and Reasoning*.
- [6] BIDOIT N. & OBJOIS M. (2005). Temporal query languages expressive power:  $\mu$ tl versus t-while. In *Proceedings of the 12th International Symposium on Temporal Representation and Reasoning*, p. 74–82.
- [7] BIDOIT N. & OBJOIS M. (2007). Stql: a preliminary proposal for t2t languages. In *Proceedings of the 14th International Symposium on Temporal Representation and Reasoning*.
- [8] CHANDRA A. K. & HAREL D. (1980). Structure and complexity of relational queries. In *IEEE Symposium on Foundations of Computer Science*, p. 333–347.
- [9] CHOMICKI J. & TOMAN D. (1998). Temporal logic in information systems. In *Logics for Databases and Information Systems*, p. 31–70.
- [10] EMERSON E. A. (1990). Temporal and modal logic. p. 995–1072.
- [11] GROHE M., GUREVICH Y., LEINDERS D., SCHWEIKARDT N., TYSZKIEWICZ J. & DEN BUSSCHE J. V. (2007). Database query processing using finite cursor machines. In *ICDT*, p. 284–298.
- [12] GROHE M., KOCH C. & SCHWEIKARDT N. (2005). Tight lower bounds for query processing on streaming and external memory data. In *ICALP*, p. 1076–1088.
- [13] THOMAS W. (1997). Languages, automata, and logic.  
[timecenter] TIMECENTER.
- [14] VARDI M. Y. (1988). A temporal fixpoint calculus. In *POPL '88: Proceedings of the 15th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, p. 250–259.
- [15] WOLPER P. (1983). Temporal logic can be more expressive. *Information and Control*, **56**(1/2), 72–99.

# Éligibilité de contraintes pour la résolution de réseaux de contraintes qualitatives temporelles et spatiales \*

Jean-François Condotta<sup>1</sup>, Gérard Ligozat<sup>2</sup> et Mahmoud Saade<sup>1</sup>

<sup>1</sup> CRIL-CNRS, Université d'Artois, 62307 Lens Cedex, France

<sup>2</sup> LIMSI-CNRS, Université de Paris-Sud, 91403 Orsay, France  
{condotta, saade}@cril.univ-artois.fr,ligozat@limsi.fr

**Résumé** : Dans cet article, nous considérons le problème de la cohérence des réseaux de contraintes qualitatives pour le temps et l'espace. La méthode la plus efficace pour résoudre ce problème consiste à utiliser un algorithme de recherche mettant en œuvre d'une part la méthode de fermeture par faible composition comme propagation locale de contraintes et d'autre part une décomposition des contraintes en relations appartenant à une classe traitable. Nous proposons une extension de cet algorithme basée sur la notion d'éligibilité de contraintes. Nous étudions cet algorithme de manière théorique et de manière expérimentale.

## 1 Introduction

De nombreux formalismes qualitatifs ont été développés ces vingt dernières années pour représenter les configurations temporelles ou spatiales et pour raisonner sur ces dernières. La représentation et le raisonnement sur les informations temporelles et spatiales est une tâche importante de nombreuses applications telles que les systèmes d'informations géographiques (GIS), la compréhension du langage naturel, la navigation de robots, la planification temporelle et spatiale. Le raisonnement temporel et spatial qualitatif a pour but de décrire des relations non numériques entre les entités temporelles ou spatiales. Habituellement un calcul qualitatif [1, 16, 9, 15, 6] utilise des éléments particuliers (sous-ensembles d'un espace topologique, points de la droite des rationnels, intervalles de la droite des rationnels) pour représenter les entités spatiales ou temporelles du système, et se focalise sur un nombre limité de relations entre ces éléments (telles que des relations topologiques entre les régions ou des relations de précédence entre les points). Chacune de ces relations représente une situation temporelle ou spatiale particulière. Par exemple, considérons le formalisme temporel d'Allen bien connu appelé algèbre des intervalles [1]. Il utilise des intervalles de la droite des rationnels pour représenter les entités temporelles. Treize relations de base entre ces intervalles sont utilisées pour modéliser les différentes situations qualitatives possibles entre les entités temporelles. Par exemple, la relation de base *overlaps* peut être utilisée pour représenter la situation où une première activité débute avant une deuxième activité et se

---

\*Ce travail a bénéficié du financement de la région Nord-Pas-de-Calais, du FEDER.

termine pendant que la deuxième se déroule encore.

Les informations temporelles ou spatiales concernant la situation d'un ensemble particulier d'entités peuvent être représentées en utilisant des réseaux de contraintes particuliers appelés réseaux de contraintes qualitatives (qualitative constraint networks en anglais, RCQ en abrégé). Chaque contrainte d'un RCQ représente un ensemble de configurations qualitatives acceptables entre les entités temporelles ou spatiales et est définie par un ensemble de relations de base. Le problème de la cohérence pour les RCQ consiste à déterminer si un réseau donné possède des solutions satisfaisant ces contraintes. Pour résoudre ce problème, des méthodes basées sur des algorithmes de propagation locale de contraintes ont été définies [13], en particulier des méthodes de fermeture par faible composition, méthodes similaires à celles utilisées pour obtenir la propriété de chemin-cohérence dans le cadre des CSP [12, 11]. Les algorithmes les plus performants utilisent également une décomposition des contraintes en relations appartenant à une classe traitable comme l'ensemble des relations préconvexes dans le cadre du formalisme d'Allen. Chaque contrainte est découpée en relations d'une classe de relations pour laquelle la méthode de fermeture par faible composition est complète. Elle est successivement instanciée par une de ces relations jusqu'à ce qu'une solution soit trouvée ou que toute solution partielle explorée conduise à un échec. Le facteur de branchement lors de la recherche est diminué grâce au découpage des contraintes.

Dans cet article nous étendons cet algorithme en introduisant une notion d'éligibilité de contraintes. L'amélioration proposée consiste à détecter au plus tôt les contraintes qui n'auront pas à être considérées lors de la recherche, à savoir celles définies à un moment donné par une relation appartenant à l'ensemble de relations utilisé pour le découpage des contraintes.

Cet article est organisé de la manière suivante. Dans la section 2, nous faisons des rappels sur les formalismes qualitatifs. La section 3 est dévolue à la définition et à l'étude théorique de notre algorithme appelé *cohérenceE*. La section 4 est consacrée aux résultats expérimentaux obtenus en utilisant cet algorithme. Nous concluons dans la section 5.

## 2 Rappels sur les formalismes qualitatifs pour le temps et l'espace

Dans la suite, nous supposons donné un formalisme qualitatif pour le temps ou l'espace défini sur un ensemble fini  $B$  de relations de base sur un domaine  $D$ . Pour des raisons de lisibilité, nous supposons que l'arité de ces relations est 2. Nous faisons également l'hypothèse que les relations de base constituent une partition de  $D \times D$ , c'est-à-dire que tout couple d'éléments de  $D$  appartient à exactement une relation de  $B$ . De plus, nous supposons que  $B$  contient la relation identité sur  $D$ , dénoté par  $Id$  dans la suite. À titre d'illustration, considérons le formalisme d'Allen : l'algèbre des intervalles  $AI$ .  $AI$  est basée sur 13 relations binaires définies sur un ensemble d'intervalles, celui de la droite des nombres rationnels par exemple. Chacune de ces relations correspond à un ordre particulier des 4 bornes de deux intervalles.

Les informations temporelles ou spatiales sur la configuration d'un ensemble d'entités peuvent être représentées à l'aide d'un réseau de contraintes dit réseau de contraintes

qualitatives (RCQ en abrégé). Ces réseaux permettent de représenter des informations incomplètes. En effet, la contrainte entre deux variables (correspondant à deux entités) est définie par un sous-ensemble de l'ensemble des relations de base, l'ensemble des relations de base qui sont compatibles avec les informations disponibles.

Formellement, un réseau de contraintes qualitatives (un RCQ en abrégé)  $\mathcal{N}$  est un couple  $(V, C)$  où  $V$  est un ensemble fini de  $n$  variables  $v_0, \dots, v_{n-1}$  (avec  $n$  entier positif) et  $C$  est une application qui, à chaque couple  $(v_i, v_j)$  de variables de  $V$  associe un sous-ensemble  $C(v_i, v_j)$  de l'ensemble des relations de base :  $C(v_i, v_j) \subseteq B$ . Nous dénoterons par  $A$  l'ensemble  $2^B$  de tous les ensembles de  $B$ . Pour  $r \in A$ , deux éléments  $x, y \in D$  satisfont  $r$ , ce que nous dénoterons par  $x r y$ , ssi il existe une relation de base  $a \in r$  telle que  $(x, y) \in a$ . Ainsi chaque élément  $r$  de  $A$  peut être considéré comme l'union de toutes les relations de base la composant. Nous utilisons le terme de "relation" pour désigner une telle union de relations de base. L'ensemble  $A$  est muni de l'opération d'intersection ( $\cap$ ) et l'opération d'union ( $\cup$ ). Il est également muni de l'opération unaire inverse ( $\sim$ ) et de l'opération binaire de faible composition ( $\circ$ ) : l'inverse d'une relation  $r$  de  $A$  est la relation de  $A$  correspondant à la transposée de  $r$  ; elle correspond à l'union des inverses des relations de base appartenant à  $r$ . La faible composition  $a \circ b$  de  $a, b \in B$  est la relation  $r$  définie par  $r = \{c : \exists x, y, z \in D, x a y, y b z \text{ et } x c z\}$ . La composition  $r \circ s$  de  $r, s \in A$  est la relation  $t = \bigcup_{a \in r, b \in s} \{a \circ b\}$ . Intuitivement,  $r \circ s$  est l'ensemble de toutes les relations de base possibles entre  $x \in D$  et  $y \in D$  lorsqu'il existe  $z \in D$  avec  $x r z$  et  $z s y$ . Remarquons que pour certains formalismes (voir [2] par exemple), cette opération de faible composition n'est pas équivalente à l'opération de composition relationnelle classique.

Concernant les RCQ nous utiliserons les définitions suivantes dans la suite :

### Définition 1

Soit  $\mathcal{N} = (V, C)$  un RCQ, avec  $V = \{v_0, \dots, v_{n-1}\}$ . Une solution partielle de  $\mathcal{N}$  sur  $V' \subseteq V$  est une application  $\sigma$  de  $V'$  dans  $D$  telle que  $\sigma(v_i) C(v_i, v_j) \sigma(v_j)$ , pour tout  $v_i, v_j \in V'$ . Une solution de  $\mathcal{N}$  est une solution partielle de  $V$ .  $\mathcal{N}$  est cohérent si et seulement si il admet une solution.  $\mathcal{N}$  est  $\circ$ -fermé<sup>1</sup> si et seulement si pour tout  $v_k, v_i, v_j \in V$ ,  $C(v_i, v_j) \subseteq C(v_i, v_k) \circ C(v_k, v_j)$ . Un sous-RCQ  $\mathcal{N}'$  de  $\mathcal{N}$  est un RCQ  $(V, C')$  où  $C'(v_i, v_j) \subseteq C(v_i, v_j)$  pour tout  $v_i, v_j \in V$ . On note  $\mathcal{N}' \subseteq \mathcal{N}$  le fait que  $\mathcal{N}'$  soit un sous-RCQ de  $\mathcal{N}$ . Un scénario de  $\mathcal{N}$  est un sous-RCQ  $(V, C')$  de  $\mathcal{N}$  tel que  $C'(v_i, v_j) = \{a\}$  avec  $a \in B$ . Un scénario cohérent de  $\mathcal{N}$  est un scénario de  $\mathcal{N}$  admettant une solution.  $\mathcal{N}' = (V', C')$  est équivalent à  $\mathcal{N}$  si et seulement si  $V = V'$  et si les deux RCQs ont les mêmes solutions.

Étant donné un RCQ  $\mathcal{N} = (V, C)$ , des algorithmes de propagation locale de contraintes [12, 11, 3, 4] peuvent être utilisés pour dériver en temps polynomial (en  $O(|V|^3)$  pour certains d'entre eux) un sous-RCQ qui est  $\circ$ -fermé et équivalent à  $\mathcal{N}$ . En bref, le principe de ces algorithmes est d'itérer l'opération suivante  $C(v_i, v_j) \leftarrow C(v_i, v_j) \cap (C(v_i, v_k) \circ C(v_k, v_j))$  pour tout triplet de variables  $v_i, v_j, v_k \in V$  jusqu'à qu'un point fixe soit obtenu. Dans la suite, l'expression "méthode de la fermeture par faible composition" désignera un tel algorithme. Le RCQ  $\circ$ -fermé obtenu par application de la méthode de

<sup>1</sup>Le terme "chemin-cohérent" peut souvent être trouvé dans la littérature. Nous évitons ce terme du fait qu'un RCQ  $\circ$ -fermé n'est pas nécessairement chemin-cohérent au sens classique utilisé pour les CSP.

la  $\circ$ -fermeture sur un RCQ  $\mathcal{N}$  sera dénoté par  $ffc(\mathcal{N})$  et appelé  $\circ$ -fermeture de  $\mathcal{N}$ . La  $\circ$ -fermeture satisfait les propriétés suivantes.

**Proposition 1**

Soient  $\mathcal{N}$  et  $\mathcal{N}'$  deux RCQ : (a)  $ffc(\mathcal{N}) \subseteq \mathcal{N}$ , (b)  $ffc(ffc(\mathcal{N})) = ffc(\mathcal{N})$ , (c) si  $\mathcal{N} \subseteq \mathcal{N}'$  alors  $ffc(\mathcal{N}) \subseteq ffc(\mathcal{N}')$ .

### 3 L'algorithme de recherche avec éligibilité

Une méthode brutale pour tester la cohérence d'un réseau de contraintes qualitatif  $\mathcal{N}$  consiste à instancier successivement chaque contrainte du réseau par chacune de ses relations de base jusqu'à trouver un scénario cohérent. Après chaque instanciation la méthode de la fermeture par faible composition est appliquée pour supprimer des relations de base non possibles et pour tester de manière grossière la cohérence de la solution partielle. En cas d'échec un retour en arrière est opéré. L'algorithme s'arrête lorsqu'un scénario  $\circ$ -fermé est obtenu ou lorsque toutes les instanciations partielles ont conduit à une incohérence. Dans le premier cas, pourvu que la méthode de la fermeture de la faible composition soit complète pour les scénarios, le scénario obtenu est une solution.

Cet algorithme de base à été rendu plus efficace à l'aide de l'utilisation de classes dites traitables [7, 13, 17]. L'idée générale est de disposer d'un ensemble *Split* de relations (contenant l'ensemble des relations de base) pour lequel la méthode de la fermeture par faible composition est complète. Lors de la recherche, au lieu d'instancier la contrainte courante par une de ses relations de base, cette contrainte est partagée en relations appartenant à l'ensemble *Split* et la contrainte est ensuite successivement instanciée par ces relations. De manière générale le facteur de branchement est diminué et l'espace de recherche plus restreint que celui de la méthode de base. Lorsque l'algorithme se termine sur un succès, nous obtenons un sous-réseau du réseau initial qui est  $\circ$ -fermé et dont toutes les relations appartiennent à *Split*. Ce réseau est donc forcément cohérent et par conséquent le réseau de contraintes initial l'est également.

Dans cette section nous proposons un raffinement de cet algorithme basé sur la notion d'*éligibilité*. Le principe de ce nouvel algorithme est simple, il consiste en la détection au plus tôt de contraintes qui n'auront pas besoin d'être instanciées pour se prononcer sur le résultat de la cohérence ou non du réseau initial. Pour cela, à chaque étape de la recherche, en particulier lors de la réalisation de la fermeture par faible composition, nous marquons comme non éligible toute contrainte définie par une relation appartenant à l'ensemble *Split*. De plus, le choix de la nouvelle contrainte à instancier se fait uniquement sur les contraintes dites éligibles. L'objectif général est d'instancier un nombre minimal de contraintes. Dans la suite nous montrerons que cet algorithme est correct et complet. Dans Algorithme 1 est donnée la définition de notre algorithme appelé *cohérenceE* (*E* pour éligible) dans la suite. La méthode prend en paramètre un RCQ  $\mathcal{N}$  et retourne une valeur booléenne correspondant à la détection ou non de la cohérence de  $\mathcal{N}$ . Un ensemble  $Split \subseteq A$  de relations est utilisé pour le découpage des relations lors des instanciations. *Split* doit contenir les relations composées d'une seule relation de base et doit être fermé pour l'opération inverse. Dans le cadre de l'al-

gèbre des intervalles *Split* peut par exemple être l'ensemble des relations préconvexes. La première étape de l'algorithme *cohérenceE* consiste à marquer comme éligible chaque contrainte de  $\mathcal{N}$  n'appartenant pas à l'ensemble *Split* et à marquer comme non éligible chacune des autres contraintes. Ensuite est retourné le résultat de l'appel de la fonction récursive *cohérenceE'*. Cette fonction met en œuvre le mécanisme expliqué auparavant. Chacun de ces appels débute par un appel de la méthode de la fermeture par faible composition ; en plus du travail classique de triangulation, cette dernière détecte les nouvelles contraintes dont les relations appartiennent à *Split* et devant être marquées comme non éligibles à ce stade du traitement. À cette étape il se peut que  $\mathcal{N}$  soit détecté incohérent. L'étape suivante consiste à sélectionner une contrainte éligible et à la traiter. Elle est tout d'abord partagée en relations appartenant à l'ensemble *Split*. Chacune de ces relations est utilisée pour instancier la contrainte sélectionnée. La recherche se poursuit en appelant de nouveau *cohérenceE'*. La recherche s'arrête lorsque toutes les contraintes sont marquées comme éligibles ou lorsque toutes les instanciations partielles se sont soldées par un échec.

#### Algorithme 1

**Function** *cohérenceE*( $\mathcal{N}$ ), avec  $\mathcal{N} = (V, C)$  et  $n = |V|$ .

```

1: for  $i, j \in 0 \dots n - 1$ 
2:   if ( $C_{ij} \in Split$ ) Marquer non éligible  $C_{ij}$  else Marquer éligible  $C_{ij}$ 
3: endFor
4: return cohérenceE'( $\mathcal{N}$ )

```

**Function** *cohérenceE'*( $\mathcal{N}$ ), with  $\mathcal{N} = (V, C)$ .

```

1: if (ffc( $\mathcal{N}$ ) == false) return false
2: Sélectionner une contrainte  $C_{ij}$  telle que  $C_{ij}$  est éligible
3: if une telle contrainte  $C_{ij}$  n'existe pas return true
4: Partage  $C_{ij}$  en sous relations  $r_1, \dots, r_k$  telles que  $r_l \in Split$ 
5: for  $l \in 1 \dots k$  do
6:    $C_{ij} \leftarrow r_l, C_{ji} \leftarrow C_{ij}^{-1}$ 
7:   Marquer non éligible  $C_{ij}$  et  $C_{ji}$ 
8:   if (cohérenceE'( $\mathcal{N}$ ) == true) return true
9: endFor
10: return false

```

**Function** *ffc*( $\mathcal{N}$ ), avec  $\mathcal{N} = (V, C)$  et  $n = |V|$ .

```

1: do
2:    $\mathcal{N}' \leftarrow \mathcal{N}$ 
3:   for  $i, j, k \in 0 \dots n - 1$ 
4:      $C_{ij} \leftarrow C_{ij} \cap (C_{ik} \circ C_{kj}), C_{ji} \leftarrow C_{ij}^{-1}$ 
5:     if ( $C_{ij} == \emptyset$ ) return false
6:     if ( $C_{ij} \in Split$ ) Marquer non éligible  $C_{ij}$  et  $C_{ji}$ 
7:   endFor
8: until ( $\mathcal{N}' == \mathcal{N}$ )
9: return true

```

Clairement, si l'algorithme se termine en retournant **false** c'est que toutes les instanciations partielles ont été détectées incohérentes à l'aide de la méthode de fermeture par faible composition. Dans le cas contraire il n'est pas évident que le RCQ soit cohérent. En suivant la ligne de raisonnement proposée par Nebel [13] nous allons montrer que

le RCQ  $\mathcal{N}$  obtenu est un RCQ  $\circ$ -fermé dont toutes les relations appartiennent à l'ensemble *Split*. Ainsi, si *Split* est complet pour la méthode de la fermeture par faible composition, le réseau obtenu est forcément cohérent. Tout d'abord nous introduisons une terminologie dont une partie est similaire à celle utilisée dans [13].

Étant donné un RCQ  $\mathcal{N} = (V, C)$ , deux entiers  $0 \leq i, j < |V|$  avec  $n$  le nombre de variables de  $\mathcal{N}$  et une relation  $r \in 2^B$ ,  $\mathcal{N}[(i, j, r)]$  représente le RCQ  $(V, C')$  définie par  $C'_{kl} = r$  (resp. la relation  $r^{-1}$ ) dans la cas où  $k = i$  et  $l = j$  (resp. dans la cas où  $k = j$  et  $l = i$ ),  $C'_{kl} = C_{kl}$  sinon. Étant donné un ensemble  $\Psi \subseteq \{0, \dots, n-1\} \times \{0, \dots, n-1\} \times 2^B$  avec  $i < j$  pour tout  $(i, j, r) \in \Psi$ , nous définissons  $\mathcal{N}[\Psi]$  par  $\mathcal{N}[(i_0, j_0, r_0)] \dots [(i_k, j_k, r_k)]$  avec  $\Psi = \{(i_0, j_0, r_0), \dots, (i_k, j_k, r_k)\}$ . Cet opérateur de substitution satisfait les propriétés suivantes :

**Proposition 2**

Soient  $\mathcal{N} = (V, C)$ ,  $\mathcal{N}' = (V, C')$  deux RCQ, un triplet  $(i, j, r)$  avec  $0 \leq i, j < n$  et  $r \in 2^B$  et un ensemble  $\Psi \subseteq \{0, \dots, n-1\} \times \{0, \dots, n-1\} \times 2^B$  avec  $n = |V|$ .

- (a) Si  $r \subseteq C_{ij}$  alors  $\mathcal{N}[(i, j, r)] \subseteq \mathcal{N}$  sinon  $\mathcal{N} \subseteq \mathcal{N}[(i, j, r)]$ .
- (b) Si  $\mathcal{N} \subseteq \mathcal{N}'$  alors  $\mathcal{N}[\Psi] \subseteq \mathcal{N}'[\Psi]$ .
- (c)  $ffc(\mathcal{N})[\Psi] \subseteq \mathcal{N}[\Psi]$ .
- (d) Si  $ffc(\mathcal{N}[\Psi]) \subseteq \mathcal{N}'$  alors  $ffc(\mathcal{N}[\Psi]) \subseteq \mathcal{N}'[\Psi]$ .

**Preuve** (a) et (b) sont immédiats. (c) découle de (b) et du fait que  $ffc(\mathcal{N}) \subseteq \mathcal{N}$ . Prouvons (d). Si  $ffc(\mathcal{N}[\Psi]) \subseteq \mathcal{N}'$  alors  $ffc(\mathcal{N}[\Psi])[\Psi] \subseteq \mathcal{N}'[\Psi]$  (Propriété (b)). De plus,  $ffc(\mathcal{N}[\Psi]) \subseteq ffc(\mathcal{N}[\Psi])[\Psi]$  (Propriété (a)). Nous pouvons en conclure que  $ffc(\mathcal{N}[\Psi]) \subseteq \mathcal{N}'[\Psi]$ .  $\dashv$

Considérons de nouveau l'algorithme Algorithme 1.  $\mathcal{N}^k$  dénotera le RCQ  $\mathcal{N}$  obtenu durant le  $k^{\text{ième}}$  appel de *cohérenceE'* après l'instanciation de la contrainte sélectionnée  $C_{i^k j^k}$  par la sous relation  $r_l^k$  (i.e. après la ligne 7 de *cohérenceE'*).  $\mathcal{N}^0$  correspond au RCQ initial  $\mathcal{N}$ . Clairement nous avons  $\mathcal{N}^k = ffc(\mathcal{N}^{k-1})[(i^k, j^k, r_l^k)]$  pour tout  $k > 0$ . L'ensemble  $\Psi_k = \{(\alpha_1^k, \beta_1^k, \gamma_1^k), \dots, (\alpha_{m_k}^k, \beta_{m_k}^k, \gamma_{m_k}^k)\}$ , avec  $k > 1$  représente l'ensemble des contraintes marquées lors de la  $k^{\text{ième}}$  exécution de la méthode de la fermeture par faible composition (ligne 1 de la fonction *cohérenceE'*). L'appartenance du triplet  $(\alpha_i, \beta_i, \gamma_i)$  à  $\Psi_k$  stipule que la contrainte  $C_{\alpha_i, \beta_i}$  a été marquée comme non éligible et était égale à la relation  $\gamma_i$  au moment du marquage.  $\Psi_k$  ne contient que les contraintes marquées  $C_{ij}$  avec  $i \leq j$ . De plus, nous définissons  $\Psi_1$  par l'ensemble les contraintes marquées comme non éligibles pendant la première exécution de la fermeture par faible composition auxquelles sont rajoutées celles marquées comme non éligibles pendant l'initialisation de la méthode, ligne 1 de la fonction *cohérenceE*. Nous avons la propriété suivante.

**Proposition 3**

$ffc(\mathcal{N}^k) = ffc(ffc(\mathcal{N}^k)[\Psi_{k+1}])$ , pour tout  $k > 0$ .

**Preuve**

- ( $\subseteq$ ) Les contraintes représentées par  $\Psi_{k+1}$  sont des contraintes marquées non éligible durant la fermeture par faible composition sur  $\mathcal{N}^k$  (et également durant l'initialisation de la méthode pour  $k = 0$ ). Par conséquent,  $ffc(\mathcal{N}^k) \subseteq ffc(\mathcal{N}^k)[\Psi_{k+1}]$ .

De la proposition 1 (c), nous déduisons que  $ffc(ffc(\mathcal{N}^k)) \subseteq ffc(ffc(\mathcal{N}^k)[\Psi_{k+1}])$ . Ainsi nous obtenons  $ffc(\mathcal{N}^k) \subseteq ffc(ffc(\mathcal{N}^k)[\Psi_{k+1}])$  (Proposition 1 (b)).  
 (⊃) À partir de la proposition 2 (c) nous avons  $ffc(\mathcal{N}^k)[\Psi_{k+1}] \subseteq \mathcal{N}^k[\Psi_{k+1}]$ . De plus, par définition de  $\Psi_{k+1}$  et de  $\mathcal{N}_k$  nous savons que  $\mathcal{N}^k[\Psi_{k+1}] \subseteq \mathcal{N}^k$ . Ainsi,  $ffc(\mathcal{N}^k)[\Psi_{k+1}] \subseteq \mathcal{N}^k$ . De la proposition 1 (c) il résulte que  $ffc(ffc(\mathcal{N}^k)[\Psi_{k+1}]) \subseteq ffc(\mathcal{N}^k)$ .

⊢

Nous définissons par  $\mathcal{S}^k$  le RCQ obtenu en substituant les relations des contraintes de  $\mathcal{N}$  marquées non éligibles jusqu'à la  $k^{\text{ième}}$  instantiation de contraintes par celles utilisées pour définir ces contraintes au moment du marquage de non éligibilité. Formellement,  $\mathcal{S}^k$  est défini par  $\mathcal{S}^0 = \mathcal{N}$  and  $\mathcal{S}^k = \mathcal{S}^{k-1}[\Psi_k][(i^k, j^k, r^k)]$  pour tout entier  $k > 0$ . Il est clair que  $\mathcal{S}^k$  est un sous-RCQ de  $\mathcal{N}$ . De plus, lorsque l'algorithme se termine par un succès, après  $k'$  instantiations, le RCQ  $\mathcal{S}^{k'}$  n'est défini qu'à partir de relations de l'ensemble *Split*. De plus nous avons la propriété importante suivante.

**Proposition 4**

$ffc(\mathcal{N}^k) = ffc(\mathcal{S}^k)$ , pour tout  $k \geq 0$ .

**Preuve** Puisque  $\mathcal{N} = \mathcal{N}^0 = \mathcal{S}^0$  la propriété est vraie pour  $k = 0$ . Maintenant supposons la vraie au rang  $k$  et prouvons la pour  $k + 1$ .

(⊆) Nous avons  $ffc(\mathcal{N}^k) = ffc(\mathcal{S}^k)$ . À partir de la proposition 1 (a) nous déduisons que  $ffc(\mathcal{N}^k) \subseteq \mathcal{S}^k$ . Il résulte que  $ffc(\mathcal{N}^k)[\Psi_{k+1}] \subseteq \mathcal{S}^k[\Psi_{k+1}]$  (Proposition 2 (b)). Ainsi,  $ffc(ffc(\mathcal{N}^k)[\Psi_{k+1}]) \subseteq \mathcal{S}^k[\Psi_{k+1}]$  (Proposition 1 (a)). À partir de cela et de la proposition 3 nous pouvons affirmer que  $ffc(\mathcal{N}^k) \subseteq \mathcal{S}^k[\Psi_{k+1}]$ . En conséquence de quoi  $ffc(\mathcal{N}^k)[(i^{k+1}, j^{k+1}, r^{k+1})] \subseteq \mathcal{S}^k[\Psi_{k+1}][(i^{k+1}, j^{k+1}, r^{k+1})]$  (Proposition 2 (b)). Par définition de  $\mathcal{N}^{k+1}$  et  $\mathcal{S}^{k+1}$  nous pouvons déduire que  $\mathcal{N}^{k+1} \subseteq \mathcal{S}^{k+1}$ . Nous pouvons conclure que  $ffc(\mathcal{N}^{k+1}) \subseteq ffc(\mathcal{S}^{k+1})$  (Proposition 1 (c)).

(⊃) Clairement  $\mathcal{S}^k[\Psi_{k+1}][(i^{k+1}, j^{k+1}, r^{k+1})] \subseteq \mathcal{S}^k$ . Ainsi  $ffc(\mathcal{S}^k[\Psi_{k+1}][(i^{k+1}, j^{k+1}, r^{k+1})]) \subseteq ffc(\mathcal{S}^k)$ . D'après la proposition 2 (d) nous pouvons déduire que  $ffc(\mathcal{S}^k[\Psi_{k+1}][(i^{k+1}, j^{k+1}, r^{k+1})]) \subseteq ffc(\mathcal{S}^k)[\Psi_{k+1}][(i^{k+1}, j^{k+1}, r^{k+1})]$ . Par définition de  $\mathcal{S}^{k+1}$  nous avons  $ffc(\mathcal{S}^{k+1}) \subseteq ffc(\mathcal{S}^k)[\Psi_{k+1}][(i^{k+1}, j^{k+1}, r^{k+1})]$ . De plus, comme la propriété est vraie au rang  $k$ , nous avons  $ffc(\mathcal{S}^k) \subseteq ffc(\mathcal{N}^k)$ . Clairement,  $ffc(\mathcal{S}^k)[\Psi_{k+1}] \subseteq ffc(\mathcal{S}^k)$ . Il résulte que  $ffc(\mathcal{S}^k)[\Psi_{k+1}] \subseteq ffc(\mathcal{N}^k)$ . En conséquence de quoi  $ffc(\mathcal{S}^k)[\Psi_{k+1}][(i^{k+1}, j^{k+1}, r^{k+1})] \subseteq ffc(\mathcal{N}^k)[(i^{k+1}, j^{k+1}, r^{k+1})]$ . À partir de tout cela nous pouvons affirmer que  $ffc(\mathcal{S}^{k+1}) \subseteq ffc(\mathcal{N}^k)[(i^{k+1}, j^{k+1}, r^{k+1})]$ . Il résulte que  $ffc(ffc(\mathcal{S}^{k+1})) \subseteq ffc(ffc(\mathcal{N}^k)[(i^{k+1}, j^{k+1}, r^{k+1})])$ . D'après la définition  $\mathcal{N}^{k+1}$  nous pouvons conclure que  $ffc(\mathcal{S}^{k+1}) \subseteq ffc(\mathcal{N}^{k+1})$ .

⊢

Ainsi, lorsque l'algorithme *cohérenceE* retourne **true** le réseau obtenu  $\mathcal{N}$  est un sous-RCQ du RCQ initial et ne contient que des relations de l'ensemble *Split* comme contraintes. De plus, il est o-fermé. Il s'ensuit que si la méthode de la fermeture par faible composition est complet pour *Split* alors le RCQ obtenu est cohérent. Nous pouvons donc déduire le théorème suivant.

**Théorème 1**

Si la méthode de la fermeture par faible composition est complète pour l'ensemble

$Split \subseteq 2^B$  alors  $cohérenceE$  est correct et complet pour le problème de la cohérence des RCQ définis sur  $2^B$ .

## 4 Résultats expérimentaux

Pour montrer l'intérêt pratique de notre approche nous avons réalisé une série de tests sur des réseaux de contraintes qualitatives de l'algèbre des intervalles. Ces réseaux ont été générés aléatoirement selon un modèle similaire à ceux proposés dans [3, 13]. Ce modèle considère les paramètres suivants : un entier  $n$  correspondant au nombre de variables du réseau ; le paramètre  $nonTrivialDensity$  qui est la probabilité qu'une contrainte soit non triviale ( i.e différente de B) ; le paramètre  $cardinalityDensity$  qui est la probabilité qu'une relation atomique appartienne à une contrainte non triviale donnée ;  $consistent$  qui est une valeur booléenne indiquant si le RCQ généré doit être cohérent ou non. Dans la suite nous donnerons des résultats expérimentaux concernant des RCQ générés avec 50 variables, avec  $nonTrivialDensity$  prenant trois valeurs possibles 0.25, 0.50 et 0.75. Le paramètre  $cardinalityDensity$  varie de 0.05 jusqu'à 0.95 avec un pas de 0.05. Par manque de place nous indiquons de manière détaillée uniquement les résultats concernant des RCQ cohérents. Pour chaque point, 100 RCQ sont générés et testés.

Les expériences ont été réalisées en JAVA à l'aide de la librairie QAT [5](Qualitative Algebra Toolkit). Nous avons implanté et comparé l'algorithme  $cohérenceE$  et celui de Nebel [13] qui n'utilise pas de marquages de contraintes. L'ensemble  $Split$  utilisé est l'ensemble des relations préconvexes [14, 8]. Concernant la méthode de fermeture par faible composition nous utilisons deux algorithmes différents. Pour le premier appel nous utilisons une méthode basée sur l'algorithme  $WCC\_Mixed$  [4] mixant  $PC1$  et  $PC2$  [10, 12]. Cette méthode est plus efficace lorsque la méthode de la fermeture par faible composition est suffisante pour détecter l'incohérence du RCQ. Pour les autres appels nous utilisons une version plus classique basée sur  $PC2$  et stockant les paires de contraintes à traiter dans une queue.

Lors de la recherche le choix des contraintes à traiter (ligne 2 de  $cohérenceE$ ) est dynamique. La première heuristique utilisée et que nous appellons  $Dom$  consiste à sélectionner une contrainte ayant la plus petite cardinalité parmi les contraintes sélectionnables. La deuxième heuristique que nous utilisons est plus originale. Nous la nommons  $DomTriangle$ . Comme l'heuristique précédente, cette heuristique considère les contraintes de plus petite cardinalité. À chacune de ces contraintes  $C_{ij}$  l'heuristique  $DomTriangle$  associe le poids  $\Sigma(|C_{ik}| + |C_{kj}|)$  où  $C_{ik}$  et  $C_{kj}$  sont non triviales et choisit une de celles ayant un poids minimal. L'idée de cette heuristique est de sélectionner une contrainte qui provoquera le plus de modifications lors des opérations de triangulation futures.

En examinant les courbes de la figure 1 nous constatons que pour la majorité des RCQ générés, l'algorithme  $cohérenceE$  s'arrête plus tôt. L'avantage est plus important avec l'utilisation de l'heuristique  $DomTriangle$  qu'avec celle de l'heuristique  $Dom$ . De plus, de manière générale, plus la densité  $nonTrivialDensity$  est faible et plus l'avantage est élevé. Une explication plausible de ce phénomène est que plus cette densité est

## Éligibilité pour la résolution de contraintes qualitatives

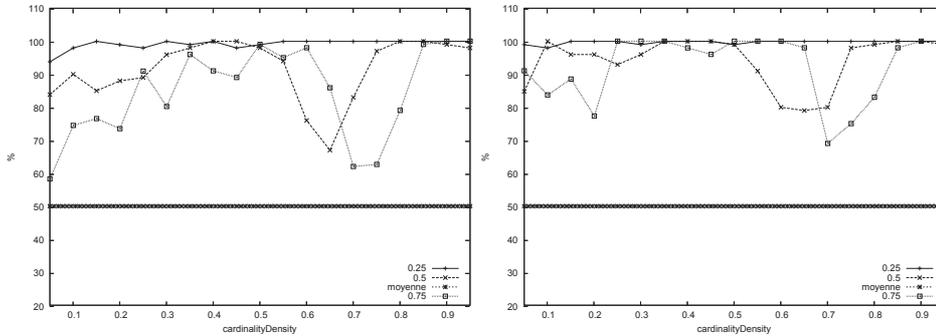


FIG. 1 – Pourcentages du nombre de RCQ résolus plus rapidement par *cohérenceE* pour  $nonTrivialDensity \in \{0.25, 0.5, 0.75\}$ , avec l’heuristique *Dom* à gauche et *DomTriangle* à droite.

faible, plus grand est le nombre de relations triviales, et donc de manière générale plus grand est le nombre de relations préconvexes, et plus tôt s’arrêtera *cohérenceE*.

Les tableaux suivants récapitulent les temps mis et le nombre de retours en arrière réalisés lors de la recherche. Le temps moyen est donné en millisecondes. La première colonne indique le temps mis par le premier appel de la fermeture par faible composition. Les deux suivantes concernent le temps de recherche effectué respectivement par *cohérenceE* et l’algorithme de Nebel. Dans les deux dernières colonnes sont indiqués les nombres de retours en arrière. Les tableaux de gauche concernent l’heuristique *Dom* et ceux de droite l’heuristique *DomTriangle*. Tout d’abord, nous remarquons que plus la densité de contraintes non triviales est élevée plus est élevée la difficulté de résolution des RCQ. De plus, pour chacune des trois densités de contraintes non triviales considérées il existe une phase correspondant à des valeurs bien précises de la densité *cardinalityDensity* pour lesquelles la résolution est particulièrement difficile. Pour  $nTD = 0.5$  cette phase correspond à  $cD \in \{0.6, 0.65\}$  et pour  $nTD = 0.75$  cette phase correspond à  $cD \in \{0.7, 0.75\}$ . Dans ces phases le nombre de retours en arrière est très important. Pour  $nTD = 0.25$ , *cohérenceE* est systématiquement plus rapide. Pour la densité  $nTD = 0.5$  et  $nTD = 0.75$ , *cohérenceE* est moins rapide dans quelques cas. Ces cas se situent toujours dans la phase difficile. De plus, nous remarquons qu’avec l’utilisation de l’heuristique *DomTriangle* ces cas sont moins nombreux : 1 pour  $nTD = 0.5$  et 2 pour  $nTD = 0.75$  contre respectivement 2 et 3 pour l’heuristique *Dom*. Nous remarquons aussi que pour la densité de contraintes non triviales 0.5 et 0.75 l’utilisation de l’heuristique *DomTriangle* permet d’avoir de meilleurs résultats en terme de temps et de retours en arrière que l’utilisation de l’heuristique *Dom*. Pour la densité de contraintes non triviales 0.25 il n’y a pas de différence significative entre ces deux heuristiques. Nous avons également réalisé des expériences sur des RCQ non forcés cohérents (des RCQ générés sans ajout d’une solution). Les premiers résultats concernant ces RCQ semblent montrer que *cohérenceE* est moins performante sur ce type de RCQ. Une explication possible à cela est que, dans le cas d’un RCQ non cohérent, la sélection des contraintes se faisant uniquement sur les contraintes marquées comme éligibles dans le cas de *cohérenceE*, celui-ci se dirige moins rapidement vers un noyau de contraintes non cohérent, contrairement à l’algorithme de Nebel qui peut sélectionner les contraintes les plus appropriées pour

nTD=0.25	$T_{ffc}$	$T_E$	$T_N$	$BK_E$	$BK_N$
cD=0.05	190	35	64	0	0
cD=0.1	203	42	78	0	0
cD=0.15	221	61	112	0	0
cD=0.2	232	86	143	0	0
cD=0.25	247	140	226	0	0
cD=0.3	240	213	322	0	0
cD=0.35	191	327	467	0	0
cD=0.4	110	450	613	0	0
cD=0.45	61	510	670	0	0
cD=0.5	37	511	689	0	0
cD=0.55	29	511	748	0	4
cD=0.6	27	502	692	0	0
cD=0.65	21	457	684	0	0
cD=0.7	19	398	610	0	0
cD=0.75	23	376	569	0	0
cD=0.8	18	314	495	0	0
cD=0.85	16	249	403	0	0
cD=0.9	13	151	252	0	0
cD=0.95	9	33	66	0	0

nTD=0.25	$T_{ffc}$	$T_E$	$T_N$	$BK_E$	$BK_N$
0.05	180	31	60	0	0
0.1	194	41	79	0	0
0.15	212	61	109	0	0
0.2	227	87	143	0	0
0.25	244	142	227	0	0
0.3	232	215	318	0	0
0.35	183	323	459	0	0
0.4	100	462	628	1	1
0.45	51	504	673	0	0
0.5	35	508	775	0	5
0.55	23	507	698	0	0
0.6	17	485	701	0	0
0.65	19	455	676	0	0
0.7	16	402	614	0	0
0.75	18	361	559	0	0
0.8	15	303	486	0	0
0.85	12	242	397	0	0
0.9	11	149	247	0	0
0.95	5	32	63	0	0

nTD=0.5	$T_{ffc}$	$T_E$	$T_N$	$BK_E$	$BK_N$
cD=0.05	137	11	17	0	0
cD=0.1	138	9	14	0	0
cD=0.15	151	11	18	0	0
cD=0.2	162	14	24	0	0
cD=0.25	175	18	33	0	0
cD=0.3	198	24	45	0	0
cD=0.35	231	34	63	0	0
cD=0.4	268	55	98	0	0
cD=0.45	296	90	155	0	0
cD=0.5	290	296	387	0	0
cD=0.55	108	1289	1320	19	12
cD=0.6	72	<b>37566</b>	<b>34338</b>	1078	839
cD=0.65	59	<b>28751</b>	<b>27061</b>	785	640
cD=0.7	52	4964	10096	136	190
cD=0.75	47	717	987	0	2
cD=0.8	48	626	873	0	0
cD=0.85	42	520	752	0	0
cD=0.9	31	384	564	0	0
cD=0.95	17	179	281	0	0

nTD=0.5	$T_{ffc}$	$T_E$	$T_N$	$BK_E$	$BK_N$
cD=0.05	135	11	17	0	0
cD=0.1	132	6	13	0	0
cD=0.15	142	9	17	0	0
cD=0.2	154	13	24	0	0
cD=0.25	175	19	34	0	0
cD=0.3	198	26	47	0	0
cD=0.35	225	33	62	0	0
cD=0.4	262	56	98	0	0
cD=0.45	298	94	153	0	0
cD=0.5	281	290	380	0	0
cD=0.55	109	1642	1724	29	23
cD=0.6	68	6272	6325	162	139
cD=0.65	51	<b>14188</b>	<b>11260</b>	429	297
cD=0.7	45	1078	1283	7	7
cD=0.75	40	715	935	0	0
cD=0.8	38	625	879	0	0
cD=0.85	36	522	748	0	0
cD=0.9	24	368	570	0	0
cD=0.95	13	167	263	0	0

nTD=0.75	$T_{ffc}$	$T_E$	$T_N$	$BK_E$	$BK_N$
cD=0.05	132	14	13	0	0
cD=0.1	134	9	12	0	0
cD=0.15	140	9	12	0	0
cD=0.2	149	9	13	0	0
cD=0.25	157	6	11	0	0
cD=0.3	180	13	17	0	0
cD=0.35	184	10	19	0	0
cD=0.4	223	20	35	0	0
cD=0.45	264	26	46	0	0
cD=0.5	282	32	63	0	0
cD=0.55	348	331	485	47	58
cD=0.6	186	561	681	1	1
cD=0.65	120	<b>3551</b>	<b>3312</b>	63	46
cD=0.7	110	<b>50848</b>	<b>43879</b>	1295	931
cD=0.75	94	<b>78982</b>	<b>66304</b>	1635	1160
cD=0.8	81	1516	2338	14	22
cD=0.85	75	756	990	0	0
cD=0.9	55	572	784	0	0
cD=0.95	32	318	474	0	0

nTD=0.75	$T_{ffc}$	$T_E$	$T_N$	$BK_E$	$BK_N$
cD=0.05	117	10	13	0	0
cD=0.1	125	8	12	0	0
cD=0.15	132	8	12	0	0
cD=0.2	143	9	13	0	0
cD=0.25	149	5	10	0	0
cD=0.3	163	8	15	0	0
cD=0.35	181	9	19	0	0
cD=0.4	216	19	37	0	0
cD=0.45	258	27	46	0	0
cD=0.5	286	34	62	0	0
cD=0.55	329	72	115	0	0
cD=0.6	175	543	650	0	0
cD=0.65	110	<b>3447</b>	<b>3179</b>	79	58
cD=0.7	91	<b>49158</b>	<b>42696</b>	1178	969
cD=0.75	84	73930	87883	1493	1536
cD=0.8	75	1308	1328	10	6
cD=0.85	65	768	1007	0	0
cD=0.9	48	567	775	0	0
cD=0.95	22	314	465	0	0

atteindre un tel noyau.

## 5 Conclusion

Dans cet article, nous avons proposé un algorithme basé sur la notion d'éligibilité de contraintes pour la résolution de réseaux de contraintes qualitatives. Nous avons prouvé que cet algorithme est correct et complet. Les premiers résultats expérimentaux obtenus montrent l'intérêt pratique de notre approche. Malgré tout de nouvelles expériences doivent être réalisées. En particulier, il serait intéressant de confronter *cohérenceE* à des RCQ de taille supérieure et d'autres formalismes que celui d'Allen. Il faudrait

également étudier le comportement de *cohérence* $E$  en utilisant différents ensembles de découpage, par exemple nous pourrions utiliser l'ensemble des relations convexes ou celui des relations pointisables dans le cadre de l'algèbre des intervalles.

## Références

- [1] J. F. Allen. An interval-based representation of temporal knowledge. In *Proc. of the Seventh Int. Joint Conf. on Artificial Intelligence (IJCAI'81)*, pages 221–226, 1981.
- [2] Philippe Balbiani, Jean-François Condotta, and Gérard Ligozat. On the consistency problem for the INDU calculus. In *Proceedings of TIME'2003*, pages 203–211, 2003.
- [3] Peter Van Beek and Dennis W. Manchak. The design and experimental analysis of algorithms for temporal reasoning. *Journal of Artificial Intelligence Research*, 4 :1–18, 1996.
- [4] J.-F. Condotta, G. Ligozat, and M. Saade. An empirical study of algorithms for qualitative temporal or spatial networks. In *Proceedings of the workshop on spatial reasoning (ECAI'06)*, 2006.
- [5] J.-F. Condotta, G. Ligozat, and M. Saade. The QAT : A qualitative algebra toolkit. In *Proceedings of the thirteenth international conference on Temporal Representation and Reasoning (TIME'2006)*, Hungary, 2006.
- [6] Amar Isli and Anthony G. Cohn. A new approach to cyclic ordering of 2D orientations using ternary relation algebras. *Artificial Intelligence*, 122(1–2) :137–187, 2000.
- [7] P. B. Ladkin and A. Reinefeld. Effective solution of qualitative interval constraint problems. *Artificial Intelligence*, 57(1) :105–124, september 1992.
- [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 Ligozat. Reasoning about cardinal directions. *Journal of Visual Languages and Computing*, 1(9) :23–44, 1998.
- [10] A. K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 1977, 8 :99–118, 1977.
- [11] A. K. Mackworth and E. C. Freuder. The Complexity of Some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems. *Artificial Intelligence*, 25(1) :65–74, 1985.
- [12] U. Montanari. Networks of constraints : Fundamental properties and application to picture processing. *Information Sciences*, 7(2) :95–132, 1974.
- [13] Bernhard Nebel. Solving hard qualitative temporal reasoning problems : Evaluating the efficiency of using the ORD-Horn class. In *Proceeding of the Twelfth Conference on Artificial Intelligence (ECAI'96)*, 1996.
- [14] 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.
- [15] Arun K. Pujari, G. Vijaya Kumari, and Abdul Sattar. INDU : An interval and duration network. In *Australian Joint Conference on Artificial Intelligence*, pages 291–303, 1999.
- [16] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *Proc. of the 3rd Conf. on Principles of Knowledge Representation and Reasoning (KR'92)*, pages 165–176, 1992.
- [17] J. Renz and B. Nebel. Efficient methods for qualitative spatial reasoning, 1998.

*RTE 2007*

# Deux approches pour la détection d'itinéraires dans du texte.

Pierre Loustau, Julien Lesbegueries

Laboratoire d'Informatique de l'UPPA  
Avenue de l'Université  
64000 Pau  
pierre.loustau@univ-pau.fr  
julien.lesbegueries@univ-pau.fr

**Résumé** : Nous présentons dans cet article deux approches aux objectifs différents mais ayant la même thématique de recherche : l'extraction et l'interprétation d'informations spatiales dans un corpus documentaire. La première est dans une problématique Recherche d'Information, la deuxième est dans le domaine de la conception d'application. Nous montrons cependant que ces deux approches peuvent converger et se compléter pour s'améliorer réciproquement dans le cas particulier des itinéraires.

**Mots-clés** : Ingénierie des connaissances, Recherche d'Information, Raisonnement Spatial

## 1 Introduction

Les travaux que nous présentons ici ont pour thème commun l'extraction et l'interprétation d'informations spatiales dans un corpus documentaire territorialisé pour deux usages différents : Recherche d'Information (RI) pour le premier et conception d'application pour le second. On entend par corpus territorialisé, un corpus composé de textes qui évoquent fréquemment un territoire particulier, ici la région des Pyrénées.

Nos précédents travaux ont porté sur l'extraction d'évocation de lieux géographiques plus ou moins complexes dans ces textes. Ces lieux ont la particularité d'apparaître en relation (syntaxique) avec un toponyme, sous-ensemble des entités nommées. Ces travaux ont permis d'implémenter d'un prototype (le prototype PIV de Lesbegueries & Loustau (2006)) et ont fait l'objet de publications (Lesbegueries *et al.* (2006)).

Nous souhaitons ici franchir un cap supplémentaire dans l'extraction et l'interprétation d'informations spatiales. En effet, l'extraction des évocations de lieux se fait à un niveau très local dans le texte : au niveau du syntagme nominal, le toponyme. Aucune relation ne peut être faite entre un lieu  $L_1$  et un lieu  $L_2$  provenant pourtant du même texte. Nous souhaitons analyser les relations entre les lieux afin de réaliser une interprétation de plus haut niveau. En effet, le besoin de mettre en relation ces lieux apparaît rapidement que ce soit pour des usages de RI ou pour des usages de conception d'ap-

plications utilisant des documents à connotation géographique. Les besoins de la RI Spécialisée dans le spatial sont de construire des indexes spatiaux multi-grains permettant l'appariement d'une requête avec l'index. Les besoins du concepteur d'application sont de pouvoir faire intervenir les contenus des documents dans la modélisation d'une application (Nodenot *et al.* (2006)).

Dans cet article, nous présentons deux approches qui ont des objectifs différents mais qui ont pour point commun d'aller plus loin dans l'interprétation de l'information spatiale. La première est dans le domaine de la RI et consiste à utiliser le géo-référencement des lieux afin de proposer des indexes spatiaux multi-grains sans pousser plus loin l'analyse linguistique. La seconde approche tente de ré-exploiter le texte pour extraire de nouveaux indices permettant au concepteur d'avoir une meilleure visibilité sur le contenu spatial des textes. Elle s'intéresse pour le moment à un contenu spatial particulier : l'itinéraire. Dans ce papier, nous montrons qu'avec des objectifs différents les deux approches distinctes que nous proposons peuvent se compléter et s'améliorer réciproquement dans le cas des itinéraires, très présents dans notre corpus.

Nos travaux s'insèrent à la fois dans le domaine des sciences cognitives, il faut effectivement définir le concept d'itinéraire, mais également dans celui du Traitement Automatique des Langues Naturelles (TALN), notamment pour l'extraction des ES et la deuxième approche que nous proposons.

Concernant le domaine de la modélisation des itinéraires, de nombreux auteurs s'accordent pour dire qu'à un haut niveau d'abstraction, un itinéraire est composé d'un point de départ, d'étapes intermédiaires et d'un point d'arrivée (Wunderlich & Reinelt (1982); Przytula-Machrouh *et al.* (2004); Denis (1994); Fraczak & Lapalme (1999)). Retenons le point commun de tous ces travaux : les propriétés de saillance des étapes intermédiaires. Dans le cas particulier des récits de voyage, des étapes intermédiaires apparaissent lorsque le voyageur traverse un lieu et qu'il décrit ce lieu.

Dans le domaine du TALN et de la linguistique, nous nous sommes intéressés aux travaux de Laur (1991); Sarda (1992); Muller & Sarda (1999) sur l'expression du déplacement en français. Ils montrent comment le sens spatial des verbes de déplacement peut-être modélisé. Ces résultats nous ont permis de construire une analyse des déplacements basée sur les verbes de déplacement.

Nous commencerons par présenter le socle commun de nos deux approches : l'extraction des ES. Nous détaillerons par la suite nos deux approches avant de conclure sur leur complémentarité éventuelle.

## **2 Extraction des évocations de lieux, vers la notion d'Entité Spatiale**

L'extraction des évocations de lieux dans des documents textes constitue la base commune des deux approches que nous proposons.

Dans cette section nous montrons l'importance qu'occupent les évocations de lieux dans la description d'un récit de voyage. Nous expliquons ensuite pourquoi nous avons évolué vers la notion d'Entité Spatiale (ES) et nous décrivons le modèle et les méthodes que nous avons mis en place pour extraire et interpréter les ES.

## 2.1 Expériences sur le corpus

L'extraction des évocations de lieux a pour origine l'observation d'un corpus documentaire constitué de récits de voyage, un genre littéraire à part entière.

On observe dans ces textes que l'itinéraire apparaît le plus souvent à l'aide d'évocations de lieux et de verbes de déplacement (*je suis arrivé à Pau à 11h, j'ai gravi le Pic du Midi d'Ossau le lendemain*, etc.). Nous nous intéressons ici aux évocations de lieux grâce à des toponymes, nous reviendrons sur les verbes de déplacement.

Texte	Mots	Propositions Position (PP)	PP avec Tp	%
James-David Forbes	3662	75	63	80%
Ann Lister	3225	40	20	50%
Bals	19015	213	170	80%

FIG. 1 – Le poids des Toponymes (Tp) parmi les Propositions Position (PP)

Cette première expérience consiste à relever dans trois récits de voyage<sup>1</sup> les propositions qui évoquent la position de l'auteur dans son voyage (PP) et à comptabiliser combien parmi elles utilisent des toponymes dans ce but (PP avec Tp).

Les résultats sont donnés dans la figure 1. L'emploi des toponymes y apparaît assez nettement. On atteint des taux de l'ordre de 80% selon les auteurs, mais aussi selon la nature du voyage qui est relaté. En effet, le déficit d'emploi de toponyme dans l'évocation du déplacement apparaît le plus souvent lorsque l'auteur évoque des déplacements de plus petites tailles (*je suis allé au parc, j'ai quitté l'hôtel*, etc.), le plus souvent lorsqu'il est à une étape intermédiaire. C'est le cas pour le texte de Ann Lister, dans lequel l'auteur évoque largement ses occupations lors des étapes le long de son voyage.

## 2.2 Modèle et méthode pour l'extraction d'ES

Au vu de ces expériences mais également de travaux antérieurs, le concept de lieu a été remplacé par la notion d'Entité Spatiale (ES). La connotation zonale du lieu semble en effet réductrice par rapport au concept d'ES dans lesquels on peut retrouver des lignes (une route, une rivière), des points (une Eglise), etc.

Nos précédents travaux sur l'extraction d'ES (Lesbegueries *et al.* (2006)) nous ont permis de définir un modèle pour représenter ces ES. On distingue deux grandes catégories d'ES, selon qu'elles puissent être géo-référencées, directement ou indirectement, dans un repère géographique donné (et ce à l'aide de ressources comme : des Gazetteers, des SIG<sup>2</sup> ...). On distinguera donc : les *ES Absolues* et les *ES Relatives*. On obtient directement la géo-localisation des *ES Absolues* comme celle de : *Pau, le bois de Zouhoure, etc.* ; alors que les *ESR* sont évoquées relativement à une où plusieurs *ES Absolues* et de relations spatiales : leurs géométries sont donc généralement moins précises (*près de Pau, à la lisière du bois de Zouhoure, etc.*).

L'extraction des ES est réalisée par une chaîne de traitement linguistique basée sur le processus classique de (i) lemmatisation, (ii) analyse morpho-syntaxique, (iii) ana-

<sup>1</sup>Le voyage de James David Forbes, l'excursion de J.-R. Bals et Le voyage d'Ann Lister

<sup>2</sup>SIG : Système d'Information Géographique

```

1 ES  --> ESA | ESR.
2 ESR --> Relation , ES.
3 Relation --> relation_orientation | relation_adjacence | etc. .
4 ESA --> toponyme.

```

FIG. 2 – Aperçu simplifié des règles DCG de notre approche

lyse sémantique. La particularité de notre approche se situe dans l'analyse sémantique et consiste, par reconnaissance de motifs, en une recherche des ESA et des indicateurs spatiaux de relation liés à celles-ci. Cette méthode a été mise en application par des lexiques et des grammaires DCG<sup>3</sup> dont un aperçu très simplifié est donné en figure 2.2. Nous renvoyons à Lesbegueries *et al.* (2006) pour plus de précisions. Dans cette approche entièrement automatique, l'ESA est confondue avec le toponyme (*Pau, le bois de Zouhoure, etc.*). Avec des ressources plus importantes ou l'intervention humaine, on pourrait envisager de caractériser de nouvelles ESA qui ne sont pas exprimées par des toponymes (*le port, la gare, etc.*).

### 3 Approche géographique

L'hypothèse de cette approche est que l'analyse d'un groupe d'ESs géo-référencées permet la déduction du type de contexte spatial (ou motif spatial) utilisé dans le récit, en particulier celui de la description d'un itinéraire. En effet, par unité de texte, c'est-à-dire par groupe de phrases, de paragraphes ou de sections, etc., des ESs sont extraites et une représentation géo-référencée leur est associée. Nous partons de ces représentations et de la configuration qu'elles ont dans l'unité pour réaliser une interprétation.

Cette analyse se fait en ne retenant dans les considérations linguistiques qu'une simplification du principe de séquentialité et considère que le "sens spatial" d'une unité de texte peut être reconstruit uniquement avec cet ensemble de zones géo-référencées égrenées dans un certain ordre le long du texte. Cette méthode permet d'envisager une utilisation dans un système d'indexation pour une application de recherche d'information spécialisée. Cette étude nous a amenés à calculer des caractéristiques basées sur des modélisations cognitives de l'espace et des itinéraires (Cohn & Hazarika (2001); Fraczak & Lapalme (1999); Klippel *et al.* (2005)).

Enfin, ce travail de caractérisation a permis de différencier la description d'itinéraire d'autres motifs spatiaux dont la description spatiale locale et la comparaison de lieux.

Nous allons présenter maintenant les caractéristiques calculées pour les sacs d'ESs afin de classer les unités de texte selon ces trois motifs.

#### 3.1 Observation des propriétés d'un groupe d'ESs

Les propriétés suivantes peuvent être observées sur un groupe d'ESs dans un espace géographique : (i) la dispersion : les ESs peuvent être contenues les unes dans les autres,

<sup>3</sup>DCG : Definite Clause Grammar

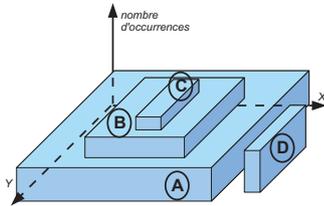


FIG. 3 –  $A_{prev}$  : exemple de calcul. Dans une unité de texte où A, B, C et D sont des ESs, (A) intersecte avec (B) et (C), mais pas avec (D).

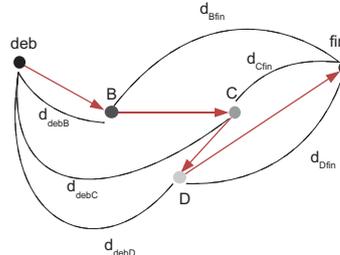


FIG. 4 –  $O_{SF_s}$  : exemple de calcul d'ordonnancement des points intermédiaires B, C, D.

être proches ou dispersées ; (ii) l'ordonnancement : les ESs connexes dans le texte dessinent un itinéraire ou non ; (iii) la saillance : les ESs connexes dans le texte forment des angles plats, obtus ou aigus.

De plus des propriétés intrinsèques aux ESs comme leur échelle par exemple, sont utilisables pour le calcul des caractéristiques.

### 3.2 Déduction des caractéristiques à évaluer pour la classification

À partir de cet étude, une première évaluation de ces propriétés a été réalisée au travers des quatre caractéristiques.  $A_{prev}$  est une évaluation de la prédominance d'une zone géographique par rapport à l'ensemble des ESs citées (figure 3). Pour cela un calcul de cumul d'intersection des zones représentant les ESs est effectué. Cette caractéristique évalue la propriété de **dispersion**.  $O_{SF_s}$  est une évaluation de l'ordonnancement géographique des entités spatiales mentionnées tout au long du texte. La propriété associée est l'**ordonnancement**. Dans la figure 4 nous avons 3 conditions (une pour chaque point intermédiaire B, C, D) et 2 d'entre elles sont vérifiées : l'ordonnancement  $B \rightarrow C$  et  $D \rightarrow E$ , en fonction des distances au point de départ  $deb$  et d'arrivée  $fin$ . L'algorithme est présenté en annexe dans la figure 8.  $D_{SF_s}$  est une évaluation des distances entre les entités spatiales contenues dans le texte. Elle correspond à la distance moyenne qui les sépare. La propriété associée est la **dispersion**.  $S_{SF_s}$  est une évaluation de la **saillance** géographique des entités spatiales. Elle permet de connaître l'orientation que prend le chemin d'une ES à l'autre (algorithme présenté en annexe, figure 9).

### 3.3 Classification

Ces caractéristiques donnent toutes une valeur entre 0 et 1, permettant de placer l'élément étudié (le groupe d'ESs correspondant à une unité de texte) dans un espace vectoriel avec pour dimensions les différentes caractéristiques calculées. Une méthode de

classification par apprentissage de type SVM<sup>4</sup> (Joachims (1998); Hasan (2006)) est alors envisagée.

Cette classification permettra de déterminer pour une unité de texte si l'auteur traite plutôt d'un itinéraire, d'une description ou d'une comparaison de lieux. Il se peut que l'unité de texte ne corresponde à aucune de ces classes. Le cas échéant elle sera classée comme nulle. Par exemple, un extrait de texte sera classifié comme itinéraire si la valeur de la caractéristique  $A_{prev}$  est petite et si les valeurs de  $O_{SF_s}$  et  $S_{SF_s}$  sont grandes.

### 3.4 Avantages et limitations

L'avantage recherché est l'efficacité de la méthode du point de vue des ressources disponibles et de sa complexité computationnelle, par l'utilisation de ressources structurées de type ressources géographiques, l'objectif final étant son intégration dans un SRI.

Les limitations de cette approche concernent la richesse d'interprétation moins forte qu'avec un traitement linguistique. Les résultats obtenus sont une classification simple avec une indication sur les ESs les plus significatifs pour la classe choisie. De plus, il peut subsister des ambiguïtés que cette approche ne peut résoudre sans l'aide de traitement linguistique supplémentaire, comme par exemple 2 classes impossibles à dissocier : la classe *itinéraire* et la classe *description d'un point de vue* où les ESs sont souvent décrites de manière ordonnée, par exemple quand l'auteur évoque une chaîne de montagne en détaillant les pics les uns à la suite des autres ou quand il les parcourt vraiment. Ces 2 classes correspondent à la même configuration (c'est-à-dire aux mêmes valeurs pour les caractéristiques). Dans ce cas, une étude sur les verbes de déplacement pourrait aider la classification.

## 4 Approche linguistico-géographique

L'approche que nous proposons ici tente de reconstruire l'itinéraire parcouru par l'auteur en restant plus près du texte. Non seulement à partir des ES évoquées mais également grâce aux déplacements de l'auteur dans l'espace géographique. Cette méthode est décrite plus en détails dans Loustau *et al.* (2007).

### 4.1 Extraction des déplacements de l'auteur

#### *Importance des verbes de déplacement dans l'évocation du déplacement*

Nous revenons ici sur l'observation du corpus documentaire en nous intéressant à l'évocation de l'itinéraire grâce aux verbes de déplacement. (*je suis sorti de Pau à 11h, j'ai gravi le Pic d'Anie le lendemain, etc.*)

Cette seconde expérience a pour but d'évaluer le poids des verbes. Elle consiste à comptabiliser dans les expressions qui évoquent le déplacement le nombre d'expressions qui utilisent des formes verbales à cette fin. Avec une moyenne de 87,7% (les résultats par document sont donnés en figure 5) cette étude montre leur prédominance.

<sup>4</sup>Support Vector Machines traduit en français par Machines à Vecteurs de Supports ou Séparateurs à Vastes Marges.

Texte	Mots	Propositions Position (PP)	PP avec verbe	%
James-David Forbes	3662	75	63	80%
Ann Lister	3225	40	35	87,5%
Bals	19015	476	389	76%

FIG. 5 – Expérience 2 : le poids des verbes dans l'évocation du déplacement

Les expériences 1 et 2 concernant l'importance des ES et des verbes de déplacement dans l'évocation du déplacement doivent encore être approfondies en augmentant le nombre de récits étudiés. Elles donnent cependant une bonne idée du poids qu'occupent les ES et les verbes de déplacement dans notre corpus. Ces observations corroborent également les travaux autour de l'évocation du déplacement dans la langue tels que ceux de Laur (1991); Sarda (1992); Muller & Sarda (1999).

**Modélisation des déplacements**

Nous reprenons ici principalement le critère de polarité aspectuelle des verbes introduit par Boons (1987) et repris par Laur (1991). Ainsi, nous modélisons les déplacements dans la langue en prenant en compte les verbes de déplacement (qui seront donc *initiaux*, *médians* ou *finaux*) obligatoirement associés à des ES et de manière facultative à des prépositions spatiales. Le Verbe, la Préposition (facultative, d'où le " ? ") et l'Entité forment le triplé (V, P?, E). Ce triplé est discriminant pour faire émerger le sens spatial de certains verbes polysémiques. En effet, si *quitter une personne* ne nous intéresse pas, *quitter Bordeaux* renferme bien un sens spatial. Les déplacements extraits seront *initiaux* (quitter), *médians* (traverser) ou *finaux* (arriver).

L'origine, la destination et la position médiane interviennent également : les constructions verbales du déplacement font en effet émerger une (*quitter Pau*), deux (*quitter Pau pour Bordeaux*) ou trois (*quitter Pau pour Bordeaux par la RN 134*) de ces propriétés.

D'autre part, les expériences réalisées sur le corpus (cf figure 5), montrent que dans une première approche, la construction des verbes de déplacement peut-être apparentée à un langage régulier, et peut alors être modélisée par des machines à états finis (figure 6). Le triplé (V,P,E) se modélise alors grâce à un transducteur. Un transducteur est un dispositif qui transforme un langage donné en un autre. Ils sont basés sur des machines à états finis mettant en correspondance deux langages réguliers.

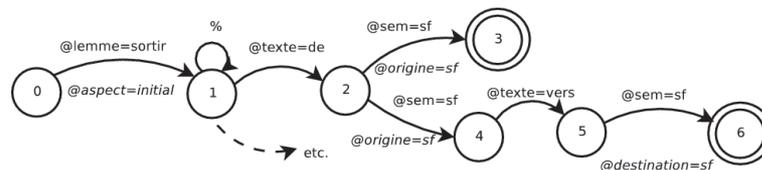


FIG. 6 – Extrait partiel et simplifié du transducteur du verbe sortir. Le symbole % représente le "joker".

**Application au corpus, implémentation des transducteurs**

La démarche générale consiste à construire une chaîne de traitement linguistico-spatiale capable d'extraire les déplacements dans des textes. Nous décrivons ici les méthodes et

les outils utilisés par chaque phase du traitement.

La spécificité de notre approche est de considérer que les déplacements sont évoqués par le triplé (V,P,E) et que ce triplé est modélisable sous forme de transducteur spatial.

La chaîne de traitement que nous proposons alors est une chaîne classique du domaine du TALN. Le texte est segmenté en unités lexicales. Une analyse morpho-syntaxique retourne alors la forme et le lemme des unités lexicales et permet de s'abstraire des formes fléchies des mots. Elle est effectuée par Tree-Tagger, un analyseur morpho-syntaxique éprouvé (Schmid (1994)). Enfin, une fois les ES extraites (cf 2.2), les transducteurs des verbes de déplacements sont traduits en grammaires DCG pour extraire les déplacements. L'analyse basée sur ces grammaires est confiée à Prolog afin de profiter des mécanismes de déduction et d'unification de ce langage.

#### ***Evaluation de la méthode***

Nous avons évalué les capacités de notre outil en les comparant à une extraction manuelle dans trois textes pour lesquels nous avons annoté manuellement les déplacements de l'auteur. On obtient un taux de réussite de l'ordre de 78% pour l'extraction automatique des déplacements de l'auteur (plus de détails dans Loustau *et al.* (2007))

## **4.2 Reconstruction de l'itinéraire à partir des déplacements extraits**

Nous décrivons ici le passage de déplacements extraits à un itinéraire. Comment s'élever au niveau du discours à partir d'éléments extraits localement ?

#### ***Méthode proposée***

La méthode que nous proposons s'appuie sur une hypothèse (les auteurs évoquent les déplacements de leur itinéraire dans un ordre chronologique) et sur des ressources (des données factuelles, quelques règles simples et des raisonnements plus complexes).

Les principales ressources sont de type géographique. Tout comme l'humain doit avoir une connaissance géographique du monde pour pouvoir raisonner sur un itinéraire (Kuipers (1977)), un système doit également avoir des capacités équivalentes. On peut classer ces ressources en deux catégories : les données et les raisonnements.

Les **données factuelles** sont des données brutes, que l'on pourrait qualifier de données universelles. Elles s'apparentent aux connaissances du monde que peut avoir l'humain. Les couches de données SIG (comme celles des communes de France, du réseau routier, ou du réseau fluvial), les Gazeteers (dictionnaire de noms de lieux, de localisation, etc.), les bases toponymiques, sont des données factuelles envisageables.

Les **raisonnements** sont de deux types. Les plus élémentaires, s'effectuent directement sur les données factuelles et peuvent être ainsi qualifiés de raisonnement *bas niveau*. Ces raisonnements sont très souvent offerts par les outils traitant l'Information Géographique, ils permettent notamment d'utiliser des opérateurs et des fonctions afin de déduire de nouvelles informations à partir des données factuelles.

Les "Contraintes", qui sont des raisonnements de plus *haut niveau*, sont apportées au système pour en modifier le comportement, ou en améliorer les interprétations. Elles tentent d'apporter au système des règles de sens commun. (i) Des règles situant l'acteur par rapport à l'entité et mettant en relation la polarité aspectuelle du verbe avec les notions d'origine de destination et d'emplacement médian. (ii) règles concernant l'étendue des zones probables de localisation de l'acteur en fonction de la polarité as-

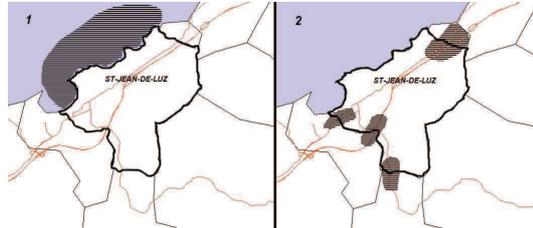


FIG. 7 – Quitter Saint-Jean-de-Luz : les zones probables selon la modalité du transport. (1) en bateau,(2) en voiture

pectuelle des verbes et de la modalité de transport. (iii) règles concernant les modalités de transport et leur localisation probable. Ces deux dernières contraintes concernant les modalités de transport sont illustrées sur la figure 7.

#### **Application sur le corpus**

Le module de reconstruction de l'itinéraire prend en entrée un fichier XML dans lequel les déplacements sont représentés selon les critères que nous avons évoqués dans la section 4.1. Il utilise le SIG PostGIS<sup>5</sup> afin de mettre en application les règles de raisonnement spatial précédemment évoquées. Ce module est en cours d'élaboration, nous en donnons ici les grands principes.

Pour chaque déplacement extrait, on cherche tout d'abord à produire une zone probable de localisation de l'acteur. Pour cela nous faisons appel aux règles concernant la polarité aspectuelle et la modalité du transport. Les fonctions *buffer* et *intersection* du SIG PostGIS ont été utilisées à ces fins. Considérons le déplacement de l'exemple : « Nous avons quitté Bordeaux pour Langon en voiture. ». Celui-ci a pour origine Bordeaux et pour destination Langon. Il fait donc émerger un segment reliant les deux relais que sont Bordeaux et Langon. La modalité du déplacement étant *voiture*, on va faire appel à un algorithme de calcul de trajet dans le réseau routier pour construire la route probablement empruntée par l'acteur du déplacement. Cette fonctionnalité est apportée par le module *pgRouting*<sup>6</sup>. Rappelons qu'il ne s'agit pas forcément de la réalité ; l'acteur a peut-être pris un autre chemin. Par cette méthode, on cherche à approcher la représentation mentale de l'itinéraire que se fait le lecteur dans sa carte cognitive.

#### **Évaluation de la méthode**

La reconstruction de l'itinéraire à partir des déplacements extraits localement devra faire l'objet d'une évaluation. Les modalités de cette évaluation restent à définir.

### **4.3 Avantages et limitations**

En prenant en compte les verbes de déplacement, leur polarité aspectuelle, les notions d'origine, de destination, de position médiane et la modalité du transport, on arrive à approcher de manière fine le chemin qu'a emprunté l'auteur durant son voyage. On peut ainsi offrir au concepteur d'application la possibilité de trouver le passage du texte

<sup>5</sup>PostGIS : extension GIS du système de gestion de base de donnée libre PostgreSQL

<sup>6</sup>*pgRouting* est une extension au SIG PostGIS lui donnant des fonctions de calcul de plus court chemin

pour lequel l'auteur a quitté telle ou telle région, est entré dans telle ville, etc. D'autre part, on évite l'introduction d'un certain bruit pouvant conduire à une mauvaise interprétation. En effet, seules les références à l'espace géographique associées à un verbe de déplacement sont prises en compte pour construire l'itinéraire. Les ES évoquées en relation avec des verbes de perception par exemple sont éliminées.

Des limites apparaissent cependant : par exemple lorsque l'auteur fait référence, alors qu'il est en train de décrire son itinéraire, à un autre voyage qu'il a fait ailleurs (souvent il les compare). Notre approche est là mise en échec. Cependant, si on combine cette approche avec l'approche précédemment proposée, on doit pouvoir remédier au problème. En effet, dans ce cas là, l'approche géographique conclut à un motif de type comparatif et dissocie donc les deux descriptions d'itinéraire.

## 5 Bilan et perspectives

Ces travaux menés en parallèle montrent que des domaines de recherche différents proposent des solutions pour la problématique de la détection d'itinéraires dans du texte.

La première approche, dont le processus est moins lourd permet d'envisager une utilisation dans une problématique de RI spécialisée. En effet, la classification d'une unité de texte en itinéraire, description ou comparaison de lieux, etc. permet de créer un index spatial au niveau des unités de texte et non plus des syntagmes nominaux (les ESs). Cet index permet alors un accès plus rapide à l'information tout en étant plus pertinent par rapport à une requête spécialisée (recherche d'itinéraire, de description ou de comparaison de lieux).

La deuxième approche est fondée sur l'analyse des verbes de déplacement. Elle permet d'extraire automatiquement les déplacements de l'auteur et de reconstruire l'itinéraire de l'auteur à l'aide de ressources afin de proposer au concepteur d'application de faire intervenir le contenu modélisé des documents directement dans la phase de conception. Il s'agit là de pistes pour la reconstruction de l'itinéraire à partir des déplacements extraits localement. Nos recherches se focalisent actuellement sur ce point.

Les notions temporelles intrinsèques aux itinéraires ont volontairement été mises de côté. Des travaux concernant le temps, proches de la démarche que nous avons adoptée pour le spatial, sont en cours et permettront d'améliorer la classification d'une part et l'interprétation linguistique d'autre part sans altérer les travaux présentés ici.

Des travaux futurs pourront nous amener à faire converger les deux approches. Par exemple, l'approche linguistico-géographique peut permettre de désambiguïser deux classes pour l'approche géographique. D'un autre côté, l'approche géographique pourrait éliminer certains "bruits" perturbant l'approche linguistico-géographique.

## Références

- BOONS J.-P. (1987). La notion sémantique de déplacement dans une classification syntaxique des verbes locatifs. *Langue Française*, **76**, 5–40.
- COHN A. G. & HAZARIKA S. M. (2001). Qualitative spatial representation and reasoning : An overview. *Fundamenta Informaticae*, **46**(1-2), 1–29.

- DENIS M. (1994). La description d'itinéraires : des repères pour des actions. In *Notes et Documents LIMSI*.
- FRACZAK L. & LAPALME G. (1999). Utilisation de stratégies cognitives dans la génération automatique de descriptions d'itinéraires. In *TALN'99*, p. 10 pages, Cargèse.
- HASAN M. (2006). *SVM : Machines à Vecteurs de Support ou Séparateurs à Vastes Marges*. Rapport interne, Versailles St Quentin, France.
- JOACHIMS T. (1998). Text categorization with support vector machines : learning with many relevant features. *Lecture Notes in Computer Science, 10th European Conference on Machine Learning, series(1398)*, 137–142.
- KLIPPEL A., DAVIES J., WINTER S. & HANSEN S. (2005). A high-level cognitive framework for route directions. In *SSC Spatial Intelligence, Innovation and Praxis*.
- KUIPERS B. (1977). Modeling spatial knowledge. In *IJCAI*, p. 292–298.
- LAUR D. (1991). *Sémantique du déplacement et de la localisation en français : une étude des verbes, des prépositions et de leur relation dans la phrase simple*. PhD thesis, Université de Toulouse II.
- LESBEGUERIES J., GAIO M., LOUSTAU P. & SALLABERRY C. (2006). Geographical information access for non-structured data. In *21st ACM Symposium on Applied Computing, ASIIS track*, p. 83–89, Dijon.
- LESBEGUERIES J. & LOUSTAU P. (2006). Structuration d'information spatiale qualitative pour la recherche d'information. In *Semaine De la Connaissance*, p. 1–6, Nantes.
- LOUSTAU P., GAIO M. & NODENOT T. (2007). Des déplacements à l'itinéraire, du syntagme au discours. Colloque SAGEO 2007, Clermont-Ferrand. A paraître.
- MULLER P. & SARDA L. (1999). Représentation de la sémantique des verbes de déplacement transitifs du français. *T.A.L.*, **39**(2), 127–147.
- NODENOT T., LOUSTAU P., GAIO M., SALLABERRY C. & LOPISTEGUY P. (2006). From electronic documents to problem-based learning environments : an ongoing challenge for educational modeling languages. In *7th International Conference on Information Technology Based Higher Education and Training*, p. 75–86 : IEEE.
- PRZYTULA-MACHROUH E., LIGOZAT G. & DENIS M. (2004). Vers des ontologies transmodales pour la description d'itinéraires. *Revue Internationale de Géomatique : n° Spécial sur les ontologies spatiales*, **14**(2), 285–302.
- SARDA L. (1992). *L'expression du déplacement dans la construction transitive directe*. In *Syntaxe & Sémantique - Sémantique du lexique verbal*, p. 121–137. 2. Presses Universitaires de Caen.
- SCHMID H. (1994). Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK.
- WUNDERLICH D. & REINELT R. (1982). How to get there from here. In R. J. JARVELLA & W. KLEIN, Eds., *Speech, Place, and Action*, p. 183–201. Chichester : Wiley.

## 6 Annexes

```

1 L ← Liste ordonnée des positions d'entités spatiales (ESs)
2   de petite granularité
3 P_deb ← Position de la première entité de petite granularité
4 P_fin ← Position de la dernière entité de petite granularité
5 // V est un vecteur dont les éléments sont des paires
6 // (distance du point de départ, distance du point d'arrivée).
7 V : Vecteur (de paires de flottants)
8 Pour chaque élément e de L Faire {
9   // La fonction distance() donne la distance géographique
10  // entre l'ES de position P_deb et l'ES de position e.
11  dist_deb ← distance(P_deb, e)
12  dist_fin ← distance(e, P_fin)
13
14  V.ajouter([dist_deb, dist_fin])
15 }
16 // V = [[d_deb1, d_fin1], [d_deb2, d_fin2], ..., [d_debn, d_fin]]
17 //
18 //
19 //
20 //
21 //
22 //
23 //
24 //
25 //
26 Soit Nv la taille du vecteur V
27 numérateur ← 0
28 dénominateur ← (Nv-1)+(Nv-2)+...+2+1 = Nv(Nv-1)/2
29 Pour a allant de 0 à Nv Faire {
30   Pour b allant de a+1 à Nv Faire {
31     d1 ← V[a][0]
32     d2 ← V[a][1]
33     d3 ← V[b][0]
34     d4 ← V[b][1]
35     // Condition 1 : (d1 < d3) et Condition 2 : (d2 > d4)
36     Si ((d1 < d3) et (d2 > d4)) Alors {
37       numérateur ← numérateur + 1
38     }
39   }
40 }
41 O_SF_s ← numérateur / dénominateur

```

FIG. 8 – Algorithme de calcul de  $O_{SF_s}$

```

1 L ← Liste ordonnée des positions d'entités spatiales (ESs)
2   de petite échelle
3 // Graphe des coûts de passage d'une orientation à l'autre :
4 // ex. : G[N,O] = 0.5, G[E,SO] = 0.0
5 //
6 //
7 G ← {
8   [1.0, 0.8, 0.5, 0.0, 0.0, 0.0, 0.5, 0.8], //N
9   [0.8, 1.0, 0.8, 0.5, 0.0, 0.0, 0.0, 0.5], //NE
10  [0.5, 0.8, 1.0, 0.8, 0.5, 0.0, 0.0, 0.0], //E
11  [0.0, 0.5, 0.8, 1.0, 0.8, 0.5, 0.0, 0.0], //SE
12  [0.0, 0.0, 0.5, 0.8, 1.0, 0.8, 0.5, 0.0], //S
13  [0.0, 0.0, 0.0, 0.5, 0.8, 1.0, 0.8, 0.5], //SO
14  [0.5, 0.0, 0.0, 0.0, 0.5, 0.8, 1.0, 0.8], //O
15  [0.8, 0.5, 0.0, 0.0, 0.0, 0.5, 0.8, 1.0] //NO
16 }
17 // En fonction de ces coûts une évaluation est faite du taux
18 // de «sailance» des entités.
19 Pour chaque élément e de L (moins le dernier) et
20 e' l'élément suivant e Faire {
21   V.ajouter(Orientation(e,e'))
22 }
23 Soit Nv la taille de V
24 numérateur ← 0.0
25 dénominateur ← Nv-1
26 Pour a allant de 0 à (Nv-1) Faire{
27   numérateur ← numérateur + G[V[a]][V[a+1]]
28 }
29 S_SF_s ← numérateur / dénominateur

```

FIG. 9 – Algorithme de calcul de  $S_{SF_s}$

# Une approche cognitive de l'itération et sa modélisation

Yann Mathet<sup>1</sup>

<sup>1</sup>Laboratoire GREYC, Université de Caen,  
mathet@info.unicaen.fr

**Résumé :** Nous étudions dans cet article une façon de représenter et de manipuler les itérations issues d'énoncés en français, comme celles provenant des syntagmes suivants : « tous les lundis », « à chaque fois que », « souvent ». Après avoir analysé les spécificités des itérations par rapport aux procès non itérés, nous en proposons un modèle objet, et mettons en œuvre ce dernier au travers d'un exemple. La visée de ce travail est en premier lieu descriptive : comment représenter au mieux, de façon formelle, manipulable, et autant que possible cognitivement plausible, les itérations ? Il pourra constituer par la suite le support de traitements automatiques d'analyse sémantique des itérations dans des textes, comme par exemple pour un système d'interrogation lors de requêtes telles que « combien de fois » ou « à quelle fréquence ».

**Mots-clés :** Sémantique temporelle, Itération, Espaces Mentaux, Modèle cognitif.

## 1 L'itération en langue : enjeux et définition

### 1.1 Présentation

La modélisation de la temporalité se fait souvent au moyen d'intervalles disposés sur un axe temporel, et liés entre eux par un certain nombre de relations. L'utilisation de différents types d'intervalles (d'énonciation, de procès notés [B1 B2], de référence notés [I II], circonstanciel) et de relations (recouvrement, antériorité, etc.) permet de rendre compte avec finesse, comme il est montré dans (Gosselin 1996), de notions telles que le temps relatif, ou encore l'aspect. Le travail sur l'itération que nous avons mené dans le cadre du projet OGRE (Lévy & al. 2005) nous montre cependant la nécessité de disposer d'un modèle plus riche, pouvant notamment se prêter à la manipulation d'entités itérées. En effet, même dans une itération simple comme :

(1) Elle est allée sept fois à la montagne,

il y a bien sûr, en filigrane, 7 procès, les **itérés**, à créer et à ordonner, mais il y a aussi, parallèlement, à coder le fait que ces différents procès résultent du même prédicat « elle <aller> à la montagne », l'**itérant**. Le lien entre les différents itérés et leur existence même n'ont de sens qu'en présence de l'itérant dont ils résultent. Par ailleurs, la façon dont les itérés sont produits est en soi un processus varié, et dont il

faut rendre compte. Pour ces raisons, nous proposons un modèle « objet » de la temporalité qui permet de considérer l'itération à part entière et a vocation à s'intégrer à la théorie SdT (Sémantique De la Temporalité) de (Gosselin 1996), afin de pouvoir notamment rendre compte, au sein même des itérations, de phénomènes tels que l'aspect.

## 1.2 Définition

Nous appelons itération la répétition dans le temps d'un même procès. Par *répétition dans le temps*, nous entendons que les intervalles de procès correspondants (c'est-à-dire les intervalles des itérés) soient tous disjoints deux à deux. Par *même procès*, nous entendons le fait qu'une forme infinitive unique puisse rendre compte de chaque itéré, ou au moins qu'il existe une forme infinitive qui subsume les différentes formes infinitives correspondant aux différents itérés, ou groupes d'itérés. Enfin, cette forme infinitive doit comporter un sujet explicite et identifiable, et ce sujet doit rester stable d'un itéré à l'autre. La difficile question de cette définition est très bien présentée en introduction de (Condamines 1992).

## 2 Analyse et représentation des itérations : itérant et itérés

### 2.1 Exemple introductif

Le processus d'itération a ceci de particulier qu'il génère plusieurs procès (les itérés) à partir d'un seul (l'itérant). Dans l'exemple 1, il résulte de l'interprétation du syntagme « sept fois » la création de sept procès distincts et disjoints qui s'étalent dans le temps. Mais ces différents procès, les **itérés**, ont un statut particulier, un air de famille, puisqu'ils résultent du clonage d'un même procès, *elle <aller> à la montagne*. Ils se distinguent en cela de 7 procès qui seraient issus de la conjugaison successive de 7 verbes. Nous pouvons schématiser ce processus, dans le cas de notre exemple, de façon simple : dans une itération, nous sommes en présence d'un itérant (*elle <aller> à la montagne*), qui donne lieu à la création d'itérés (sept procès étalés dans le temps) par le biais d'un itérateur (*sept fois*), selon le schéma de la figure 1.

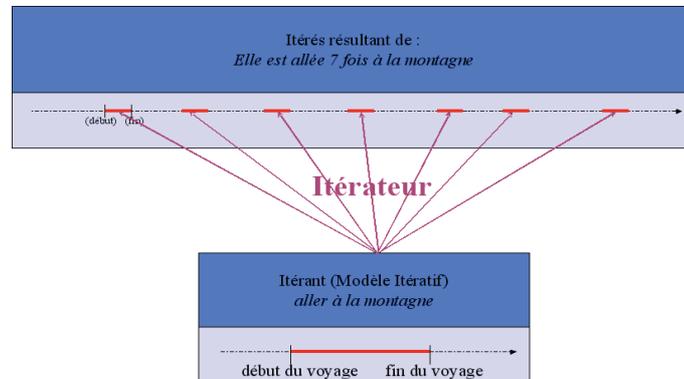


Fig. 1 – Itérant et itérés

## 2.2 Une hypothèse cognitive : les deux facettes de l'itération

La figure 1 fait apparaître deux aspects fondamentaux de l'itération. Comment intégrer cette dualité au contexte linguistique d'une part, et au modèle d'autre part ? Dans cette étude, nous faisons l'hypothèse suivante : les itérations temporelles sont appréhendées selon le double point de vue extensionnel et intensionnel, parfois indépendamment, parfois simultanément, de façon opportuniste et variable selon le contexte et les individus. En d'autres termes, nous supposons que face à une situation itérée, nous pouvons être amené à considérer ce tout : soit comme un ensemble de procès étalés dans le temps (point de vue extensionnel, correspondant au haut de la figure 1), venant le cas échéant se placer parmi d'autres procès (issus ou non d'autres itérations) ; soit comme une situation unique (point de vue intensionnel, correspondant au bas de la figure 1), dont on sait certes qu'elle est dupliquée (i.e. dont on connaît la réalité extensionnelle) mais desquelles duplications on prend le parti de s'abstraire afin de n'en considérer que le « modèle » commun.

Certaines configurations linguistiques semblent orienter le point de vue plutôt vers l'une ou l'autre de ces deux facettes, mais il y a toujours moyen de basculer d'un point de vue à l'autre. Observons ce continuum linguistique. « Le trois septembre, elle est allée pour la première fois à la piscine, faisant une heure de brasse. Elle y est retournée trois semaines plus tard, pour une séance de deux heures environ, et une troisième fois hier matin, de 8h à 10h » privilégie clairement la facette extensionnelle, décrivant chaque procès indépendamment. A l'opposé, « A cette époque, ma journée commençait vers 8 heures. Après une rapide douche, je descendais déjeuner, puis ... » met en avant la facette intensionnelle, décrivant une journée type. Enfin, « Tous les jeudis, nous faisons une partie de cartes » se situe entre les deux : on peut le ramener vers l'une ou l'autre des facettes, en le faisant suivre respectivement par « En tout, il y a eu 12 jeudis. Le premier jeudi, on a joué de 20h à 22h, les trois jeudis suivants, etc. » (facette extensionnelle), ou par « La partie commençait par le mélange puis la distribution des cartes. Puis, etc. » (facette intensionnelle). Le présent article se focalise principalement sur la facette intensionnelle, mais des études algébriques

comme (Becher&Enjalbert 2005) abordent la question sous un angle extensionnel en considérant des intervalles non connexes comme support de l'itération.

### 2.3 Espaces Mentaux et Modèles Itératifs

Même si notre objectif et notre objet d'étude diffèrent notablement de ceux des « Espaces Mentaux » de (Fauconnier, 1984), nous reprenons à notre compte un certain nombre de principes et d'éléments de ces derniers. En effet, les deux visions possibles des itérations semblent pouvoir correspondre à deux espaces mentaux différents :

à partir d'un espace mental de référence M, dans lequel sont placés, de façon extensionnelle, des procès correspondant à une situation itérée, il est possible de construire un espace plus abstrait M', dans lequel cette situation itérée est appréhendée de façon unique.

L'espace M' sert à construire et à manipuler la situation itérante, c'est-à-dire la situation servant de modèle aux différents itérés qui sont présents dans M.

Les propriétés de M' se déduisent à la fois de celles de M, et de l'introduit-eur itératif qui en est à l'origine. Ces propriétés sont d'ordre temporel d'une part, et référentiel de l'autre. **Temporellement**, M' ne permet qu'une appréhension réduite de l'axe temporel, d'une longueur inférieure ou égale à celle du plus grand itéré potentiellement constructible de M. Par ailleurs, M' n'a pas accès aux différents intervalles de ses espaces parents. Enfin, et de façon corollaire, M' n'a aucune visibilité sur les intervalles d'énonciation de ses espaces parents. Il en résulte un détachement de ces espaces mentaux vis-à-vis de la temporalité absolue (passé, présent et futur). **Du point de vue de la référence**, Les éventuelles entités itérées dans M sont vues comme une entité unique dans M', laquelle ne garde que les propriétés communes à l'ensemble des entités itérées (nivellement par la non-spécificité). Par exemple, dans « l'année dernière, elle s'est achetée une (nouvelle) robe tous les jeudis, toujours dans les tons pastels », nous trouvons 52 référents de robes dans l'espace M (peut-être certaines rouge pastel, d'autres vert pastel, si la suite du texte le précise), et une seule robe « modèle » dans l'espace M', dont la propriété couleur est simplement d'être pastel. Par contre, les entités non itérées de M sont potentiellement accessibles dans M'. De même que chez Fauconnier, les espaces mentaux itératifs peuvent être bâtis de façon récursive, c'est-à-dire qu'un espace M'' peut être construit à partir de l'espace parent M'.

## 3 Un modèle de l'itération

### 3.1 Itération

Une itération est composée d'un **Itérateur** et d'un **modèle itératif**.

L'**itérateur** est un objet permettant d'itérer des événements dans le temps. Cela peut être en proposant un ensemble d'intervalles comme dans « tous les jeudis », mais aussi par d'autres moyens comme dans « 3 fois », « souvent », « quand », etc.

Le **modèle itératif** correspond à ce qui est itéré. Il est constitué d'un ou de plusieurs procès modèles, qui peuvent être liés les uns aux autres par des relations temporelles ou d'un autre ordre (par exemple la causalité).

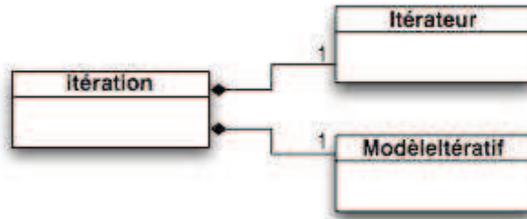


Fig 2 – Itération

### 3.2 Itérateur

Le rôle d'un itérateur est de proposer une suite de positions de l'axe temporel à chacune desquelles on pourra faire correspondre temporellement une instance du modèle d'itération. Les différentes classes d'itérateurs sont les suivantes :

Ensemble d'intervalles calendaires (cas particulier : intervalle non convexe) : tous les jeudis (calendaire régulier), un lundi sur trois (calendaire fréquentiel), lundi et mercredi derniers (itérateur calendaire)

Numéraire : 3 fois, quelques fois, manger 3 pommes

Événementiel : quand ils viennent, dès qu'ils arrivent, lorsqu'il fait beau : état sujet à changements vrai/faux

Fréquentiel : souvent, fréquemment, de temps en temps

Composé, c'est-à-dire résultant de la juxtaposition de plusieurs itérateurs

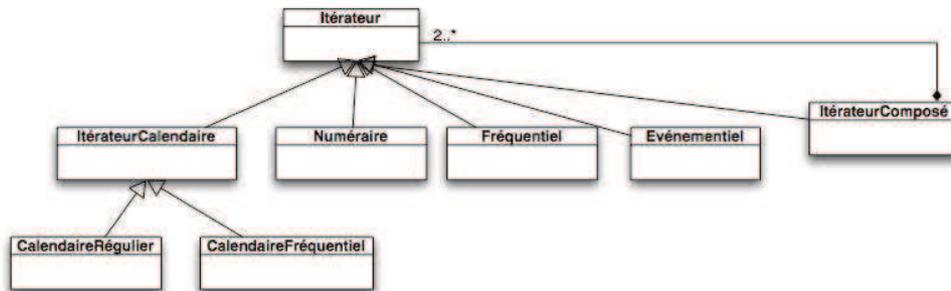


Fig. 3 – Itérateur

### 3.3 Le modèle Itératif et ses constituants

#### 3.3.1 Modèle itératif

Le rôle d'un modèle itératif est de servir de modèle à ce qui est itéré. Par exemple, dans l'exemple 1, il s'agit du processus modèle *elle [aller] à la montagne*. Formellement, le modèle itératif est constitué de 1 à n processus modèles, et de relations entre ces différents processus modèles.

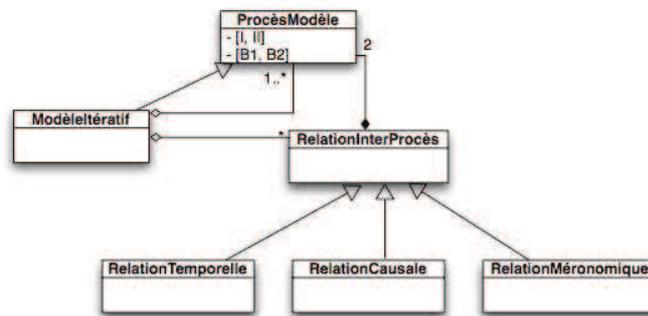


Fig. 4 – Modèle itératif

Ce schéma représente l'agrégation entre le modèle itératif et ses processus modèles d'une part, et entre le modèle itératif et les relations inter processus d'autre part. Par ailleurs, une relation inter processus met en jeu exactement deux processus modèles.

#### 3.3.2 Procès modèle

Un processus modèle est un processus qui vient se placer dans un espace mental modèle. Rappelons que dans un tel espace, l'accès à la temporalité est entièrement focalisé sur une représentation générique d'un ou de plusieurs processus (en particulier, les intervalles présents aux niveaux supérieurs, via les espaces parents, ne sont pas accessibles). Il reprend à son compte les caractéristiques principales des processus classiques.

#### 3.3.3 Relations entre processus modèles

N'ayant pas travaillé spécifiquement sur ces relations, nous nous contentons d'en proposer un simple aperçu, au travers de 3 cas : relation temporelle dans « elle arrivait après Pierre », relation méronomique d'élaboration dans « les parties de poker commençaient par la distribution des cartes », et relation causale.

#### 3.3.4 Constructions récursives : l'itération comme processus modèle.

Enfin, une itération peut elle-même constituer un processus modèle, et donc être inscrite dans une autre itération, récursivement, comme dans l'exemple (2) :

(2) Tous les dimanches, ils se baignaient deux fois.

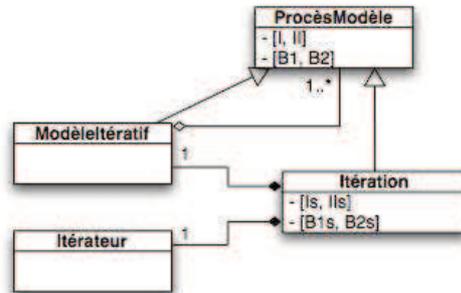


Fig. 5 – Constructions récursives

Toutefois, lorsqu'une Itération sert de ProcèsModèle à une autre itération, ses itérés s'inscrivent dans un espace mental modèle.

### 3.4 Mise en œuvre des espaces modèles : liens entre procès modèles

Les espaces modèles permettent de s'affranchir des difficultés de manipulation et de représentation posées par les itérations. Tout revient finalement à considérer, dans de tels espaces, les procès comme s'ils n'étaient pas itérés :

(3) Chaque jeudi, à 20 heures, le jeu débutait. A cette heure là, la maison était généralement assez calme depuis un long moment. Les parties s'enchaînaient alors. La soirée se terminait vers 22 heures, par un apéritif.

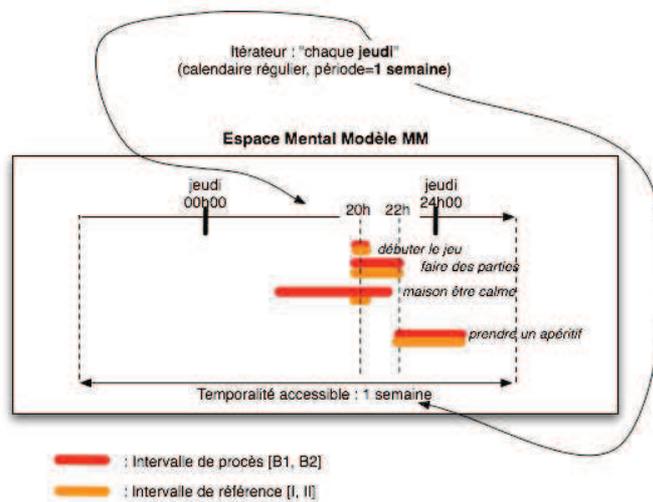


Fig. 6 – Liens entre procès modèles

## 4 Les sélections

Les itérations permettent, comme nous venons de le modéliser, de créer un certain nombre d'itérés à partir d'un itérant. Un mécanisme supplémentaire, que nous appelons *sélection*, permet, une fois une itération créée, de sélectionner un sous-ensemble de ses itérés. D'un point de vue linguistique, elle répond aux syntagmes tels que *une fois sur trois, les trois premières fois, la troisième fois* (sélecteur singleton), ou des fréquentiels tels que *souvent, parfois, de temps en temps* (dans le contexte d'une itération). Reprenons l'exemple 1, et faisons le suivre par :

(4) Les trois premières fois, c'était dans les Alpes.

La phrase 4 opère une sélection sur les itérés de l'itération induite par 1, au sens où « les trois premières fois » fait référence aux trois premiers itérés.

D'un point de vue formel, une sélection opérant une restriction sur une itération, elle constitue elle-même, une itération. Nous ferons donc de la classe Sélection une classe fille de la classe Itération, et la munirons de deux attributs principaux : l'itération sur laquelle elle opère, et son sélecteur, qui est la fonction de restriction de l'ensemble des itérés. Les différentes façon qui existent en langue de faire des sélections nous ont amené à construire une hiérarchie de sélecteurs sous forme d'arbre d'héritage, dans le détail duquel nous ne pouvons entrer ici.

Bien sûr, le rôle d'une sélection ne se limite jamais à la seule sélection d'itérés, auquel cas elle n'apporterait rien au discours, mais toujours à modifier ou enrichir les itérés sélectionnés. Nous ne pouvons développer cet aspect ici, mais l'illustrons dans la mise en œuvre proposée dans la partie 5.

## 5 Mise en œuvre sur un exemple

Voyons un exemple mettant en œuvre les notions essentielles présentées dans cet article, et notamment les notions d'itération, de modèle itératif, et de sélection :

(5) Souvent, quand je me promenais, je rencontrais Jean. Plus rarement, je rencontrais Anne. Ces fois là, j'étais vraiment heureux. Ma promenade était toujours suivie de petits achats en centre ville.

Une itération est créée par la première phrase, bâtie sur les occurrences de « je <se promener> », de type événementielle.

Le modèle itératif correspondant comprend pour le moment ce seul procès modèle. Simultanément, cette première phrase crée une sélection, issue de « souvent », indiquant que parmi toutes les occurrences de promenade, beaucoup comprennent l'événement « je <rencontrer> Jean ». Nous ici la puissance linguistique qui, en deux mots successifs (souvent, quand), crée coup sur coup une itération et une sélection. La deuxième phrase est relative à une autre sélection, elle aussi bâtie sur l'itération de base, et de fréquence moindre. Enfin, la dernière phrase revient sur le modèle initial, et vient enrichir ce dernier en lui ajoutant le procès modèle « je <faire achats> ».

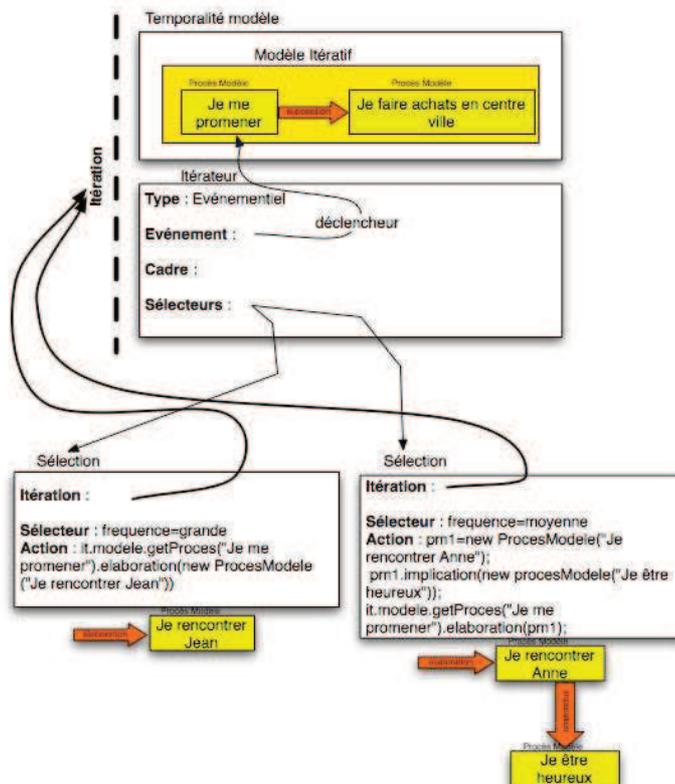


Fig. 8 – Mise en œuvre

## 6 Conclusion

Si les phénomènes temporels sont largement étudiés en linguistique, et mis en œuvre dans différents modèles de TAL, la question spécifique de l'itération l'est

beaucoup plus rarement, et, à notre connaissance, un modèle dédié faisait défaut. Des travaux comme ceux de (LIM 2002) offrent un panorama très complet des manifestations linguistiques de l'itération en français, mais la prise en compte des phénomènes sémantiques associés et leur intégration à une théorie de la temporalité nous semble faire défaut. Notre proposition s'intègre au cadre de la SdT de L. Gosselin, et en offre une extension permettant de prendre en considération l'itération à part entière, comme le fait la langue. Elle pourra donc venir compléter l'implémentation partielle du modèle SdT effectuée par (Person 2004).

D'un point de vue linguistique, l'étude de l'itération fait apparaître des phénomènes sémantiques complexes, souvent insoupçonnés, et nous invite à revisiter la notion de même de procès et de référence. Ainsi, si l'on considère l'exemple « tous les dimanches, il est d'astreinte. Sa journée commence à 4h30 (...) », on peut s'interroger sur le statut du référent de « sa journée ». Il s'agit manifestement de « tous les dimanches », bien que la phrase fasse état d'une journée, en employant un sujet au singulier. Notre modèle rend tout à fait compte de ce phénomène, notamment en proposant de se plaçant dans un modèle itératif dans lequel on envisage « tous les dimanches » comme une seule « journée », comme le fait le locuteur.

Du point de vue du TAL, cette étude est une première étape au cours de laquelle nous avons identifié les spécificités de l'itération, et proposé un modèle en rendant compte. Nous orientons nos recherches actuelles dans deux directions complémentaires. L'une consiste à circonscrire automatiquement les itérations dans des textes ; l'autre à produire, à partir des éléments linguistiques précédemment circonscrits, des représentations sémantiques dans les termes du présent modèle. Nous espérons à terme pouvoir interroger un texte sur son contenu itératif.

## Références

- BECHER G. & ENJALBERT P. (2005). Sémantique des Compléments Circonstanciels Temporels Itératifs (CTI) : Bases algébriques, Projet OGRE, rapport technique PI-TCAN, 54-72.
- CONDAMINES A. (1992). Expression de la répétition dans une phrase à subordonnée temporelle. *J. Fontanille (ed.) : La quantité et ses modulations qualitatives : Pulim/Benjamins, collection Nouveaux Actes Sémiotiques*, 111-128.
- FAUCONNIER G. (1984). *Espaces Mentaux*. Editions de Minuit.
- GOSSSELIN L. (1996). *Sémantique de la temporalité en français*, Duculot, Louvain-la-Neuve.
- GOSSSELIN L. & PERSON C. (2005). Temporalité, *Sémantique et traitement automatique du langage naturel*, dirigé Par Enjalbert P., Lavoisier, Paris, 173-211
- KLEIBER G. (1987). *Du côté de la référence verbale : les phrases habituelles*, Peter Lang, Berne.
- LEVY F. & AL. (2005), Projet OGRE « Ordre de Grandeur et Répétition », rapport technique PI — TCAN Programme interdisciplinaire CNRS « Traitement des connaissances, apprentissage et NTIC »
- LIM J.-H. (2002). *La fréquence et son expression en français*, Honoré Champion Editeur, Paris.
- PERSON C. (2004). *Traitement automatique de la temporalité du récit, implémentation du système SdT*, Thèse de l'Université de Caen.

# Vers le Web sémantique géospatial avec ONTOAST

Alina Dia Miron — Jérôme Gensel — Marlène Villanova-Oliver — Hervé Martin

Laboratoire d'Informatique de Grenoble,  
681 Rue de la Passerelle BP 72, 38402 Saint Martin d'Hères Cedex  
prenom.nom@imag.fr

**Résumé:** La représentation et l'analyse des relations spatiales complexes qui lient plusieurs ressources font partie des défis posés actuellement par la construction du Web sémantique. L'explicitation de telles relations permettrait d'étendre les capacités des moteurs de recherche pour prendre en compte des critères spatiaux dans les requêtes afin d'en préciser davantage la formulation et le résultat attendu. Or, les langages d'ontologies pour le Web sémantique, OWL en tête, n'offrent pas encore les caractéristiques nécessaires à une représentation et une exploitation spécialisées des données spatiales. Nous décrivons ici ONTOAST, une extension du Système de Représentation de Connaissances AROM, dédiée à la modélisation d'ontologies géographiques. ONTOAST intègre des relations spatiales qualitatives permettant l'expression de requêtes spatiales plus flexibles. Il permet également de déduire par inférence de nouvelles relations qualitatives, augmentant ainsi les capacités de recherche de données spatialement référencées.

**Mots-clés :** Web sémantique, ontologie, relation spatiales qualitative, inférence.

## 1. Introduction

La plupart des activités humaines étant bien localisées dans l'espace géographique, il n'est pas étonnant que la plus grande partie des documents présents sur le Web contiennent des références spatiales à travers l'utilisation de noms de lieux ou d'adresses... S'il est patent que, dans la plupart des cas, la composante géospatiale n'est pas systématiquement utilisée (notamment lors de la recherche de documents via les moteurs existants), elle peut s'avérer utile, par exemple, pour affiner certaines requêtes. Ainsi, dans le cas trivial de la recherche d'une personne, dont on connaît le nom, l'emploi d'un moteur de recherche sur ce seul critère du nom retourne parfois plusieurs résultats, qui ne font pas tous référence au même individu. Une solution pour affiner les réponses est de faire appel aux informations concernant la localisation de la personne visée (sa ville de résidence, l'adresse de son travail, ...). L'exploitation d'une information géospatiale, qu'elle soit une donnée ou une métadonnée, requiert des formalismes de représentation et de raisonnement adaptés aux particularités de ce type d'information et compatibles avec le Web sémantique (Berners-Lee *et al.*, 2001). Ce nouveau Web a été imaginé comme une évolution du Web d'aujourd'hui vers une gigantesque base de connaissances répartie, facilement exploitable autant par les humains (par l'intermédiaire des moteurs de recherche) que par les agents logiciels. Pour satisfaire à cet objectif certaines conditions sont requises. Dans un premier temps, à chaque ressource sur le Web devrait être associée une représentation bien formalisée de son contenu informationnel ainsi que des méta-données (son créateur, sa localisation, etc.). Le formalisme de modélisation qui s'est imposé, pour associer du sens aux ressources, est celui des ontologies (Gruber, 1993). Une ontologie est un

modèle du domaine de discours contenant un ensemble de concepts correspondant à l'ensemble des entités généralement (et consensuellement) reconnues comme représentant le domaine d'application traité, leurs définitions et les relations qui les lient.

Le langage qui s'impose comme standard pour la représentation et la modélisation des ontologies pour le Web sémantique est OWL (McGuinness & van Harmelen, 2005). Il découle de travaux de recherche sur les Logiques de Descriptions, sur les Langages de *Frames* ou sur les Réseaux Sémantiques et offre une grande expressivité ainsi qu'une palette étendue de raisonneurs (Pellet, RacerPro, Fact, Cerebra...). Cependant, s'agissant de données référencées spatialement ou/et temporellement, on constate qu'un langage évolué comme OWL s'avère trop générique et finalement peu adapté à la gestion spécifique de ce genre de données. Ce constat nous a amené à faire évoluer AROM (Page *et al.*, 2000), un Système de Représentation de Connaissances par Objets (SRCO) qui appartient à la même famille de langages de représentation de connaissances que OWL, pour le doter de capacités de modélisation des données géospatiales. AROM dispose ainsi de deux extensions: i) AROM-ST (Moisuc *et al.*, 2004), qui permet la gestion de bases de connaissances spatio-temporelles dédiées au domaine de la géomatique et ii) AROM-ONTO (Miron *et al.*, 2007), qui permet d'exprimer en AROM des ontologies à la manière de OWL.

Notre objectif est, dans un premier temps, d'intégrer ces deux extensions dans un nouveau module : ONTOAST (pour ONTOlogies en Arom-ST). Les atouts d'ONTOAST résident autant dans ses capacités à modéliser et à exploiter des ontologies spatio-temporelles que dans sa compatibilité avec le standard OWL, et donc son ouverture vers l'annotation des ressources du Web sémantique. Dans un deuxième temps, nous proposons l'introduction d'un ensemble de relations qualitatives géospatiales dans ONTOAST, pour accroître sa flexibilité et son expressivité ainsi que pour permettre des raisonnements lorsqu'on ne dispose pas de données spatiales précises. Des associations géospatiales complexes pourront désormais être définies à l'aide d'équations exprimées dans le Langage de Modélisation Algébrique d'AROM, étendu ici par des opérateurs de topologie, d'orientation et de proximité des objets. Bénéficiant de la compatibilité entre AROM et OWL, ONTOAST permettra donc de décrire des ontologies géospatiales pour le Web sémantique, ainsi que de formuler et de traiter des requêtes qualitatives sur ces ontologies, en exploitant les mécanismes de raisonnement d'AROM.

L'article est organisé comme suit. La section 2 détaille le contexte dans lequel se situe notre travail. La section 3 présente le système AROM, les extensions AROM-ST et AROM-ONTO et leur intégration telle qu'envisagée au sein du module ONTOAST. La section 4 décrit les relations qualitatives spatiales, topologiques, d'orientation et de proximité gérées par ONTOAST et la section 5 conclut en donnant les perspectives de ce travail.

## **2. Contexte**

Parmi les défis à relever dans la concrétisation d'un Web sémantique que l'on souhaiterait (aussi) géospatial, on trouve celui de l'analyse efficace et exhaustive des relations complexes (thématiques, spatiales ou temporelles) liant plusieurs ressources, ainsi que le besoin de gérer des requêtes – et donc des données – flexibles et imprécises. Une formulation plus souple (car plus proche des termes employés dans le

langage usuel) des requêtes adressées aux sources disponibles sur le Web peut être favorisée par l'usage de formalismes *qualitatifs* de représentation et de raisonnement. La recherche de données dans le domaine du *qualitatif* (Cohn *et al.*, 2001) simplifie les raisonnements et permet des déductions lorsque l'on ne dispose pas d'informations numériques, offrant ainsi davantage de flexibilité.

Une gestion efficace des relations spatiales qualitatives se fait au sein des Systèmes d'Information Géographique (SIG). Cependant, ces dernières contiennent rarement des relations spatiales explicitées, celles-ci étant plutôt déterminées par des calculs géométriques effectués à la demande. Par exemple, il est peu probable qu'une base de données géospatiale contienne explicitement la relation topologique d'inclusion entre une ville et le pays dont elle fait partie. Il est alors plausible d'utiliser l'opérateur d'inclusion topologique reposant sur le calcul géométrique, pour vérifier cette contrainte, au moment où ce genre d'interrogation apparaît. Ces approches sont nécessaires, car il n'est pas raisonnable de pré-calculer et stocker dans les bases de données *toutes* les relations topologiques existantes entre *tous* les objets spatiaux. Nous pouvons aisément transposer le problème de la gestion de données qualitatives au sein d'un SIG au Web sémantique, dès lors que le Web sémantique est vu comme une base de données (les ressources) géospatiales.

Il convient donc de munir la sphère du Web sémantique de langages de représentation et de mécanismes de raisonnement dédiés à l'information géospatiale, dont elle ne dispose pas encore aujourd'hui. Plus concrètement, OWL n'a pas été développé dans le but précis de modéliser des données spatio-temporelles. En conséquence, la plupart des ontologies OWL ne parviennent pas encore à encoder correctement ce genre d'information. Si (Katz & Grau, 2006) montrent que la représentation spatiale n'est pas une limitation fondamentale de OWL, elle est toutefois encore loin d'être intuitive. Par leur part, (Dolbear & Hart, 2006) ont étudié la traduction d'une base de données spatiales ORACLE vers une ontologie. Pour modéliser la spatialité des concepts, les auteurs proposent l'utilisation de tableaux de coordonnées, en vue de pallier le manque de types spatiaux dédiés en OWL. Leur conclusion est que même si elle est ardue, une telle traduction est possible. Cependant, les informations exportées depuis les SIG vers les ontologies OWL restent peu exploitables, car les raisonneurs OWL n'offrent pas de moyens d'inférer de nouvelles relations spatiales à partir des connaissances existantes. Ainsi, pour décrire la spatialité des ressources, on doit s'en remettre à l'information explicitement décrite par les ontologies auxquelles on fait référence. Par exemple, si on connaît les coordonnées géographiques (le contour) de la France et celles de Grenoble, on peut les associer aux ressources qui représentent respectivement la France et la ville de Grenoble, mais on ne peut pas inférer que Grenoble est une ville française. Néanmoins, les relations spatiales qualitatives importées depuis les ontologies peuvent être intégrées et exploitées par les SIG spécialisés. Ayant les coordonnées de la France et la relation d'inclusion avec la ville de Grenoble, un SIG spécialisé pourra inférer, par exemple, le fait que Grenoble est située dans l'Europe, au Nord de la mer Méditerranée...

Nous présentons, dans la section suivante, le SRCO AROM, outil de modélisation et de gestion d'ontologies, capable de gérer des données spatiales et temporelles,

### 3. De AROM à ONTOAST

AROM (Page *et al.*, 2000) est un outil générique pour la modélisation et l'exploitation des connaissances, qui s'inscrit dans la lignée des SRCO.

L'une des originalités d'AROM par rapport à ses prédécesseurs est la représentation explicite des relations entre *classes* à l'aide d'*associations*. De plus, AROM intègre un LMA (Langage de Modélisation Algébrique) qui permet de décrire des variables, des requêtes et des contraintes, dans un formalisme proche des notations mathématiques. Le LMA d'AROM est utilisé pour vérifier et maintenir la cohérence des données ou bien pour interroger et exploiter les entités d'une base de connaissances. Dans ce but, le LMA gère un ensemble d'opérateurs prédéfinis pour les types AROM, parmi lesquels on peut citer de façon non exhaustive : les opérateurs arithmétiques de base (addition, soustraction, puissance, partie entière, etc.), les opérateurs de comparaison (supérieur, inférieur, différent, etc.), les opérateurs trigonométriques (sinus, cosinus, tangente, etc.).

Un atout supplémentaire d'AROM est la présence d'un module de gestion de types extensible (Capponi, 1995). En exploitant cette extensibilité nous avons créé le module AROM-ST (Moisuc *et al.*, 2004) qui intègre une partie du modèle de données GML proposé par l'OpenGIS<sup>1</sup>. Il s'agit des types simples *Point*, *Polyline*, *Polygon*, *Line* et *LinearRing*, ainsi que des types géométriques complexes (composés) : *MultiPoint* (nuage de points), *MultiLine* et *MultiArea*. Pour pouvoir faire des requêtes, exprimer des contraintes, ou encore raisonner sur les types spatio-temporels en AROM-ST, nous avons aussi étendu le LMA d'AROM en introduisant une série d'opérateurs spatiaux et temporels. Il s'agit des *opérateurs topologiques* exprimés par des prédicats binaires qui testent la position relative de deux objets dans l'espace (*disjoint*, *touches*, *overlaps*, *inAdjacent*, *within/contains*, *crosses* et *equals*), des *opérateurs ensemblistes*, qui traitent l'espace comme un *ensemble* de régions (*union*, *intersection*, *symmetricalDifference* et *difference*) et des *opérateurs de mesure* (*dimension* et *distance*). En conclusion, AROM-ST permet de modéliser et d'exploiter la dimension spatiale des concepts définis dans ses bases de connaissances.

Dans de récents travaux (Miron *et al.*, 2007), nous avons aussi étudié l'ouverture d'AROM vers le Web sémantique, en tentant de le rapprocher de OWL. Cette étude comparative a été menée selon trois axes : *représentation*, *typage*, et *inférences*. Après avoir conclu à un déficit d'AROM vis-à-vis de OWL sur le plan de la puissance de description, nous avons proposé une extension de son méta-modèle visant à réduire cet écart (Miron *et al.*, 2007). Le nouveau méta-modèle AROM-ONTO intègre de nouvelles structures de représentation qui rapprochent sensiblement AROM de OWL. Enfin, nous avons positionné AROM en termes de typage et d'inférences vis-à-vis de OWL, en constatant et en revendiquant une certaine complémentarité.

ONTOAST (pour ONTOlogies en Arom-ST) est une extension d'AROM qui englobe les types et opérateurs spatiaux et temporels gérés par AROM-ST, ainsi que les modifications du noyau apportées par AROM-ONTO pour assurer la compatibilité avec OWL (voir Fig.1). ONTOAST sera, en conséquence, un outil pour la modélisation et l'exploitation d'ontologies spatio-temporelles qui formera une passerelle entre le domaine des SIG et le Web sémantique. Par la suite, nous proposons de munir ONTOAST de relations spatiales qualitatives visant à compléter

<sup>1</sup> OpenGIS home page: OpenGIS Consortium, OpenGIS Reference Model, 2004, <http://www.opengis.org/>.

les connaissances sur les objets modélisés et à permettre une formulation de requêtes plus souple. Ces relations qualitatives pourront être modélisées de manière textuelle dans la syntaxe d'AROM-ONTO, ou graphiquement à travers une prochaine version de l'environnement interactif (IDE). Notre but est de répondre à des requêtes spatiales sans qu'il soit nécessaire de pré-calculer et de stocker toutes les relations possibles entre les entités.

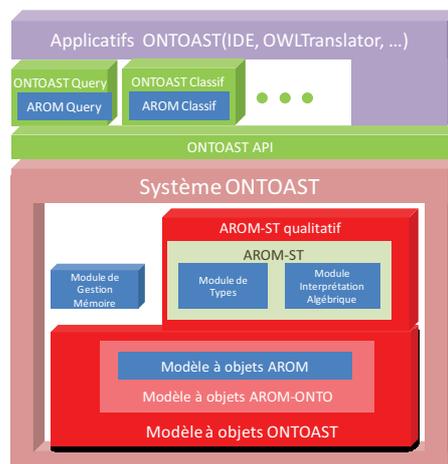


Fig. 1 – L’extension qualitative des modules AROM-ST et AROM-ONTO et l’organisation modulaire de la plate-forme ONTOAST

#### 4. Relations spatiales qualitatives en ONTOAST

Notre proposition cherche à faire coexister à la fois des données spatiales quantitatives sous la forme de géométries exactes d’objets et des données imprécises sous la forme de relations qualitatives. Nous prenons en compte trois catégories de relations qualitatives : *topologiques*, *d’orientation* et *de distance*, qui peuvent être issues d’une déclaration explicite de la part de l’utilisateur ou bien inférées à partir des informations existantes. En conséquence, lors d’une requête utilisateur, la réponse sera construite : *i*) en utilisant les connaissances explicites, dans le cas où la relation cherchée est stockée dans la base, *ii*) en réalisant des inférences qualitatives à partir des relations explicites existant entre les objets, *iii*) en utilisant des méthodes numériques d’estimation et de calcul pour déduire des relations qualitatives impliquant les individus visés par la requête, et *iv*) en appliquant des raisonnements qualitatifs autant sur les relations spatiales descriptives existantes que sur celles déduites par l’intermédiaire des calculs géométriques. Les sections 4.1, 4.2 et 4.3 présentent des exemples pour chacune de ces quatre situations.

Pour intégrer les relations spatiales qualitatives en ONTOAST, nous avons créé un nouveau type d’association, *AromQSA* (AromQualitativeSpatialAssociation), qui regroupe les trois catégories de relations que nous proposons: *Topology*, *Distance* et *Direction*. Une association qualitative spatiale est une association dont les rôles désignent des objets spatiaux, même si on ne dispose pas de leurs géométries exactes. L’association topologique (*Topology*) est une association binaire, qui possède huit sous-classes (*Disjoint*, *Intersects*, *Touche*, *Within*, *Crosses*, *Overlaps*, *Contains* et *Equals*), définissant le comportement de chaque relation topologique qui peut exister entre deux régions. L’association *Distance* modélise des relations ternaires qui

contiennent, en plus des objets décrits, un objet de référence. Nous avons considéré l'existence de trois relations de proximité : *Closer*, *Farther* et *Equidistant*, qui décrivent respectivement le fait qu'un objet *A* est plus proche, plus loin ou approximativement (avec une marge d'erreur établie au préalable) à la même distance qu'un objet *B*, par rapport à un troisième objet *C*. Pour représenter les relations de direction, nous avons eu recours à des associations binaires (sous-associations de *Direction*) qui modélisent les neuf positions cardinales (*N*, *NE*, *E*, *SE*, *S*, *SW*, *W*, *NW*) dans lesquelles un objet spatial *A* peut se trouver par rapport à un autre objet *B*. Les sections suivantes illustrent chacune de ces associations et présentent leur comportement.

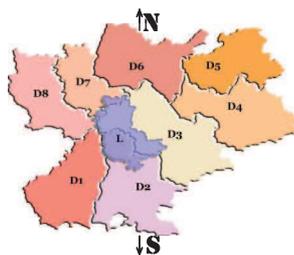


Fig. 2 – Exemple de situation réelle qui sera modélisée à l'aide de ONTOAST.

Pour illustrer l'utilisation d'ONTOAST, nous considérons un exemple simple mais concret qui modélise un ensemble de 8 *Départements* ( $D_1$  à  $D_8$ ) et le *Lac* *L* de la Figure 2. Nous disposons donc de leurs coordonnées géographiques et de leurs géométries exactes (contours modélisés par des polygones). Nous considérons aussi l'existence d'un autre département  $D_9$ , pour lequel nous ne disposons pas de données géométriques, et qui est lié au modèle par l'intermédiaire d'un ensemble de relations spatiales qualitatives explicitées par l'utilisateur. Les trois sous-sections qui suivent montrent les requêtes qui peuvent être résolues en utilisant les raisonnements offerts par le module ONTOAST.

L'ontologie ONTOAST qui modélise les entités de la Figure 2 est illustrée par la Figure 3. On identifie les concepts *Département* et *Lac*, tous deux étant des objets spatiaux (ayant au moins un attribut dont le type est un type spatial). On observe aussi les associations qualitatives *West* de type *Direction* et *Disjoint* de type *Topology*. Au niveau des instances, l'ontologie contient les huit départements pour lesquels nous connaissons les géométries, le département  $D_9$  dont on sait qu'il est situé à l'Ouest de  $D_1$ , ainsi que le lac *L* dont on sait qu'il n'appartient pas au  $D_4$ , ni il en est limitrophe.

#### 4.1. Relations topologiques

Les relations topologiques considérées par notre approche sont des relations binaires entre des objets possédant une dimension spatiale, mutuellement disjointes et qui traduisent, en termes d'associations, les opérateurs topologiques d'AROM-ST introduit dans la section 3. Les opérateurs topologiques du LMA seront utilisés pour la construction des associations topologiques à partir de l'information géométrique associée aux individus. Nous illustrons notre démarche, en reprenant l'exemple de la Figure 3. Supposons que l'objectif soit de répondre aux deux questions suivantes : *i*) quelles sont les relations topologiques entre le lac *L* et les départements  $D_1$  et  $D_4$ ?, *ii*) quels sont les départements sur lesquels s'étend le lac *L*? Le premier pas dans le raisonnement est de vérifier si l'association recherchée se trouve ou non dans le modèle. Dans le cas de la question *i*), la base de connaissances contient la définition

explicite de la relation *Disjoint* entre *L* et *D<sub>4</sub>*. Dans cette situation, le raisonnement s'arrête ici car on a trouvé la réponse. Cependant, entre *L* et *D<sub>1</sub>* il n'existe pas de relation topologique explicitement définie. La question de la topologie reliant *D<sub>1</sub>* et *L* sera résolue en faisant appel aux opérateurs du LMA, car les géométries des deux objets sont connues. Le système déduit le type du tuple topologique impliquant *D<sub>1</sub>* et *L* par des questions successives (*D<sub>1</sub> disjoint L ?*, *D<sub>1</sub> touches L ?*, *D<sub>1</sub> overlaps L*). De façon analogue le système déduit dans quels départements se situe le lac *L*.

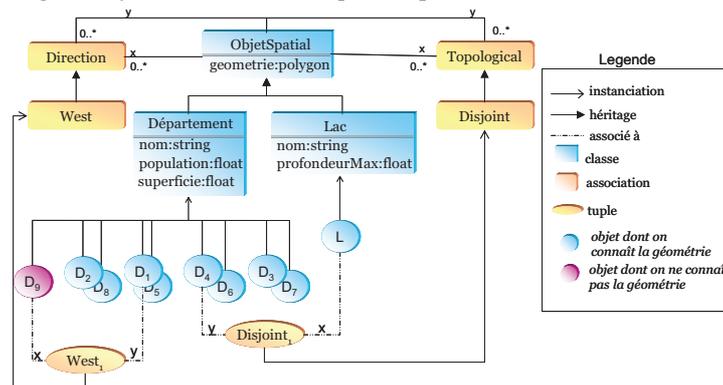


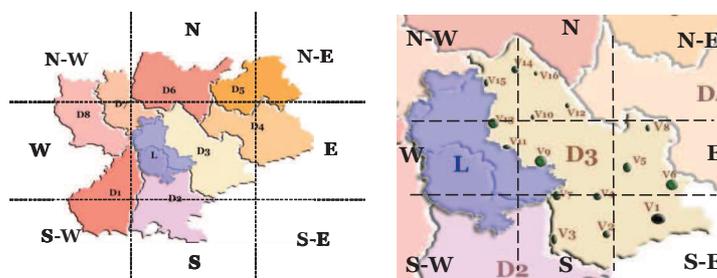
Fig. 3 – Modèle ONTOAST d'un domaine spatial

La cohérence de la base sera vérifiée lors de l'ajout, par déclaration explicite, de nouvelles associations. Pour gérer les conflits, nous prenons en compte, d'un côté, le principe selon lequel les géométries priment dans le raisonnement, et, de l'autre, la confiance dans la source des données. Ainsi, le système connaissant les géométries des départements *D<sub>1</sub>* et *D<sub>2</sub>* (Figure 3), une déclaration qui définirait entre eux la relation *Crosses* sera rejetée en raison du conflit créé avec la relation *Touches* issue du calcul.

#### 4.2. Relations de direction

Afin de caractériser la position d'un objet spatial par rapport à un autre, l'être humain utilise des catégories qualitatives, telles que: « à droite du bâtiment », « au nord de Paris », « au sud de la France », « sur la table », etc. en fonction des dimensions de l'univers du discours. Si le positionnement se fait dans un espace limité ou très limité (qui peut être observé en totalité par l'œil humain sans être obligé de changer le point de vue), les catégories qualitatives employées le plus fréquemment sont : « en face », « derrière », « à gauche », et « à droite », rapportées soit à la position de l'observateur, soit à un objet de référence. Dans des espaces plus larges, comme en géographie, pour modéliser de manière équivalente l'orientation on utilise les directions cardinales (N, S, E et W). Une correspondance peut être définie entre les deux ensembles, car ils traduisent les mêmes relations mais dans des contextes différents. L'orientation est donc vue comme une relation ternaire, impliquant l'objet caractérisé, l'objet de référence et le système de référence. En fonction du contexte, (Frank, 1996) propose, pour définir l'orientation, le partitionnement de l'espace en huit régions ou plus. L'idée est de considérer des objets représentés par des points et de partitionner l'espace en plusieurs zones qui correspondent aux différentes classes de direction (*north*, *southeast*, *east*, *northwest*, etc.) et dont l'origine est l'objet de référence. Cependant, le problème devient plus complexe si l'on considère des objets étendus représentés par des polygones

(Egenhofer *et al.*, 2003) non seulement parce qu'ils peuvent croiser plusieurs secteurs lors d'une comparaison avec d'autres objets étendus (Figure 4.a), mais aussi parce qu'ils disposent de directions intrinsèques (par exemple on parle du *nord d'un département*) (Figure 4.b). Dans la suite de cette section, nous adoptons la représentation spatiale basée sur des régions et non sur des points.



**Fig. 4**— Exemple de relations d'orientation : a) l'objet à caractériser ( $D_1$ ) croise plusieurs directions (*Ouest*, *Sud-Ouest*, et *Sud*) par rapport à l'objet de référence ( $D_3$ ); b) l'objet caractérisé ( $D_3$ ) contient des directions intrinsèques.

A l'aide du *footprint* (rectangle minimal qui inclut l'ensemble des polygones définissant la géométrie d'un objet) de la région de référence ( $D_3$  dans l'exemple choisi) nous partitionnons l'espace comme montré par la Figure 4.a. Pour chacun des neuf secteurs obtenus, il est possible de vérifier dans quelle mesure l'objet à caractériser  $D_1$  y est inclus. Pour faciliter les calculs quantitatifs et pour bien séparer les niveaux de raisonnement, nous avons introduit neuf nouveaux opérateurs dans le LMA : un opérateur pour chaque direction considérée (N, NE, E, SE, S, SW, W, NW) et un pour le secteur central (C). Chacun de ces opérateurs calcule le pourcentage de la région à caractériser (dans notre cas  $D_1$ ) qui se trouve dans la direction considérée par rapport à la région de référence ( $D_3$ ) (e.g.  $D_1$  est 3% au *Sud* de  $D_3$ , 46% au *Sud-Ouest* de  $D_3$  et 39% à l'*Ouest* de  $D_3$ ). Finalement, pour décider de la relation résultat, on choisit soit la direction correspondant au pourcentage le plus grand si l'écart dépasse une limite prédéfinie (ex. 45%), soit on utilise le tableau de composition proposé par (Frank, 1996), en prenant les deux relations dominantes comme entrées. Pour l'exemple choisi, en composant les directions dominantes W et S-W, nous obtenons la direction générale S-W.

Concernant les relations de direction, une autre question consiste à situer un département (e.g.  $D_9$ ) vis-à-vis des autres départements. Comme la seule liaison entre  $D_9$  et le modèle est le fait que  $D_9$  soit à l'Ouest de  $D_1$  (cf. Figure 3), le système vérifiera (dans la mesure de l'existence des informations) où se trouve  $D_1$  par rapport aux départements et essaiera de déduire la position réelle de  $D_9$ , en faisant appel à la composition de relations cardinales. Pour l'exemple de la Figure 4.a, nous savons que  $D_1$  se trouve au *Sud-Ouest* de  $D_3$  et que  $D_9$  est à l'*Ouest* de  $D_1$ . Donc  $D_9$  sera au *Sud-Ouest* de  $D_3$ . Cependant, une telle requête peut parfois rester sans réponse. Si l'on considère l'existence d'un autre département  $D_{10}$ , situé à l'Ouest de  $D_1$ , et pour lequel nous ne disposons pas de sa géométrie, il est difficile, voire impossible, de déduire la position qu'il occupe par rapport à  $D_9$  sans faire appel à des informations supplémentaires. Même si le système déduit le fait que  $D_{10}$  et  $D_9$  sont dans la même direction par rapport à  $D_1$ , il est difficile de savoir si  $D_9$  est à l'Ouest de  $D_{10}$  ou bien à l'Est. Pour répondre à des requêtes comme : « *Quelles sont les villes du Sud du département?* » qui considèrent autant la relation d'inclusion que celle de direction, nous proposons le raisonnement illustré par la Figure 4.b. Il s'agit de déterminer le

rectangle minimal (R) qui englobe le département, de le diviser en neuf secteurs égaux et d'appliquer le raisonnement de déduction de la direction en considérant le secteur central comme nouvel objet de référence. Pour cet exemple nous déduisons, que les villes du Sud du département  $D_3$  sont :  $V_3$  et  $V_2$ .

### 4.3. Relations de distance

Lorsque l'on s'intéresse à une relation de distance, la première précision à apporter concerne la définition même de la distance étudiée. S'intéresse-t-on à une distance à vol d'oiseau, au plus court chemin dans un réseau routier, ou encore au temps d'accès en fonction d'un moyen de transport déterminé (train, voiture, etc.) ? De plus, il existe deux catégories de relations qualitatives qui visent la distance entre objets : les distances *absolues* et *relatives*. Dans le premier cas, on utilise une échelle qui divise la ligne des réels en plusieurs secteurs (par exemple : *very close*, *close*, *equidistant*, *far* et *very far*) en fonction de la granularité souhaitée. La valeur numérique obtenue lors du calcul numérique de la distance est alors positionnée sur cette échelle (i.e. dans un secteur) pour donner une distance absolue. La distance relative est obtenue suite à une comparaison entre la distance considérée et une distance de référence et aboutit à une relation ternaire telle que *closer than*, *equidistant* ou *farther than*.

Raisonnement avec les distances qualitatives conduit parfois à des difficultés impossibles à surmonter. C'est le cas d'une séquence linéaire de points  $p_1, p_2, \dots, p_n$  chacun étant impliqué dans une relation *close* avec son prédécesseur, et pour lesquels on ne peut pas dire à partir de quelle valeur de  $n$ , il est correct de dire que  $p_n$  est lié par une relation *far* à  $p_1$ . Par ailleurs, pour combiner les distances, il ne suffit pas de disposer des distances en elles-mêmes. Il faut aussi les positions des points impliqués. Par exemple, si le point  $B$  est loin du point  $A$  et si  $C$  est loin de  $B$  alors  $A$  et  $C$  peuvent soit être très loin l'un de l'autre si les trois points sont alignés, soit, au contraire, être proches si l'angle décrit par les segments  $AB$  et  $BC$  est petit.



Fig. 5– Le réseau de transport du département  $D_3$ .

Pour définir des distances absolues en utilisant le formalisme qualitatif on doit se rapporter au contexte des déclarations. Plus concrètement, une relation *close* définie pour deux villes n'est pas équivalente à une relation *close* valable pour deux pays. Pour le moment, on se limite à exprimer en ONTOAST des distances relatives simples, comme : «  $V_3$  est plus loin de  $V_1$  que l'est  $V_2$  » (Figure 5). Pour cela, si on dispose des géométries des objets, on fait appel à l'opérateur *distance* du LMA pour calculer (selon le système métrique choisi) les deux distances numériques qui seront comparées pour obtenir le type d'association valable. En comparant, par exemple dans le réseau de transport illustré dans la Figure 5, les distances entre la ville  $V_3$  et les villes  $V_2$  et  $V_4$ , nous obtenons un chemin plus long de  $V_3$  à  $V_2$  que celui de  $V_3$  à  $V_4$  même si, physiquement,  $V_3$  est plus proche de  $V_2$ .

## 5. Conclusion et perspectives

Cet article a présenté ONTOAST, une extension du Système de Représentation de Connaissances par Objets AROM dédiée à la modélisation et l'exploitation d'ontologies géographiques pour le Web sémantique. ONTOAST repose sur l'intégration de deux modules d'AROM : AROM-ST qui permet la gestion de types spatiaux et temporels dans des bases de connaissances, et AROM-ONTO qui rapproche les capacités de représentation d'AROM de celles du standard OWL. ONTOAST devient, de fait, un outil de description et d'interrogation d'ontologies géographiques destinées au Web sémantique. De plus, afin de permettre une formulation de requêtes plus proche du langage usuel dans sa flexibilité, nous avons intégré dans le méta-modèle de ONTOAST trois nouveaux types de relations spatiales qualitatives : *topologie*, *orientation* et *distance*. Il devient dès lors possible de mener des raisonnements qualitatifs sur des ontologies géographiques gérées par ONTOAST. Il n'en reste pas moins que le chemin est encore long pour accéder à des raisonnements spatiaux complexes. A ces fins, nos recherches s'orientent vers de nouveaux moyens d'analyse et d'interrogation sémantiques des informations contenues dans de telles ontologies. En perspective, nous envisageons la mise en œuvre d'une analytique sémantique dans ONTOAST, permettant de découvrir les connexions intrinsèques et implicites contenues dans une base d'objets spatiaux.

## Références

- Berners-Lee T. & Hendler J.A. & Lassila O. (2001). The Semantic Web . *Scientific American*, 284(5), pp. 34-43.
- Capponi C. (1995). Identification et exploitation des types dans un modèle de connaissances à objets. *Thèse de l'Université Joseph Fourier, Laboratoire Lijia/Imag*.
- Cohn A. & Hazarika S. M. (2001). Qualitative Spatial Representation and Reasoning: An Overview. *Fundamenta Informaticae* 46, IOS Press, pp. 1-29.
- Dolbear C. & Hart G. (2006). So what's so special about spatial? *Ordnance Survey of Great Britain*.
- Egenhofer M. & Rodriguez M. A. & Blaser A. (2003). Query Pre-processing of Topological Constraints: Comparing a Composition-Based with Neighborhood-Based Approach. *SSTD*, pp. 362-379.
- Frank A.U. (1996). Qualitative Spatial Reasoning: Cardinal Directions as an Example. *International Journal of Geographical Information Science*, vol 10, nr 3, pp 269-290.
- Gruber T. R. (1993). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer Academic Publishers.
- Katz Y. & Grau B. C. (2006). Representing Qualitative Spatial Information in OWL-DL. *Proceedings of OWL: Experiences and Directions*.
- McGuinness D. & van Harmelen F. (2005). OWL Web Ontology Language –Overview. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>
- Miron A. & Capponi C. & Gensel J. & Villanova-Oliver M. & Ziébelin D. & Genoud P. (2007). Rapprocher AROM de OWL... *LMO*, Toulouse, France.
- Moisuc B. & Gensel J. & Martin H. (2004). Représentation de connaissances par objets pour les SIG à représentations multiples. *CASSINI-SIGMA, Géomatique et Analyse Spatiale*, Grenoble.
- Page M. & Gensel J. & Capponi C. & Bruley C. & Genoud P. & Ziébelin D. (2000) Représentation de connaissances au moyen de classes et d'associations : le système AROM. *LMO*, pp. 91-106, Mont Saint-Hilaire, Canada.