représente sous une granularité plus faible. Une occurrence temporelle est aussi un objet qui est visible différemment suivant les points de vue (il peut être invisible, visible comme un instant à l'aide de son attribut `position` ou visible comme un intervalle avec les attributs `debut` et `fin`). Il est clair qu'implémenter la granularité à l'aide des points de vue est une déviation de la destination des points de vue: il n'y a pas équivalence entre point de vue et granularité donc pas de raison de les joindre. Ce n'est ici qu'une facilité d'implémentation.

Ainsi, sur l'exemple du courrier évoqué plus haut, trois évènements sont dans un point de vue: «aller chercher le courrier», «aller au marché» et «arrivée du facteur» dont l'occurrence temporelle est un intervalle. Les deux premiers objets sont associés par une contrainte portant la valeurs {b m}. Sous une autre granularité, «aller chercher le courrier» peut être représenté par quatre évènements «aller à la boîte», «ouvrir la boîte», «prendre la lettre» et «fermer la boîte» successivement liés par la même relation.

Une contrainte est représentée par un objet contenant deux attributs pour les occurrences temporelles en relation et une liste de symboles représentant les relations. Ces relations sont localement réduites à l'aide d'un algorithme classique de propagation de contraintes temporelles [ALL 83]. Ainsi, la réponse à la question «quelle est la relation entre «aller à la boîte» et «fermer la boîte»?» s'obtient en créant un objet relation entre ces deux évènements dont la liste des symboles est l'ensemble des 13 relations et en demandant au propagateur de contraintes de réduire ces contraintes. Le propagateur est alors capable de retourner la réponse {b}.

Sur un unique point de vue auquel est associé une granularité, la résolution des contraintes temporelles fonctionne donc comme dans un système de représentation de connaissance classique. L'intérêt du système réside, bien sûr, dans la collaboration entre les granularités. Cette collaboration se fait au travers des histoires.

6.2. Traitement des histoires

Une *histoire* permet d'exprimer la conversion d'un évènement sous une granularité en un ensemble d'évènements, sous une granularité plus fine, auquel correspond un ensemble d'expressions temporelles et de contraintes. Une histoire est implémentée par un objet contenant une liste d'évènements et une liste de relations entre ces évènements. Dans l'exemple du courrier, l'histoire correspondant à «aller chercher le courrier» contient l'ensemble des évènements le décomposant ainsi que les relations exprimées entre eux.

Les contraintes peuvent être résolues à un certain niveau de granularité en considérant les histoires comme des expressions temporelles simples. Puis, il est possible de passer à une granularité inférieure et de résoudre les contraintes internes à une histoire précise. Cela permet de répondre à une requête se plaçant à un niveau précis sans résoudre les contraintes à un niveau inférieur. Il est possible de hiérarchiser cette propagation et donc de tirer parti de la structure hiérarchique des différents grains. Ainsi, le système ne redescendra pas à une

granularité très fine si une contrainte peut se résoudre à la granularité courante (voir [BON 88]). La granularité devient alors une heuristique pour traiter les contraintes temporelles.

Le système est capable de transposer un objet représentant un évènement d'une granularité vers une autre en utilisant la décomposition de cet évènement en histoire. Le propagateur utilise alors cette décomposition pour opérer ces déductions. Il exploite les propriétés établies pour la propagation de contraintes entre différentes granularités: le propagateur de contraintes est classique mais convertit les objets temporels dans la plus haute granularité commune aux objets à comparer afin de procéder à la propagation de contraintes. Le résultat peut, ensuite, être descendu si nécessaire. Ceci assure à la propagation de contraintes le maximum de précision puisque les contraintes sont montées avant d'être propagées puis propagées avant d'être descendues. Ceci est tout à fait en accord avec les résultats obtenus dans la section précédente. En réalité, l'algorithme ne raisonne que sur les histoires: la remontée se fait d'histoire en histoire, en résolvant à chaque étape l'ensemble des contraintes locales à l'histoire courante avant de remonter les contraintes existant entre l'élément à remonter et l'histoire qui l'englobe.

Ainsi, dans le cas du récit présenté à la fin du paragraphe précédent, il est possible de déduire assez simplement que le facteur est arrivé avant que la personne ne soit allée au marché: en effet, le moment d'arrivée du facteur est celui d'ouverture de la boîte à lettres, or celui-ci est inclus dans le moment d'«aller chercher le courrier» qui précède l'«aller au marché».

Dans la réalité, pour résoudre ce problème, le système a besoin de connaissances linguistiques (qui ne sont ici que simulées). Ce raisonnement, nommé *identification d'évènement*, permet de replacer l'évènement concernant «l'arrivée du facteur» par rapport aux autres évènements. Pour cela, la seule information disponible est que cet évènement est simultané à l'évènement «fermer la boîte à lettres». Deux problèmes se posent: le second évènement n'est pas apparu dans le cours du texte et le niveau de granularité courant est celui d'«aller chercher le courrier». L'algorithme à utiliser pour résoudre ces références est globalement le suivant:

(1) Essayer d'identifier l'évènement avec les évènements de la granularité courante.
(2) En cas d'échec de (1), décomposer les évènements de la granularité courante à l'aide de leur histoire, passer à la granularité inférieure et réappliquer (1).

Bien entendu, cette ébauche de procédure n'est pas complète, elle fait l'impasse sur la possible recherche dans des granularités plus élevées (voir [GAY 91] pour un exemple d'algorithme plus détaillé) et elle ne fait pas usage de connaissance pragmatique pour restreindre l'espace de recherche. Elle permet cependant d'identifier l'évènement «fermer la boîte à lettres».

Poser la contrainte de simultanéité entre «fermer la boîte à lettres» et «arrivée du facteur» ne peut se faire qu'entre entités de même granularité. L'évènement «arrivée du facteur» est donc descendu sous la granularité inférieure. La propagation de contraintes temporelles va ensuite permettre d'établir, entre autre, la relation existante entre l'«arrivée du facteur» et l'évènement

«aller chercher le courrier»: {d}. Cela est permis parce que l'histoire lie toujours ses bornes avec celles de l'évènement de niveau plus élevé qu'elle représente. Par contre, les contraintes existant entre l'évènement plus élevé et les autres évènements de granularité supérieure ne sont pas descendues. Cette relation est remontée au niveau de la requête et est préservée. La propagation de contraintes au niveau supérieur peut alors jouer et déduire par transitivité que {d}×{b m} donne {b} et que par conséquent: «le facteur est arrivé avant que j'aille au marché».

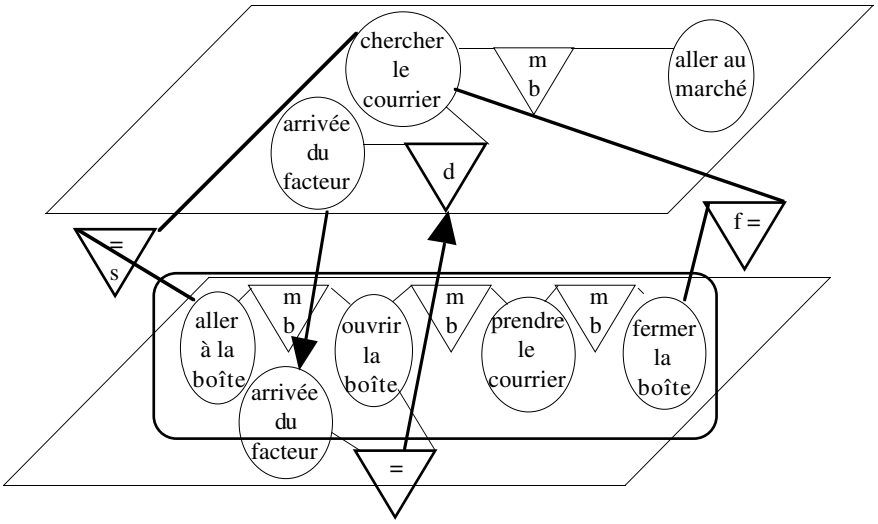


Figure 6. Les deux granularités impliquées dans la compréhension du récit sont présentées ci-dessus. En gras, figurent les trois étapes de la résolution de contraintes: Le rectangle représente la transformation d'aller «chercher le courrier» (C) en une histoire plus complexe. La flèche descendante représente la descente de l'évènement «arrivée du facteur» (F). La flèche montante représente la montée de la relation entre F et C qui permettra de situer F au niveau supérieur et d'en déduire sa position par rapport à «aller au marché».

Dans le même ordre d'idées, la recherche en résolution de contraintes temporelles prône l'identification de petits ensembles de contraintes connexes pour localiser la propagation. C'est l'exploitation du concept de localité évoqué en introduisant les *intervalles de référence* [ALL 81, KOO 89]. Il s'agit en fait d'un ensemble d'intervalles dont toutes les relations sont calculées en référence à un intervalle précis et qui n'ont de relation avec les autres intervalles que par le biais de leur intervalle de référence.

L'histoire permet de définir des intervalles de référence en les dotant d'une sémantique pertinente vis-à-vis du domaine d'application et non plus du seul processus de propagation de contraintes. À cet égard, les histoires sont à mi-chemin entre les intervalles de référence et les *scripts* [SCH 82]. D'autre part, les composants d'histoires peuvent être utilisés de manière paresseuse, c'est-à-dire qu'ils existent implicitement mais qu'il est inutile de les créer tant qu'ils ne sont pas utilisés. Ainsi, c'est l'algorithme d'identification d'évènements qui crée les éléments d'histoire lors de sa recherche.

6.3. Développements

L'implémentation de la représentation temporelle sur la première maquette de TROPES a permis d'expérimenter le bien fondé de l'approche suivie. Elle n'est qu'un banc d'essais pour la notion d'histoire et ne prétends pas être utilisable en l'état. En particulier, il y manque:

— Les aspects numériques: y lier les aspects numériques nécessite le développement d'un gestionnaire général du temps de type TMM ("Time map manager" [DEA 87, GHA 89]) permettant de stocker et d'inférer efficacement les informations à la fois numériques et symboliques.

— L'identification automatique des évènements: l'instanciation des histoires doit être déclenchée par l'utilisateur. Les histoires elles-mêmes sont créées à l'aide d'une procédure fournie par l'utilisateur (qui crée les évènements et pose les contraintes) au lieu d'être exprimées dans un langage particulier.

— Des algorithmes complets vis-à-vis de la recherche dans les granularité plus élevées [GAY 91].

Par ailleurs, il apparaît, au cours de l'exposé, que la notion de décomposition est importante dans la gestion des changements de granularités. Si cette notion n'est pas abordée d'emblée et que l'exposé en reste à des notions classiques de représentation du temps, c'est que la représentation de la composition dépasse largement le cadre de la représentation du temps et n'a pas encore reçu de réponse satisfaisante. Il semble que la voie la plus prometteuse dans la liaison entre points de vue et granularité consiste à considérer la liaison entre une expression temporelle et son histoire comme un lien de composition de TROPES [MAR 93] (une constatation similaire a été exploitée dans [CIA 93b]).

7. Granularité spatiale

Les aspects caractéristiques du temps (l'irréversibilité en particulier) n'ont pas été pris en considération ici. L'exploitation du temps ne concerne que des points ou des segments de droites sur une droite. Il devrait donc être possible d'étendre la granularité à d'autres représentations dont la droite est aussi le modèle (séquence génomique par exemple [GUI 93, SCH 94]) ainsi qu'à d'autres modèles à plusieurs dimensions. Ainsi, les solutions proposées devraient s'appliquer à la granularité de n'importe quel type de représentations. À l'instar du raisonnement temporel, le raisonnement sur l'espace semble être un champ d'utilisation de la granularité.

Par exemple, la carte Didier Richard consacrée au Vercors n'est pas une simple carte au 1/50 000e: elle contient aussi sur sa couverture une carte au 1/1 000 000e et un schéma de la région Rhône-Alpes au 1/4 000 000e sur laquelle se trouvent les contours des autres cartes de la collection. Ce type de disposition est tout d'abord pratique (voir figure 7), mais il pose aussi beaucoup de problèmes aux cartographes. Ce sont les problèmes rencontrés lors de la

construction numérique de la granularité: où se trouve le centre des villes lorsqu'elles doivent être représentées par un point à une échelle plus petite, quels éléments doivent disparaître (la seconde carte retient les villes principales, les autoroutes, les nationales et les fleuves alors que la dernière se cantonne aux grandes villes et fleuves)?

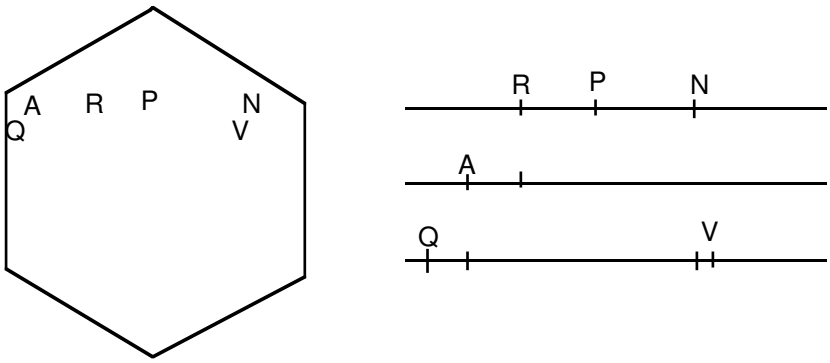


Figure 7. Pour se rendre de Quiberon à Villiers-lès-Nancy, il n'est pas nécessaire d'utiliser une carte très précise afin de déduire qu'il faut se déplacer vers l'est. Les contraintes à résoudre pour se déplacer de Quiberon à Villiers-lès-Nancy se résolvent tout d'abord en assimilant, sous une granularité grossière, Quiberon à Auray, Auray à Rennes et Villiers-lès-Nancy à Nancy. Rennes étant à l'ouest de Paris et Nancy à l'est de Paris, il va falloir se rendre de Rennes à Paris puis de Paris à Nancy. En raffinant, Quiberon dépendant ferroviairement d'Auray, il va donc falloir se rendre d'Auray à Rennes. Plus précisément, il faudra aussi se rendre de Quiberon à Auray et de Nancy à Villiers-lès-Nancy, ce qui peut nécessiter un plan d'une précision importante.

L'exemple précédent ne recouvre qu'une modification d'échelle (de type numérique). Mais la représentation granulaire de l'espace met en évidence de nombreux découpages irréguliers: ainsi en est-il des découpages administratifs français en régions, départements, cantons, communes et arrondissements qui «en principe» se décomposent en une hiérarchie d'inclusion [BUI 90]. Mais qu'advient-il si l'on y superpose les circonscriptions académiques, militaires ou téléphoniques? On y retrouve le même genre de difficultés que lors d'assemblage de cartes génomiques diverses obtenues à des résolutions différentes à l'aide de techniques différentes: même une propriété aussi capitale que le non croisement ne peut plus être garantie [GUI 93].

L'intuition introduite pour la représentation du temps devrait cependant fonctionner dans le cas de l'espace. Il faut toutefois élaborer, en fonction de la structure de la représentation considérée, les règles qui gouvernent la conversion d'une granularité vers une autre. La structure est gouvernée par les relations qu'entretiennent entre elles les entités représentées. À chaque système de relations devra correspondre des opérateurs de conversion.

7.1. Espace, espaces

Plutôt que de donner un système de représentation symbolique pour l'espace (plusieurs sont actuellement en concurrence) il faut se demander quel type de représentation permet de généraliser le travail présenté ici. Les caractéristiques de l'espace par rapport au temps sont les suivantes:

— les mouvements qui s'y passent sont différents des mouvements dans le temps;

— l'espace se décompose suivant plusieurs dimensions;

— l'espace ne définit pas facilement un ordre sur ses régions.

La première caractéristique n'a pas été prise en compte ci-dessus. Elle ne nécessitera pas d'aménagements supplémentaire. La seconde n'a aucune incidence sur la relation symbolique du temps fondée sur des critères purement topologiques (connexité). Elle a, par contre, une certaine incidence sur la représentation numérique et sa granularisation. Elle conditionne aussi la troisième en permettant d'introduire la notion d'ordre liée à des notions vectorielles. La troisième caractéristique est reliée à une structure vectorielle de l'espace (repère, direction). Cette notion d'ordre est apparente dans les relations symboliques (avant, après).

Les deux notions principales à discuter ici concernent donc les notions de dimensions (et d'interprétation des régions dans un espace à plusieurs dimensions) et la notion d'ordre.

L'interprétation usuelle de l'espace est $\mathbb{R}^n$. Cette interprétation s'accorde correctement avec la notion de point. Reste à interpréter la notion de région dans cet espace. Une région peut être considérée comme un ensemble de points de l'espace. En fonction de structures attribuées à cet espace, il est possible d'exprimer différentes propriétés des régions: connexité (espace topologique: m, m^{-1}…), distance (espace métrique), ordre (espace vectoriel: b, b^{-1}…).

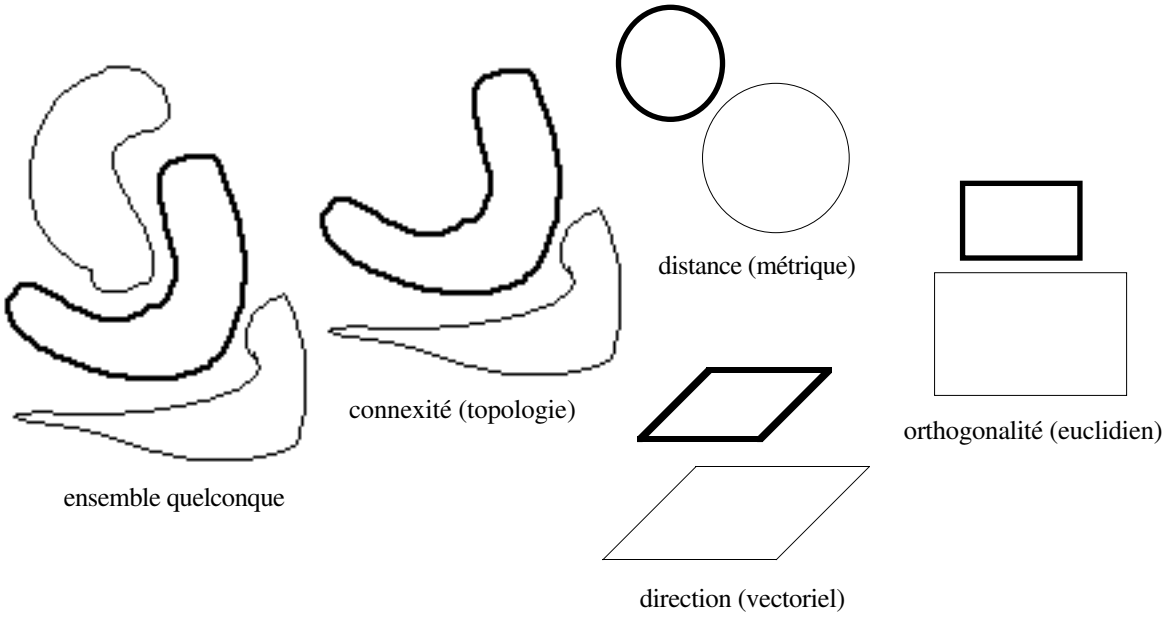


Figure 8 (inspirée de [BUI 90]). Un exemple des types de régions manipulables à l'aide des notions introduites par les différent espaces mathématiques.

Représenter les régions par des entités géométriques définies peut se faire de différentes manières: intervalles (voir plus bas), arbres 2^n-aires (généralisation en dimension n des "quadtrees" ou arbres quaternaires [SAM 84]) ou simplexes [EGE 90]. Ces représentations supportent en général des opérations d'intersection, d'union, de localisation de manière efficaces et elles permettent de représenter numériquement les régions sous une forme compacte. Elles supportent assez bien la notion de granularité numérique (les quadtrees par

exemple sont réputés constituer une hiérarchie de modèles de l'espace qui est alors infiniment discrétisé) car ces figures se décomposent naturellement en un assemblage de figures de même nature. Seule la notion d'intervalle entre deux points de l'espace (qui généralise la notion d'intervalle temporel) est considérée ici. Elle utilise la notion d'espace vectoriel et est développée plus longuement dans l'annexe C car elle permet simplement d'hériter les propriétés définies pour les intervalles temporels grâce à la simple notion de produit.

Cependant, une région spatiale réduite à une figure régulière n'est pas très adaptée à certaines utilisations. Son utilisation presque exclusive est celle qui permet des calculs sur la *boite englobante* d'une région ou, au contraire, la décomposition, toujours plus approchée d'une région particulière en un ensemble de régions régulières disjointes. L'intervalle est donc peu pertinent quand la dimension est supérieure à 1.

7.2. Une représentation symbolique de l'espace

Il est possible de songer à une structure plus souple pour la région en fonction des propriétés qui leur sont affectées (voir figure 9). Ces propriétés sont discutées ci-dessous avant d'aborder une représentation des relations entre régions qui s'inspire de celle des intervalles temporels en autorisant une grande variété de régions.

L'«algèbre d'intervalles» utilisée par James Allen ne prend pas en compte la dimension de l'espace et la notion d'intervalles n'est présente que dans l'interprétation qui est faite de ces deux relations (en particulier, les problèmes apparaissant dès que l'on désire s'intéresser aux bornes des intervalles ont poussé Allen à ne jamais les considérer). Cette algèbre peut être appauvrie à loisir (en considérant un sous-ensemble de relations qui groupe certaines relations entre elles, voir figure 9).

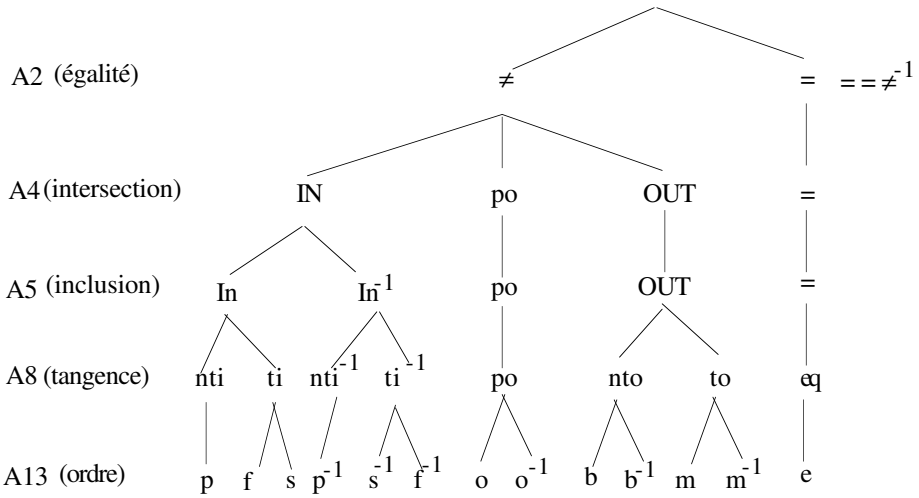


Figure 9. Hiérarchie d'algèbres d'intervalles. Toutes les relations sont leur propre réciproque sauf si mentionné ($^{-1}$). Il y a d'autres systèmes possibles (par exemple, ici, A_3 présenté au §2.2.1, n'est pas représenté car il est constitué à partir de A_2 et de la notion d'ordre), l'ensemble des systèmes possibles se décompose plutôt en un graphe qu'un arbre. Ainsi, en utilisant les cinq primitives utilisées ici («être égal», «avoir une intersection non vide avec», «être inclus dans»,

«être tangent avec» et «être supérieur à») et que leurs réciproques, on obtient un treillis de systèmes.

Le système présenté ici est une généralisation de l'algèbre d'intervalles temporels: l'algèbre A_8 [RAN 92a] correspond justement à un appauvrissement de A_{13} dans lequel l'ordre n'est plus pris en compte. Ainsi, b et b^{-1}, s et f, s^{-1} et f^{-1}, o et o^{-1}, m et m^{-1} qui n'étaient distingués que par la faculté d'ordonner des intervalles (ou les bornes des intervalles) peuvent-ils être réunis dans une seule relation non ordonnée (correspondant à nto, ti, ti^{-1}, po et to).

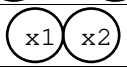
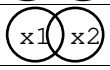
relation (r): *x1* r *x2*	*x1*/*x2*	réciproque: *x2* r^{-1} *x1*
est strictement distinct (nto)		nto
est tangentiellement distinct (to)		to
recouvre partiellement (po)		po
égale (eq)		eq
englobe tangentiellement (ti)		est tangentiellement englobé
englobe strictement (nti)		est strictement englobé

Table 11. Les 8 relations entre deux régions *x1* et *x2*.

A_8 a été axiomatisée en fonction de la notion de connexion (la clôture topologique de deux régions partagent une partie commune) dans [RAN 92a] et la table de transitivité de $\times_8$ est introduite dans [RAN 92b]. Les régions s'interprètent comme des ensembles de points sans aucune restriction supplémentaire (en particulier, elles n'ont pas à être connexe ou convexe).

Les mêmes dispositions touchent les points entre eux (deux relations = et ≠) et les points vis-à-vis des régions ($\in$ et $\notin$). Pour ces dernier, des problèmes similaires à ceux posés pour la situation par rapport aux bornes de l'intervalle (resp. à la limite ou enveloppe topologique de la région) se reproduisent [RAN 92a]; comme précédemment, ils ne sont pas pris en compte.

7.3. Circulation des représentation spatiales

Comme cela a été fait pour les intervalles temporels, la représentation spatiale granulaire peut s'interpréter sur $\mathbb{N}^n$ et non sur $\mathbb{R}^n$. Cependant, de même que l'approximation de $\mathbb{R}$ par une machine est limitée, l'approximation de $\mathbb{R}^n$ par $\mathbb{N}^n$ devient dramatique si des calculs de distance sont nécessaires par exemple [EGE 90]. Contrairement à beaucoup d'autres travaux, les inférences considérées ici seront beaucoup plus topologiques que métriques (voir la discussion à ce sujet dans [BUI 90]).

Quelles propriétés de la circulation entre granularités doivent être valides? Tout d'abord l'abandon de la notion d'ordre rend caduque la propriété de non-croisement (elle se retrouve dans le cas particulier des intervalles; voir annexe C). Cette propriété était très importante pour fixer les conversions. Elle peut être approchée par une propriété de conservation de la

connexité: les enveloppes convexes des régions ne peuvent pas se «croiser» — c'est-à-dire avoir une intersection non vide — lors de la monté si elles ne se croisaient pas avant. Aucune propriété (logique ou topologique) plus précise n'est donnée à l'appui de cette conception (peut-être est-elle axiomatisable en fonction de la connexité [RAN 92a]). La figure 10 en donne une illustration dans le cas de la relation «po», la plus problématique. En fait, ce raisonnement s'applique correctement aux régions connexes, mais il devient beaucoup plus périlleux lorsque les régions ne le sont plus et que certaines parties connexes de régions disparaissent complètement. Dans un tel cas, la contrainte donnée ici est trop forte; elle est cependant maintenue car elle s'applique bien à l'appauvrissement de l'algèbre d'intervalles temporels (où la notion de connexité des intervalles apparaît, dés lors, comme vitale).

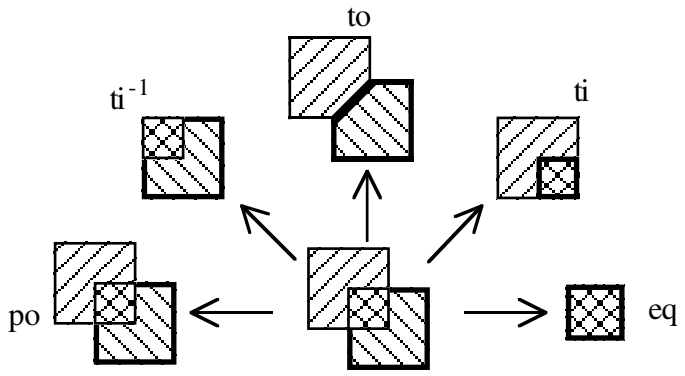


Figure 10. Les transitions autorisées lors de la conversion ascendante de deux régions si la propriété de non croisement des enveloppes est respectée.

Par contre, la transitivité cumulée, la conservation descendante et le principe de compatibilité peuvent être préservés sans difficulté.

Dans la représentation symbolique définie plus haut, les tables de monté et descente des relations peuvent être obtenue par simple abandon de l'ordre (voir tables 12, 13 et 14).

relation: r	${}_g\uparrow^{g'}$r	${}_g\downarrow^{g'}$r
=	=	= ≠
≠	≠ =	≠

Table 12. Conversion des relations spatiales (sans ordre) entre points.

relation: r	${}_g\uparrow^{g'}$r	${}_g\downarrow^{g'}$r
∈	∈	∈ ∉
∉	∉ ∈	∉

Table 13. Conversion des relations spatiales (sans ordre) entre point et région. Le problème des limites réapparaît.

relation (r): *x1* r *x2*	${}_g\uparrow^{g'}$r	${}_g\downarrow^{g'}$r
nto	nto to	nto
to	to	to nto po
po	po to ti ti^{-1} eq	po
eq	eq	eq ti nti po ti^{-1} nti^{-1}
ti	ti eq	ti po nti
nti	nti eq ti	nti

Table 14. Conversion des relations spatiales (sans ordre) entre régions.

Les deux tables obéissent au principe de compatibilité et par simple construction, les deux contraintes suivantes (inclusion du converti et distributivité entre conversion et réciprocité) sont satisfaites:

$$\rho \subseteq g \rightarrow g' \rho$$

$$g \rightarrow g' \rho^{-1} = (g \rightarrow g' \rho)^{-1}$$

Par contre, il est nécessaire d'établir, pour la monté:

$$({}_g\uparrow^{g'} \rho_1) \times_8 ({}_g\uparrow^{g'} \rho_2) \subseteq {}_g\uparrow^{g'} (\rho_1 \times_8 \rho_2)$$

et pour la descente (valide pour les points et les régions):

$${}^g\downarrow_{g'} (\rho_1 \times \rho_2) \subseteq ({}^g\downarrow_{g'} \rho_1) \times ({}^g\downarrow_{g'} \rho_2).$$

L'indépendance de la représentation (régions ou points) n'est pas établie car la conversion entre les deux représentations n'est pas définie (voir annexe C pour l'établissement d'une telle correspondance). Pour la conversion des entités elles-mêmes, l'utilisation de schémas de décomposition comme les histoires — qui peuvent être qualifiés de plans (dotés d'une échelle) — sera certainement la meilleure façon d'aider la conversion par la connaissance du domaine considéré. Ces résultats peuvent être vérifiés exhaustivement (voir annexe A), mais sont prouvés axiomatiquement à partir des résultats obtenus sur A_{13} (resp. A_3, voir annexe B). Il sont ainsi utilisables avec d'autres appauvrissement.

Bien entendu, les représentations de l'espace représentés ici ne couvrent pas l'étendue des besoins ni des applications (vision, raisonnement dans l'espace, mouvements et trajectoires…). Mais les concepts mis en avant pour la représentation granulaire du temps semblent s'adapter, moyennement quelques aménagements, relativement bien à l'espace.

8. Autres travaux

Jerry Hobbs [HOB 85] a introduit le concept de granularité. Pour une théorie logique considérée, une granularité détermine un ensemble de prédicats intéressants sous celle-ci. Une relation de congruence est alors définie qui considère comme indiscernables sous cette granularité les termes qui ne peuvent être discriminés par ces prédicats intéressants. Cette proposition soulève plusieurs problèmes:

— Tout d'abord, il faut disposer d'une théorie initiale au plus bas niveau qui comprenne tous les termes utilisables à tous les niveaux. Cela exclut qu'une source de connaissance fournisse uniquement la connaissance pertinente à son niveau.
— Ensuite, tous les calculs doivent se faire au plus bas niveau et la représentation granulaire ne peut apporter d'avantages calculatoires.
— Enfin, la discrétisation est obtenue à partir de la possibilité de discriminer des termes à l'aide d'un ensemble de prédicats. Cependant, si le monde évolue et que la théorie courante doive aussi évoluer, il faudra alors faire évoluer la discrétisation des termes.

Ces trois problèmes, sont liés à la volonté de définir une granularité a posteriori (en fonction des prédicats intéressants). Cette proposition a cependant l'avantage de ne pas imposer une représentation a priori et d'induire la représentation du temps en fonction de l'intérêt que va lui porter celui qui en a besoin. La représentation du temps est donc relative. Le présent travail s'est focalisé sur la granularité a priori (telle que Jerry Hobbs l'a proposée pour les mesures de températures) tout en réutilisant des résultats partiels de [HOB 85] signalés dans le cours du texte.

Patrick Hayes et James Allen [HAY 87] ont donné une axiomatisation des intervalles temporels en fonction de la relation «termine au début» ("meets"). Ils y introduisent les instants comme les endroits où les intervalles se rencontrent et les moments comme des intervalles indivisibles. Afin de disposer des instants dans la théorie, les moments sont associés aux instants qui les terminent. Le résultat remarquable de ce travail est que l'axiomatisation est la même après un changement de granularité (un changement dans la définition d'indivisible). Ainsi, comme présenté plus haut, la présence de la granularité n'affecte pas les représentations classiques du temps. Aucun résultat n'est donné, par contre, sur les rapports entre les changements de granularité et la propagation de contraintes. Par ailleurs, ce système ne prend pas en compte les aspects numériques.

Cependant, afin de préserver la transitivité de la relation de congruence sans tomber dans le travers évoqué au §4.2 où tous les instants sont congrus au même instant, les auteurs sont obligés de poser une hypothèse très restrictive (en particulier vis-à-vis des changements de granularité successifs): deux moments ne peuvent se rencontrer. À l'opposé, notre attitude lors de la montée est plus proche de celle préconisée par Jerry Hobbs, à savoir découper des parties de la «droite temporelle» dont tous les points vont être congrus au même représentant. Par ailleurs, un intervalle ne peut jamais disparaître lors du changement de granularité (même s'il devient un moment). Il n'est donc pas possible d'utiliser cette axiomatisation pour fonder les opérations de montée et de descente présentées ici: elles ne respectent pas cette propriété.

Enfin, il faut noter que si ces travaux concernent la montée, aucun ne s'est jamais préoccupé de la descente.

Françoise Gayral et Philippe Grandemange [GAY 91] ont abordé la nécessité de considérer une même entité temporelle à la fois sous un aspect duratif et un aspect instantané. Leur travail est motivé par la compréhension de texte. Un mécanisme de montée/descente d'entités temporelles est introduit et modélisé dans un cadre logique. Il ne prend en compte que les contraintes symboliques et ne convertit que les entités et non les relations. Cette représentation s'appuie sur la notion de composition qui permet de décomposer successivement les débuts et fins d'intervalles en nouveaux intervalles: c'est une notion réduite d'histoire. Le niveau de granularité est déterminé par l'élection d'un «individu distingué» parmi l'ensemble des entités considérées et le fait qu'il soit considéré comme duratif ou instantané. Ce travail est donc principalement *relatif*, c'est-à-dire que la granularité courante est relative à la façon de considérer (comme duratif ou instantané) un évènement particulier alors que la conception présentée ici est *absolue* car elle concerne la granularité de manière transversale à toute une base de connaissance. La liaison entre les deux notions est envisageable au travers de la prise en compte de la relation de décomposition des entités temporelles dans le modèle proposé ici, c'est-à-dire en prenant en compte de manière plus générale la notion d'histoire.

Indépendamment du travail présenté ici, Angelo Montanari, Enrico Maim, Emanuele Ciapessoni et Elena Ratto [MON 92] proposent un cadre semblable pour la représentation de la granularité temporelle. Celle-ci est, au contraire de celle présentée aux paragraphes précédents, essentiellement numérique et se fonde sur un modèle du temps très hiérarchisé. Les contraintes symboliques n'y sont pas considérées. L'univers temporel est un ensemble de modèles du temps ordonnés du plus grossier au plus fin $T_1 > \ldots > T_n$ où les T_i sont des univers temporels discrets, à l'exception de T_n qui peut être dense. Pour chaque paire de domaines T_i, T_{i+1}, il existe une application qui associe à chaque instant I de T_i un intervalle $<I^- I^+>$ de T_{i+1} tel que:

(contiguïté) à chaque paire d'instants «contigus» est associée une paire d'intervalles contigus (relations m, m^{-1}).

(préservation de l'ordre) qui correspond à la contrainte de non croisement évoquée ici.

(couverture) l'union des intervalles de l'image de T_i couvre tout T_{i+1}.

(homogénéité) la longueur ($d_{i,i+1}$) des intervalles images d'instants par l'application est constante.

Cette application se généralise en une application d'un domaine grossier quelconque vers un domaine plus fin correspondant à l'opération de descente. Du fait de toutes ces restrictions, le travail présenté ici est plus général mais les propriétés obtenues sont moins favorables. Les travaux plus récent de la même équipe permettent de proposer encore plus de restrictions sur cette application afin d'obtenir des comportements temporels plus précis [CIA 93b] mais encore de prendre en compte des granularités ordonnées uniquement par un ordre partiel [CIA 93a].

Les auteurs proposent également des opérateurs de montée et de descente. La montée naïve est similaire à l'opérateur présenté ici, elle préserve l'ordre mais pas la différence entre deux

entités. La descente correspond à la descente présentée ci-dessus mais un certain nombre de contraintes sont imposées afin que des assertions contradictoires et contiguës ne soient pas valides simultanément dans l'intervalle image de leur point de contact par la descente. Pour cela, les instants de terminaison et de commencement de ces assertions sont représentés sous le grain le plus fin par des instants ordonnés distincts des bornes de l'intervalle image. Cela est réalisé à l'aide de la notion de *macro-évènements* (similaires aux histoires), introduite par Chris Evans, qui permet de représenter la descente. Il faut noter que ces macro-évènements étant définis en fonction des assertions exprimées dans la base, ils relèvent d'une conception a posteriori de la granularité. Cependant, s'ils sont dépendants du type d'évènements, ou encore, exprimés comme des histoires, alors ils peuvent être considérés comme donnés a priori. Le problème de la confusion de deux expressions temporelles lors de la montée a été aussi résolu de cette façon: en considérant que deux expressions temporelles qui se trouveraient confondues lors de la montée sont en fait des épisodes d'une même histoire qui doit donc les remplacer (toutes les deux) au niveau supérieur.

Sous l'aspect implémentation, Thomas Dean [DEA 89] présente la notion de *bassins* ("buckets"): un ensemble de données temporelles discriminées suivant une série de conditions. Ils sont utilisés en particulier dans les *schémas de partition hiérarchique* pour discriminer suivant la position dans une partition du temps homogène définie a priori (l'exemple donné est la partition en mois, jours, heures, minutes et secondes). Les bassins fonctionnent en quelque sorte comme les arbres quaternaires pour l'information spatiale, ils se décomposent au fur et à mesure qu'ils contiennent plus d'éléments. Cette organisation, bien que fondée sur une décomposition du temps en unités de même taille, est assez différente de celle préconisée par la granularité: un intervalle d'une durée de la minute peut se trouver stocké au niveau du siècle pour peu qu'il se passe au changement de siècle. L'organisation est duale de la représentation granulaire: la première organise les données suivant leur position dans le temps alors que la seconde les organise suivant leur taille. Johannes Koomen [KOO 89] a montré que les intervalles de référence [ALL 83] rendent des résultats incomplets s'ils sont utilisés rigidement et a proposé de maintenir automatiquement une organisation hiérarchique des intervalles de référence. Cette proposition correspond aux schémas de partition hiérarchique, mais organise les bassins en fonction de leur contenance et non plus de leur position.

Il faut, par ailleurs, signaler les travaux de Catherine Joubel et Olivier Raiman [JOU 90], orthogonaux à ceux développés ici. Au lieu de considérer l'influence de la granularité sur le temps comme cela a été fait ici, ils évaluent celle du temps sur le contexte de raisonnement. Le temps ne fait donc pas partie des données à représenter mais détermine le point de vue.

9. Conclusion

L'utilisation d'informations temporelles sous divers niveaux de détail a conduit à mettre en avant la notion de granularité de la représentation temporelle. Celle-ci permet de limiter sous un point de vue la puissance de la représentation temporelle à la taille du plus petit évènement pertinent.

La représentation temporelle granulaire présentée se diffuse sur deux axes:

— la représentation des expressions temporelles sous une granularité précise;

— la conversion des expressions temporelles d'une granularité vers une autre.

Sur le premier axe, la représentation proposée permet de conserver les représentations temporelles symboliques classiques: l'appareil complet des représentations symboliques est disponible sous chacune des granularités. Cependant l'échelle utilisée permet de considérer le temps comme congru non plus à $\mathbb{R}$ mais à $\mathbb{N}$, ce qui conduit à limiter la complexité des opérations sur la représentation. Par ailleurs, le fait que, suivant la granularité, une information temporelle puisse durer ou non mène à l'identification des concepts d'instant et d'intervalle.

Les règles de conversion sont compatibles avec les représentations temporelles sur chacun des niveaux. En particulier, la conversion entre intervalles et instants se retrouve à tous les niveaux de granularité. Du coté numérique, les règles de conversion définies remplissent certaines des conditions escomptées (non croisement d'entités, compatibilité ascendante). Des restrictions sur le modèle général proposé ont été avancées afin d'obtenir certaines propriétés additionnelles. Ainsi, pour disposer de la propriété de transitivité cumulée, lors de la montée, il faut utiliser des contraintes sur l'ensemble des grains disponibles.

Les apports de la représentation temporelle granulaire sont les suivants:

— La connaissance peut être introduite au niveau de détail le plus précis où elle est disponible et exploitée de manière consistante sur les autres niveaux de détail.

— D'autre part, la granularité permet de ne considérer que les entités temporelles présentes à un niveau de détail et d'éliminer les autres. Ceci est très intéressant lorsqu'une information est recherchée d'une manière volontariste: il est possible de restreindre la recherche à un certaine granularité, puis de l'étendre à d'autres granularités si la première n'est pas suffisante.

Il semble que ce travail sur la granularité temporelle puisse être considérée comme un cas particulier de la granularité dans un espace de dimension quelconque finie.

L'imprécision apparente introduite lors de la propagation des contraintes symboliques entre les granularités est nécessaire à la compatibilité entre information symbolique et numérique (elle est le reflet du souci de laisser la porte ouverte à toutes les interprétations des expressions

temporelles manipulées). Elle nécessite l'utilisation systématique de procédures de propagation de contraintes symboliques déduisant automatiquement les relations les plus précises possibles entre entités temporelles. La réduction de l'imprécision introduite par la conversion peut être obtenue en ajoutant de la précision, c'est-à-dire de la connaissance supplémentaire. Celle-ci peut provenir de différentes sources:

— Des contraintes générales imposées à la structure de la représentation granulaire ([MON 92, CIA 93b]).
— De la connaissance spécifique à l'entité temporelle manipulée ajoutée par l'utilisateur (encore faut-il qu'il en dispose, auquel cas il est naturel qu'il l'exprime).
— De la connaissance générique liée au type d'entités manipulées par l'application tel que les macro-évènements ou les histoires (voir §6.2).

Remerciements

Philippe Kirsch et Jean-Marc Pugin participaient à la discussion au cours de laquelle les premières idées sur la représentation granulaire ont écloses. Les nombreuses discussions serrées avec Patrice Uvietta, Johannes Stein, Pierre Girard et Danielle Ziébelin des précédentes versions de ce papier ont grandement participé à sa maturation. Enfin, les lecteurs de la «revue d'intelligence artificielle» ont, par leurs suggestions judicieuses, contribué à la bonne formulation de ce travail

Références

[ALL 81] James ALLEN, «An interval-based representation of temporal knowledge», Actes 7th IJCAI, Vancouver (CA), pp221-226, 1981

[ALL 83] James ALLEN, «Maintaining knowledge about temporal intervals», *Communication of the ACM* 26-XI, pp832-843 (rep. dans Ronald Brachman, Hector Levesque (éds.), Readings in knowledge representation, Morgan Kaufmann, Los Altos (CA US), pp509-521, 1985), 1983

[BAR 70] Marc BARBUT, Bernard MONJARDET, *Ordre et classification: algèbre et combinatoire (tome 2)*, Hachette, Paris (FR), 1970

[BES 89] Hélène BESTOUGEFF, Gérard LIGOZAT, *Outils logiques pour le traitement du temps: de la linguistique à l'intelligence artificielle*, Masson, Paris (FR), 1989

[BON 88] Eric BONTÉ, Jacqueline CASTAING, Philippe GRANDEMANGE, Stéphane GRUMBACH, Daniel KAYSER, François LÉVY, «Description succincte d'un raisonneur à profondeur variable», Actes 8ièmes journées sur les systèmes experts et leurs applications, Avignon (FR), pp117-132, 1988

[BUI 90] Laurent BUISSON, *Le raisonnement spatial dans les systèmes à base de connaissances: application à l'analyse de sites avalancheux*, Thèse, Université Joseph-Fourier, Grenoble (FR), 1990

[CIA 93a] Emanuele CIAPESSONI, Edoardo CORSETTI, Angelo MONTANARI, P. SAN PIETRO, «Embedding time granularity in a logical specification language for

sychronous real-time systems», *Science of computer programming* 20-I, pp141-171, 1993

[CIA 93b] Emanuele CIAPESSONI, Edoardo CORSETTI, Manlio MIGLIORATI, Elena RATTO, «Logical specification of real-time granular systems in an object-oriented language», Actes 13th IJCAI, Chambéry (FR), pp881-886, 1993

[DEA 87] Thomas DEAN, Drew MAC DERMOTT, «Temporal data base management», *Artificial intelligence* 32-I, pp1-55, 1987

[DEA 89] Thomas DEAN, «Using temporal hierarchies to efficiently maintain large temporal databases», *Journal of the ACM* 36-IV, pp687-718, 1989

[EGE 90] Max EGENHOFER, Andrew FRANK, Jeffrey JACKSON, «A topological model for spatial databases», *Lecture notes on computer science* 409 (Alejandro Bachmann, Oliver Günther, Terence Smith, Yuan-F. Wang (éds.), actes Symposium on design and implementation of very large spatial data dases, Santa-Barbara (CA US), 1989), pp325-344, 1990

[EUZ 93] Jérôme EUZENAT, «Représentation granulaire du temps», *Revue d'intelligence artificielle* 7-III, pp329-361, 1993

[FRE 93] Christian FREKSA, «Temporal reasoning based on semi-intervals», *Artificial intelligence* 54-I, pp199-227, 1992

[GAY 91] Françoise GAYRAL, Philippe GRANDEMANGE, «Événements: ponctualité et durativité», Actes 8ième RFIA, Villeurbanne (FR), pp905-910, 1991

[GHA 89] Malik GHALLAB, Amine MOUNIR-ALAOUI «Relations temporelles symboliques: représentations et algorithmes», *Revue d'intelligence artificielle* 3-III, 67-115, 1989

[GUI 93] John GUIDI, Thomas RODERICK, «Inference of order in genetic systems», Actes 1st international conference on intelligent systems for molecular biology, Bethesda (MD US), pp163-171, 1993

[HAJ 89] Elzbieta HAJNICZ, «Absolute date and relative dates in an inferential system on temporal dependencies between events», *International journal of man-machine studies* 30, pp537-449, 1989

[HAY 87] Patrick HAYES, James ALLEN, «Short time periods», Actes 10th IJCAI, Milan (IT), pp981-983, 1987

[HOB 85] Jerry HOBBS, «Granularity», Actes 9th IJCAI, Los Angeles (CA US), pp432-435, 1985

[JOU 90] Catherine JOUBEL, Olivier RAIMAN, «How time changes assumptions», Actes 10ièmes journées internationnales sur les systèmes experts et leurs applications, Avignon (FR), pp197-208, 1990 & actes 9th ECAI, Stockholm (SE), pp378-383, 1990

[KOO 89] Johannes KOOMEN, «Localizing temporal constraint propagation», Actes 1st KR, Toronto (CA), pp198-202, 1989

[LER 93] Xavier LEROY, Pierre WEIS, *Manuel de référence du langage CAML*, InterÉditions, Paris (FR), 1993

[MCC 69] John MAC CARTHY, Patrick HAYES, «Some philosophical problems from the standpoint in artificial intelligence», *Machine intelligence* 4, pp463-502, 1969

[MCD 81] Drew MAC DERMOTT, «A temporal logic for reasoning about processes and plans», Rapport de recherche 196, Yale university, New-Heaven (CT US) (rep. *Cognitive Science* 6, pp101-155, 1982), 1981

[MAR 90] Olga MARIÑO, François RECHENMANN, Patrice UVIETTA, «Multiple perspectives and classification mechanism in object-oriented representation», Actes 9th ECAI, Stockholm (SE), pp425-430, 1990

[MAR 93] Olga MARIÑO, *Raisonnement classificatoire dans une représentation à objets multi-points de vue*, Thèse, Université Joseph-Fourier, Grenoble (FR), 1993

[MAU 91] Michel MAUNY, «Functional programming using CAML», Rapport technique 129, INRIA, Rocquencourt (FR), 1991

[MON 92] Angelo MONTANARI, Enrico MAIM, Emanuele CIAPESSONI, Elena RATTO, «Dealing with time and granularity in the event calculus», Actes 4th FGCS, Tokyo (JP), pp702-712, 1992

[RAN 92a] David RANDELL, Zhan CUI, Anthony COHN, «A spatial logic based on regions and connection», Actes 3rd KR, Cambridge (MA US), pp165-176, 1992

[RAN 92b] David RANDELL, Zhan CUI, Anthony COHN, «Computing transitivity tables: a challenge for automated theorem provers», *Lecture notes in computer science (lecture notes in artificial intelligence)* 607 (proc. 11th CADE, Saratoga Springs (NY US), 1992), pp786-790, 1992

[RIT 88] Jean-François RIT, *Modélisation et propagation de contraintes temporelles pour la planification*, Thèse, Institut National Polytechnique, Grenoble (FR), 1988

[SAM 86] Hanan SAMET, «The quadtree and related hierarchical data structures», *ACM computing surveys* 16(2):187-260, 1986

[SCH 82] Roger SCHANK, *Dynamic memory, a theory of reminding and learning in computers and people*, Cambridge university press, Cambridge (GB), 1982

[SCH 94] Olivier SCHMELTZER, Frank DORKELD, «Modélisation de cartes génomiques», Rapport non publié, INRIA Rhône-Alpes, Grenoble (FR), 1994

[SHO 86] Yoav SHOHAM, *Reasoning about change: time and causation from the standpoint of artificial intelligence*, Thèse, Yale university, New-Heaven (CT US) (rev. The MIT press, Cambridge (MA US), 1988), 1986

[STE 91] Johannes STEIN, «Entwicklung eines Zeitmodells mit der Fähigkeit zur Variation der Granularität und dessen Implementierung im Repräsentationswerkzeug TROPES», Diplomarbeit, Universität Kaiserslautern (DE), 1991

[VAN 89] Peter VAN BEEK, «Approximation algorithms for temporal reasoning», Actes 11th IJCAI, Detroit (MI US), pp1291-1296, 1989

[VIL 86] Marc VILAIN, Henry KAUTZ, «Constraint propagation algorithms for temporal reasoning», Actes 5th AAAI, Philadelphia (PA US), pp377-479, 1986

[WEI 93] Pierre WEIS, Xavier LEROY, *Le langage CAML: introduction à la programmation*, InterÉditions, Paris (FR), 1993

Annexe A. Programme de vérification des résultats énoncés pour les algèbres de relations A_{13}, A_8, A_3 et A_2

Toutes les propriétés énoncées ci-dessus ont été vérifiées exhaustivement. Afin de réduire le travail du programme, la propriété suivante (si deux fonctions quasi-commutent — l'ordre d'utilisation des deux opérateurs donne des résultats en relation d'inclusion — sur tout singleton, sous-ensemble d'un domaine précis, alors elles quasi-commutent sur tout sous-ensemble non-vide du domaine) est utilisée:

$$\forall x \in \mathfrak{D},\ f_1(f_2(x)) \subseteq f_2(f_1(x))$$

$$\Rightarrow \forall E \subseteq \mathfrak{D},\ \bigcup_{x \in E} f_1(f_2(x)) \subseteq \bigcup_{x \in E} f_2(f_1(x))$$

Ce qui s'étend à $f_1(f_2(E)) = f_2(f_1(E))$ en inversant les rôles de f_1 et f_2. Par conséquent, il suffit de montrer que deux opérations quasi-commutent sur tous les singletons pour montrer qu'elles quasi-commutent sur tous les sous-ensembles non vide d'un domaine particulier. C'est ce qui est réalisé ici.

Le programme qui suit, écrit en CAML [MAU 91, LER 93, WEI 93], vérifie chacune de ces propriétés. Ce programme fonctionne avec les systèmes Caml 2.6-1 and Caml Light 0.6. Il nécessite la définition des primitives ensemblistes suivantes: `set_mapunion` (∪), `set_product` (×), `set_eq` (=), `set_incl` (⊇), `set_is_incl` (⊆), et `set_uniq` (qui supprime les doublons dans les listes).

Propriétés de la représentation temporelle symbolique (A_{13} et A_3)

La relation «o» est représentée par `ov` car `o` est l'opérateur de composition de fonctions en ML. Les parties du texte qui suivent en `courrier` constituent le programme; les commentaires en police courante sont l'interprétation de chaque programme.

```
(* ------------------------------------------------------------------ *)
(* timesys.ml *)

(* Table 1, A3 = {<,=,>} *)
type instrel = sup | equ | inf;;
let A3 = [sup;equ;inf];;

(* ∀x∈A3, x^-1 = inv_inst(x) *)
let inv_inst = function sup -> inf | equ -> equ | inf -> sup;;

(* Table 2, ∀(x,y)∈A3×A3, x×3y = X_inst (x,y) *)
let X_inst = function sup,inf -> A3 | sup,_ -> [sup]
                    | equ,x -> [x]
                    | inf,sup -> A3 | inf,_ -> [inf];;
```

```
(* Table 3, A13 = {b, d, o, s, f, m, e, m-1, f-1, s-1, o-1, d-1, b-1 } *)
type intrel = b | d | ov | s | f | m | e | mi | fi | si | oi | di | bi;;
let A13 = [b;d;ov;s;f;m;e;mi;fi;si;oi;di;bi];;

(* ∀x∈A13, x-1 = inv_int(x) *)
let inv_int = function e -> e
 | b -> bi | d -> di | ov -> oi | s -> si | f -> fi | m -> mi
 | mi -> m | fi -> f | si -> s | oi -> ov | di -> d | bi -> b;;

(* Table de transitivité [ALL 83], ∀(x,y)∈I13×I13, x×13y = X_int (x,y),
  utilise la propriété x×13y = (y-1×13x-1)-1 *)
let rec X_int = function
    e,x -> [x]
  | s,s -> [s]
  | f,f -> [f]
  | ov,s | fi,ov | fi,s -> [ov]
  | b,fi | b,ov | b,s | b,m | b,si | d,b | m,b | s,m | m,m | m,ov | d,m
  | m,di | ov,b | ov,m | fi,b | b,di | m,fi | b,b | s,b | f,b -> [b]
  | d,d | d,s | d,f | s,f | f,s | s,d | f,d -> [d]
  | m,s | m,si | fi,m | f,m -> [m]
  | s,oi | oi,d | oi,s | mi,d | mi,ov | mi,s | si,d -> [oi;d;f]
  | ov,ov | ov,fi | s,ov | s,fi -> [b;ov;m]
  | m,d | m,f | m,oi | ov,d | ov,f | f,ov | fi,d -> [ov;d;s]
  | m,mi | f,fi | fi,f -> [f;fi;e]
  | s,si | mi,m | si,s -> [s;e;si]
  | ov,di | s,di | di,b | oi,b | mi,b | si,b -> [b;ov;m;di;fi]
  | b,d | b,f | b,oi | b,mi | d,ov | d,fi -> [b;ov;m;d;s]
  | ov,oi | di,d | oi,ov -> [ov;oi;d;di;s;si;f;fi;e]
  | b,bi | d,di | bi,b -> A13
  | x,y -> map inv_int (X_int ((inv_int y),(inv_int x)));;

(* Table 4 : Conversion intervalles en instants repre'sente's par
  des vecteurs de 4 e'le'ments d1.d2 d1.f2 f1.d2 f1.f2,
  ∀x∈A13, ⇒x = int_to_inst(x) *)

let int_to_inst = function
    b -> inf, inf, inf, inf    | bi -> sup, sup, sup, sup
  | d -> sup, inf, sup, inf    | di -> inf, inf, sup, sup
  | ov -> inf, inf, sup, inf   | oi -> sup, inf, sup, sup
  | s -> equ, inf, sup, inf    | si -> equ, inf, sup, sup
  | f -> sup, inf, sup, equ    | fi -> inf, inf, sup, equ
  | m -> inf, inf, equ, inf    | mi -> sup, equ, sup, sup
  | e -> equ, inf, sup, equ;;

(* Convertion vecteurs d'intervalles en instants,
  ∀x∈A3×A3×A3×A3, ⇐x = inst_to_int(x) *)
exception Not_interval;;

let inst_to_int = function
    inf, inf, inf, inf -> b   | sup, sup, sup, sup -> bi
  | sup, inf, sup, inf -> d   | inf, inf, sup, sup -> di
  | inf, inf, sup, inf -> ov  | sup, inf, sup, sup -> oi
  | equ, inf, sup, inf -> s   | equ, inf, sup, sup -> si
  | sup, inf, sup, equ -> f   |   inf, inf, sup, equ -> fi
  | inf, inf, equ, inf -> m   | sup, equ, sup, sup -> mi
  | equ, inf, sup, equ -> e   | _,_,_,_ -> raise Not_interval;;
```

```
(* ------------------------------------------------------------------ *)
(* tegcirc.ml *)

(* Operateurs de montee et de descente *)

(* Table 6, ∀x∈A3, ↑x = montee_inst(x) *)
let montee_inst =
   function sup -> [sup;equ] | inf -> [inf;equ] | equ -> [equ];;

(* Table 7, ∀x∈A13, ↑x = montee_int(x)  *)
let montee_int = function e -> [e]
    | b -> [b;m]           | bi -> [bi;mi]
    | m -> [m]             | mi -> [mi]
    | ov -> [ov;fi;s;m;e]  | oi -> [oi;si;f;mi;e]
    | d -> [d;f;s;e]       | di -> [di;si;fi;e]
    | s -> [s;e]           | si -> [si;e]
    | f -> [f;e]           | fi -> [fi;e];;

(* Table 8, ∀x∈A3, ↓x = descente_inst(x)  *)
let descente_inst = function  sup -> [sup]| equ -> I3| inf -> [inf];;

(* Table 9, ∀x∈A13, ↓x = descente_int(x)  *)
let descente_int = function e -> [ov;fi;di;s;e;si;d;f;oi]
    | b -> [b]             | bi -> [bi]
    | m -> [b;m;ov]        | mi -> [bi;mi;oi]
    | ov -> [ov]           | oi -> [oi]
    | d -> [d]             | di -> [di]
    | s -> [ov;s;d]        | si -> [oi;si;di]
    | f -> [d;f;oi]        | fi -> [di;fi;ov];;

(* ------------------------------------------------------------------ *)
(* tegtest.ml *)

(*Extension des fonction sur des relations aux ensembles de relations*)

(* ∀ρ1⊆Γ, ∀ρ2⊆Γ, ρ1×Γρ2 = transit_star(×Γ ρ1 ρ2 ) *)
let transit_star tab relset1 relset2 =
  set_mapunion tab (set_product relset1 relset2);;

(* ∀ρ⊆Γ, →ρ = convert_star(→ρ ) *)
let convert_star tab relset = set_mapunion tab relset;;

(*Application de la conversion à un vecteur de relations entre instants
∀<r1,r2,r3,r4>∈Γ×Γ×Γ×Γ, →r1×→r2×→r3×→r4 = convert_vect(→ r1,r2,r3,r4)  *)
let convert_vect tab (w,x,y,z) =
  set_uniq
   (map (function (w,(x,(y,z))) -> w,x,y,z)
        (set_product (tab w)
            (set_product (tab x) (set_product (tab y) (tab z))))));;

(* Fonction permettant de tester les équations du type
   ∀x∈D, P(f1(x), f2(x)) ≡ test (P f1 f2 D) *)
exception False;;

let test pred f1 f2 set =
  try do_list (function x -> if not (pred (f1 x) (f2 x))
                             then raise False)
              set;
      true
  with False -> false;;
```

```
(* Test de distributivité d'un opérateur de conversion et d'une table
   de transitivité:
∀(x,y)∈Γ×Γ, P(→x×Γ→y, →(x×Γy)) ≡ distrib_convert_transit (P → ×Γ Γ) *)
let distrib_convert_transit testfn convert transit support =
  let f1 (r1,r2) = transit_star transit (convert r1) (convert r2)
  and f2 r1xr2 = convert_star convert (transit r1xr2)
  in test testfn f1 f2 (set_product support support);;

(* Test de commutativité d'un opérateur de conversion et d'un
   opérateur d'invertion
  ∀x∈Γ, P(→x^-1, (→x)^-1) ≡ commut_convert_inv (P → ^-1 Γ) *)
let commut_convert_inv testfn convert inv support =
  let f1 rel = convert (inv rel)
  and f2 rel = map inv (convert rel)
  in test testfn f1 f2 support;;

(* Test d'indépendance de la représentation
  ∀x∈Γ, P(→x, ⇐→⇒x) ≡ commut_convert_inv (P → →Γ) *)
let commut_convert_repr testfn convert_int convert_inst support =
  let f2 rel = map_intervals inst_to_int
                    (convert_vect convert_inst (int_to_inst rel))
    where rec map_intervals oper = function [] -> []
    | x::l -> let resl = map_intervals oper l
              in try (oper x)::resl
                 with Not_interval -> resl
  in test testfn convert_int f2 support;;
```

Le test de chacune des propriétés donne les résultats suivants:

```
#commut_convert_inv set_equal montee_inst inv_inst A3;;
- : bool = true (* ∀r∈A3, ↑r^-1 = (↑r)^-1 *)
#distrib_convert_transit set_equal montee_inst X_inst A3;;
- : bool = true (* ∀(r1,r2)∈A3×A3, ↑r1×3↑r2 = ↑(r1×3r2)) *)
#commut_convert_inv set_equal montee_int inv_int A13;;
- : bool = true (* ∀r∈A13, ↑r^-1 = (↑r)^-1 *)
#distrib_convert_transit set_equal montee_int X_int A13;;
- : bool = false (* ∀(r1,r2)∈A13×A13, ↑r1×13↑r2 ≠ ↑(r1×13r2)) *)
#distrib_convert_transit set_is_incl montee_int X_int A13;;
- : bool = true (* ∀(r1,r2)∈A13×A13, ↑r1×13↑r2 ⊆ ↑(r1×13r2)) *)
#commut_convert_repr set_equal montee_int montee_inst A13;;
- : bool = true (* ∀r∈A13, ↑x = ⇐↑⇒r *)
#commut_convert_inv set_equal descente_inst inv_inst A3;;
- : bool = true (* ∀r∈A3, ↓r^-1 = (↓r)^-1 *)
#distrib_convert_transit set_incl descente_inst X_inst A3;;
- : bool = true (* ∀(r1,r2)∈A3×A3, ↓r1×3↓r2 ⊇ ↓(r1×3r2)) *)
#commut_convert_inv set_equal descente_int inv_int A13;;
- : bool = true (* ∀r∈A13, ↓r^-1 = (↓r)^-1 *)
#distrib_convert_transit set_equal descente_int X_int A13;;
- : bool = false (* ∀(r1,r2)∈A13×A13, ↓r1×13↓r2 ≠ ↓(r1×13r2)) *)
#distrib_convert_transit set_incl descente_int X_int A13;;
- : bool = true (* ∀(r1,r2)∈A13×A13, ↓r1×13↓r2 ⊇ ↓(r1×13r2)) *)
#commut_convert_repr set_equal descente_int descente_inst A13;;
- : bool = true (* ∀r∈A13, ↓r = ⇐↓⇒r *)
```

Propriétés de la représentation spatiale symbolique (A_8 et A_2)

Ici, la relation «to» est représentée par le symbole `tou` car `to` est un mot-clé du langage ML.

```
(* ------------------------------------------------------------------ *)
(* spacsys.ml *)

(* A2 = {=,≠} *)
type ptrel = egal | diff;;
let A2 = [egal;diff];;

(* ∀ x∈A2, x-1 = inv_pt(x) *)
let inv_pt = function x -> x;;

(* ∀ (x,y)∈A2×A2, x×2y = X_pt (x,y) *)
let X_pt = function egal,egal -> [egal] | diff,diff -> A2 | _,_ -> [diff];;

(* Table 11, A8 = {nto, to, ov, e, ti, nti, tii, ntii} *)
type sprel = nto|tou|po|eq|ti|nti|tii|ntii;;
let A8 = [nto;tou;po;eq;ti;nti;tii;ntii];;

(* ∀ x∈A8, x-1 = inv_sp(x) *)
let inv_sp = function
  nti -> ntii | ntii -> nti | ti -> tii | tii -> ti
  | x -> x;;

(* Table de transitivité [RAN 92b], ∀ (x,y)∈A8×A8 x×8y = X_sp (x,y),
  utilise la propriété x×8y = (y-1×8x-1)-1 *)
let X_sp = function
  eq,x -> [x]
  | x,eq -> [x]
  | ti,nto | nti,nto | nti,tou | nto,tii | nto,ntii | tou,ntii -> [nto]
  | nti,ti | ti,nti | nti,nti -> [nti]
  | ntii,tii | tii,ntii | ntii,ntii -> [ntii]
  | ti,ti -> [ti;nti]
  | tii,tii -> [tii;ntii]
  | ti,tou | tou,tii -> [nto;tou]
  | ntii,tou | tii,po | ntii,po -> [po;tii;ntii]
  | po,ti | tou,nti | po,nti | tii,nti-> [po;ti;nti]
  | ntii,ti -> [po;tii;ntii]
  | tii,tou -> [tou;po;tii;ntii]
  | tou,ti -> [tou;po;nti;ti]
  | tii,ti -> [po;ti;tii;eq]
  | nto,tou | nto,po | tou,po | ti,po | nti,po | nto,ti | nti,tii
  | nto,nti -> [nto;tou;po;ti;nti]
  | tou,nto | po,nto | tii,nto | ntii,nto | po,tou | po,tii | po,ntii
  | ti,ntii -> [nto;tou;po;tii;ntii]
  | tou,tou | ti,tii -> [nto;tou;po;ti;tii;eq]
  | ntii,nti -> [po;ti;nti;tii;ntii;eq]
  | nto,nto | po,po | nti,ntii -> A8;;
```

```
(* ------------------------------------------------------------------ *)
(* spgcirc.ml *)

(* Operateurs de montee et de descente *)

(* Table 12, ∀x∈A2, ↑x = montee_pt(x) *)
let montee_pt = function egal -> [egal] | diff -> A2;;

(* Table 14, ∀x∈A8, ↑x = montee_sp(x)  *)
let montee_sp = function
  nto -> [nto;tou]        | tou -> [tou]
  | eq -> [eq]            | po -> [po;tou;ti;tii;eq]
  | ti -> [ti;eq]         | tii -> [tii;eq]
  | nti -> [nti;ti;eq]    | ntii -> [ntii;tii;eq];;

(* Table 12, ∀x∈A2, ↓x = descente_pt(x)  *)
let descente_pt = function egal -> A2 | diff -> [diff];;

(* Table 14, ∀x∈A8, ↓x = descente_sp(x)  *)
let descente_sp = function
  nto -> [nto]            | tou -> [tou;nto;po]
  | po -> [po]            | eq -> [eq;ti;nti;po;tii;ntii]
  | ti -> [ti;po;nti]     | tii -> [tii;po;ntii]
  | nti -> [nti]          | ntii -> [ntii];;
```

Le test de chacune des propriétés donne les résultats suivants:

```
#commut_convert_inv set_equal montee_pt inv_pt A2;;
- : bool = true (* ∀r∈A2, ↑r-1 = (↑r)-1 *)
#distrib_convert_transit set_equal montee_pt X_pt A2;;
- : bool = true (* ∀(r1,r2)∈A2×A2, ↑r1×2↑r2 = ↑(r1×2r2)) *)
#commut_convert_inv set_equal montee_sp inv_sp A8;;
- : bool = true (* ∀r∈A8, ↑r-1 = (↑r)-1 *)
#distrib_convert_transit set_equal montee_sp X_sp A8;;
- : bool = false (* ∀(r1,r2)∈A8×A8, ↑r1×8↑r2 ≠ ↑(r1×8r2)) *)
#distrib_convert_transit set_is_incl montee_sp X_sp A8;;
- : bool = true (* ∀(r1,r2)∈A8×A8, ↑r1×8↑r2⊆ ↑(r1×8r2)) *)

#commut_convert_inv set_equal descente_pt inv_pt A2;;
- : bool = true (* ∀r∈A2, ↓r-1 = (↓r)-1 *)
#distrib_convert_transit set_incl descente_pt X_pt A2;;
- : bool = true (* ∀(r1,r2)∈A2×A2, ↓r1×2↓r2 ⊇ ↓(r1×2r2)) *)
#commut_convert_inv set_equal descente_sp inv_sp A8;;
- : bool = true (* ∀r∈A8, ↓r-1 = (↓r)-1 *)
#distrib_convert_transit set_equal descente_sp X_sp A8;;
- : bool = false (* ∀(r1,r2)∈A8×A8, ↓r1×8↓r2 ≠ ↓(r1×8r2)) *)
#distrib_convert_transit set_incl descente_sp X_sp A8;;
- : bool = true (* ∀(r1,r2)∈A8×A8, ↓r1×8↓r2 ⊇ ↓(r1×8r2)) *)
```

Annexe B. Preuve axiomatique des propriétés de A_8 et A_2

L'algèbre A_8 (resp. A_2) est obtenue par appauvrissement de A_{13} (resp. A_3). Il est par conséquent possible d'obtenir les propriétés des algèbres appauvries à partir des algèbres initiales. Les opérateurs d'appauvrissement correspondants sont introduits sur deux algèbres quelconques Γ (initiale) et Γ' (appauvrie):

$$\Downarrow: 2^{\Gamma} \longmapsto 2^{\Gamma'}$$
$$\Uparrow: 2^{\Gamma'} \longmapsto 2^{\Gamma}$$

respectant les propriétés suivantes:

(1) $\Downarrow\Uparrow\rho=\rho$

(2) $\Uparrow\Downarrow\rho=\rho$

Les opérateurs de réciprocité, transitivité et conversion (ascendante et descendante) sont définis par l'appauvrissement:

(définition $^{-1}$ sur Γ') $\rho^{-1}=\Downarrow(\Uparrow\rho)^{-1}$

(définition $\times$ sur Γ') $\rho_1\times\rho_2=\Downarrow(\Uparrow\rho_1\times\Uparrow\rho_2)$

(définition $\rightarrow$ sur Γ') $\rightarrow\rho=\Downarrow\rightarrow\Uparrow\rho$

On peut alors prouver (sur Γ'):

(commutativité $\rightarrow/\Downarrow$ sur Γ)

$\rightarrow\Downarrow\rho$ (définition $\rightarrow$ sur Γ')
$= \Downarrow\rightarrow\Uparrow\Downarrow\rho$ (2)
$= \Downarrow\rightarrow\rho$ ◊

(commutativité $\rightarrow/\Uparrow$ sur Γ')

$\rightarrow\Uparrow\rho$ (2)
$= \Uparrow\Downarrow\rightarrow\Uparrow\rho$ (définition $\rightarrow$ sur Γ')
$= \Uparrow\rightarrow\rho$ ◊

(idempotence de $\rightarrow$)

$\rightarrow\rightarrow\rho$ (définition $\rightarrow$ sur Γ')
$= \Downarrow\rightarrow\Uparrow\Downarrow\rightarrow\Uparrow\rho$ (2)
$= \Downarrow\rightarrow\rightarrow\Uparrow\rho$ (idempotence de $\rightarrow$ sur Γ)
$= \Downarrow\rightarrow\Uparrow\rho$ (définition $\rightarrow$ sur Γ')
$= \rightarrow\rho$ ◊

(inclusion dans le converti)

ρ (1)
$= \Downarrow\Uparrow\rho$ (inclusion dans le converti sur Γ)
$\subseteq \Downarrow\rightarrow\Uparrow\rho$ (définition $\rightarrow$ sur Γ')
$= \rightarrow\rho$ ◊

(distributivité conversion/réciprocité)	$\rightarrow\rho^{-1}$	(définition $^{-1}$ sur Γ')
	$= \rightarrow\Downarrow(\Uparrow\rho)^{-1}$	(commutativité $\rightarrow/\Downarrow$ sur Γ)
	$= \Downarrow\rightarrow(\Uparrow\rho)^{-1}$	(distributivité sur Γ)
	$= \Downarrow(\rightarrow\Uparrow\rho)^{-1}$	(2)
	$= \Downarrow(\Uparrow\Downarrow\rightarrow\Uparrow\rho)^{-1}$	(définition $^{-1}$ sur Γ')
	$= (\Downarrow\rightarrow\Uparrow\rho)^{-1}$	(définition $\rightarrow$ sur Γ')
	$= (\rightarrow\rho)^{-1}$	$\Diamond$
(distributivité de $\rightarrow$ sur $\times$)	$\rightarrow\rho_1\times\rightarrow\rho_2$	(définition $\times$ sur Γ')
	$= \Downarrow(\Uparrow\rightarrow\rho_1\times\Uparrow\rightarrow\rho_2)$	(commutativité $\rightarrow/\Uparrow$ sur Γ')
	$= \Downarrow(\rightarrow\Uparrow\rho_1\times\rightarrow\Uparrow\rho_2)$	(distributivité de $\rightarrow/\times$ sur Γ)
	$R\ \Downarrow\rightarrow(\Uparrow\rho_1\times\Uparrow\rho_2)$	(commutativité $\rightarrow/\Downarrow$ sur Γ)
	$= \rightarrow\Downarrow(\Uparrow\rho_1\times\Uparrow\rho_2)$	(définition $\times$ sur Γ')
	$= \rightarrow(\rho_1\times\rho_2)$	$\Diamond$

Pour la propriété de distributivité, elle est établie par la relation *R* entre les deux termes de la preuve. Cette relation est variable suivant les algèbres et les opérateurs considérés (elle peut être $=$, $\supseteq$ ou $\subseteq$).

Si l'on considère A_{13} (resp. A_3) pour Γ et A_8 (resp. A_2) pour Γ':

- $\Uparrow$ est défini par l'union des conversions de Γ vers Γ' du §7.2; $\Downarrow$ est défini par sa réciproque. Cette réciproque n'est pas définie sur 2^{Γ} tout entier mais l'est sur son sous-ensemble correspondant aux unions de conversions du §7.2. Ce sous-ensemble est clos par $^{-1}$, $\times$ et $\rightarrow$ ce qui suffit pour établir les propriétés ci-dessus: $\Downarrow$ est toujours défini.
- les propriétés (1-2) sont valides pour celles-ci;
- les opérateurs $^{-1}$ (table 11), $\times$ [RAN92b] et $\rightarrow$ (table 14) sont définis exactement comme préconisé.

Par conséquent, comme les propriétés à démontrer sont valides dans A_{13} (resp. A_3) elles le sont aussi dans A_8 (resp. A_2).

Annexe C. Intervalles en dimension quelconque

Cette annexe présente l'extension du travail proposé pour le temps à l'espace. L'espace est modélisé comme un espace vectoriel et la généralisation se fait au travers des produits (applicable à la représentation symbolique comme à la représentation numérique). Ici, tous les exemples sont en dimension 2.

L'ordre sur la droite permet de séparer celle-ci en trois zones. Dans un espace de dimension n il permet de séparer l'espace en 3^n zones (voir figure 11).

>	NW	N	NE
=	W	=	E
<	SW	S	SE
d2/d1	<	=	>

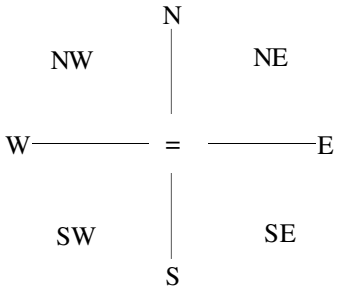


Table 15 et Figure 11. Les 3^2=9 relations symboliques exhaustives et exclusives deux à deux entre des points en dimension 2.

Celle-ci, à son tour, permet de considérer une propriété de non-croisement: elle est vérifié s'il n'y a croisement sur aucune des dimensions. En dimension n, «supérieur» (noté $>^n$) signifie que la relation liant les deux points est le produit sur chacune des dimensions de > ou = mais pas exclusivement =. En dimension 2, cela se traduit par {N NE E}. L'égalité est définie par le produit de n = (noté $=^n$). La définition du croisement ne subit alors pas de modification; une fonction f satisfait à la propriété de non croisement si et seulement si:

$$\forall x,y \; x >^n y \equiv \neg(fx <^n fy)$$

La généralisation de l'intervalle sur la droite à l'intervalle en dimension n se fait très simplement: un intervalle est caractérisé par deux bornes (quelquesoit la dimension). Les deux bornes sont des points dont l'un (la fin) est supérieur ou égal à l'autre (le début). Cet intervalle représente l'ensemble des points qui se trouvent entre les deux bornes. «Entre» s'interprète numériquement comme le fait que chaque coordonnée du point se trouve entre les coordonnées des bornes dans chacune des dimensions (par exemple debut$\leq^n$point$<^n$fin). Géométriquement, l'intervalle constitue un (hyper-)parallélépipède. La discussion sur l'appartenance ou non des bornes à l'intervalle, abordée au §2.3 est reconductible ici. Les propriétés de l'intervalle sont préservées (l'intersection de deux intervalles est toujours un intervalle, etc.).

b													
m													
o													
s													
f													
d													
e													
d^{-1}													
f^{-1}													
s^{-1}													
o^{-1}													
m^{-1}													
b^{-1}													
d_2/d_1	b^{-1}	m^{-1}	o^{-1}	s^{-1}	f^{-1}	d^{-1}	e	d	f	s	o	m	b

Table 16. Les 13^2=169 relations symboliques exhaustives et exclusives deux à deux entre des régions interprétées comme des intervalles de dimension 2. Chaque case représente la relation ■ r □ où r est le produit de deux relations entre intervalles de dimension 1.

Un ensemble de 13^n relations exhaustives et exclusives deux à deux entre intervalles en dimension *n* peut être établie. De ces relations seule celle composée de «e» est sa propre réciproque. Les autres ayant pour réciproque le produit des réciproques de leurs facteurs (symétrie par rapport à la case centrale de la table 16). La conversion de ces relations en relations sur les bornes est aussi préservée: les relations sont maintenant exprimées à l'aide d'un quadruplé de relations définies à la table 15. Par ailleurs, l'interprétation de ceci en utilisant le projeté des points (et des intervalles) sur les droites du repère de l'espace donne exactement les mêmes résultats.

La preuve de l'extension des résultats en dimension *n* se fait en considérant l'extension des résultats en dimension 1. Soit une structure $<\Gamma \times^{-1}>$, on veut prouver la validité des résultats pour la structure $<\Gamma^n \times^{n\ -1}>$ définie de telle sorte que:

$\forall r_1 \in \Gamma, \ldots r_n \in \Gamma,\ r=(r_1,\ldots r_n) \in \Gamma^n$

(définition de $^{-1}$ sur Γ^n) $\quad r^{-1} = (r_1^{-1},\ldots r_n^{-1})$

(définition de $\times$ sur Γ^n) $\quad r \times r' = \bigcup_{r_1'' \in r_1 \times r_1', \ldots r_n'' \in r_n \times r_n'} \{(r_1'',\ldots r_n'')\}$

(définition de $\rightarrow$ sur Γ^n) $\quad \rightarrow r = \bigcup_{r_1' \in \rightarrow r_1, \ldots r_n' \in \rightarrow r_n} \{(r',\ldots r_n')\}$

Ces trois opérateurs sont respectivement étendus de telle sorte que $\rho^{-1}=\{r^{-1}; r \in \rho\}$, $\rho \times \rho' = \bigcup_{r \in \rho, r' \in \rho'} r \times r'$ et $\rightarrow \rho = \bigcup_{r \in \rho} \rightarrow r$.

On peut alors prouver (sur Γ^n):

(distributivité conversion/réciprocité)

$$\rightarrow\rho^{-1} \qquad \text{(extension de } {}^{-1} \text{ sur } \Gamma^n)$$
$$= \rightarrow\{r^{-1}; r\in\rho\} \qquad \text{(extension de } \rightarrow \text{ sur } \Gamma^n)$$
$$= \bigcup_{r\in\rho} \rightarrow r^{-1} \qquad \text{(définition de } {}^{-1} \text{ sur } \Gamma^n)$$
$$= \bigcup_{r\in\rho} \rightarrow (r_1^{-1},\ldots r_n^{-1}) \qquad \text{(définition de } \rightarrow \text{ sur } \Gamma^n)$$
$$= \bigcup_{r\in\rho} \bigcup_{r'_1\in\rightarrow r_1^{-1},\ldots r'_n\rightarrow r_n^{-1}} \{(r'_1,\ldots r'_n)\} \qquad \text{(distributivité } \rightarrow/^{-1} \text{ dans } \Gamma)$$
$$= \bigcup_{r\in\rho} \bigcup_{r'_1\in(\rightarrow r_1)^{-1},\ldots r'_n\in(\rightarrow r_n)^{-1}} \{(r'_1,\ldots r'_n)\} \qquad \text{(définition de } {}^{-1} \text{ sur } \Gamma^n)$$
$$= \bigcup_{r\in\rho} \bigcup_{r'_1\in(\rightarrow r_1),\ldots r'_n\in(\rightarrow r_n)} \{(r'_1,\ldots r'_n)^{-1}\} \qquad \text{(extension de } {}^{-1} \text{ sur } \Gamma^n)$$
$$= \left[\bigcup_{r\in\rho} \bigcup_{r'_1\in\rightarrow r_1,\ldots r'_n\in\rightarrow r_n} \{(r'_1,\ldots r'_n)\}\right]^{-1} \qquad \text{(définition de } \rightarrow \text{ sur } \Gamma^n)$$
$$= \left[\bigcup_{r\in\rho} \rightarrow r\right]^{-1} \qquad \text{(extension de } \rightarrow \text{ sur } \Gamma^n)$$
$$= (\rightarrow\rho)^{-1} \qquad \Diamond$$

(inclusion dans le converti)

$$\rho \qquad \text{(définition de } \Gamma^n)$$
$$= \bigcup_{r\in\rho} \{(r_1,\ldots r_n)\} \qquad \text{(inclusion du converti dans } \Gamma)$$
$$\subseteq \bigcup_{r\in\rho} \bigcup_{r'_1\in\rightarrow r_1,\ldots r'_n\in\rightarrow r_n} \{(r'_1,\ldots r'_n)\} \qquad \text{(définition de } \rightarrow \text{ sur } \Gamma^n)$$
$$= \bigcup_{r\in\rho} \rightarrow r \qquad \text{(extension de } \rightarrow \text{ sur } \Gamma^n)$$
$$= \rightarrow\rho \qquad \Diamond$$

(distributivité de $\rightarrow$ sur $\times$)

$$\rightarrow(\rho\times\rho') \qquad \text{(extension de } \times \text{ sur } \Gamma^n)$$
$$= \rightarrow \bigcup_{r\in\rho, r'\in\rho'} r\times r' \qquad \text{(définition de } \times \text{ sur } \Gamma^n)$$
$$= \rightarrow \bigcup_{r\in\rho, r'\in\rho'} \bigcup_{r''_1\in r_1\times r'_1,\ldots r''_n\in r_n\times r'_n} \{(r''_1,\ldots r''_n)\} \qquad \text{(extension de } \rightarrow \text{ sur } \Gamma^n)$$
$$= \bigcup_{r\in\rho, r'\in\rho'} \rightarrow \bigcup_{r''_1\in r_1\times r'_1,\ldots r''_n\in r_n\times r'_n} \{(r''_1,\ldots r''_n)\} \qquad \text{(extension de } \rightarrow \text{ sur } \Gamma^n)$$
$$= \bigcup_{r\in\rho, r'\in\rho'} \bigcup_{r''_1\in\rightarrow(r_1\times r'_1),\ldots r''_n\in\rightarrow(r_n\times r'_n)} \{(r''_1,\ldots r''_n)\} \qquad \text{(distributivité de } \rightarrow \text{ sur } \times)$$
$$\mathrm{R} \bigcup_{r\in\rho, r'\in\rho'} \bigcup_{r''_1\in\rightarrow r_1\times\rightarrow r'_1,\ldots r''_n\in\rightarrow r_n\times\rightarrow r'_n} \{(r''_1,\ldots r''_n)\} \qquad \text{(extension de } \times \text{ sur } \Gamma^n)$$
$$= \bigcup_{r\in\rho, r'\in\rho'} \left[\bigcup_{r''_1\in\rightarrow r_1,\ldots r''_n\in\rightarrow r_n} \{(r''_1,\ldots r''_n)\}\right] \times \left[\bigcup_{r''_1\in\rightarrow r'_1,\ldots r''_n\in\rightarrow r'_n} \{(r''_1,\ldots r''_n)\}\right]$$
(extension de $\times$ sur Γ^n)
$$= \left[\bigcup_{r\in\rho} \bigcup_{r''_1\in\rightarrow r_1,\ldots r''_n\in\rightarrow r_n} \{(r''_1,\ldots r''_n)\}\right] \times \left[\bigcup_{r'\in\rho'} \bigcup_{r''_1\in\rightarrow r'_1,\ldots r''_n\in\rightarrow r'_n} \{(r''_1,\ldots r''_n)\}\right]$$

(définition de $\rightarrow$ sur Γ^n)

$$= \left[\bigcup_{r\in\rho} \rightarrow r\right] \times \left[\bigcup_{r'\in\rho'} \rightarrow r'\right]$$ (extension de $\rightarrow$ sur Γ^n)

$$= \rightarrow\rho \bowtie \rightarrow\rho' \qquad \Diamond$$

Pour la propriété de distributivité, elle est établie par la relation *R* entre les deux termes de la preuve. Cette relation est variable suivant les algèbres et les opérateurs considérés (elle peut être =, $\supseteq$ ou $\subseteq$).

Si l'on considère A_{13} (resp. A_3) pour Γ, comme les propriétés à démontrer sont valides pour A_{13} (resp. A_3) elles le sont aussi sur leurs produits.

Annexe D. Voisinage conceptuel

De récents travaux ont portés sur la notion de voisinage conceptuel [FRE 92]. Cette notion a été utilisée intuitivement dans la construction des tables de monté et de descente. Le lien est fait ici entre celles-ci.

Deux relations de Γ sont *conceptuellement voisines* si elles peuvent être transformées l'une en l'autre par une déformation continue des objets sur lesquels elles portent. La déformation des objets reste un terme vague. Trois systèmes sont proposés pour A_{13} et dépendent de l'opération retenue pour faire varier les intervalles.

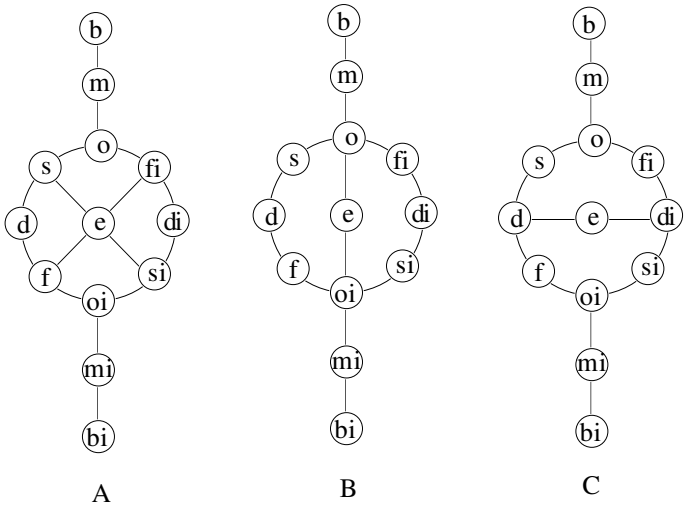


Figure 12 (d'après [FRE 92]). Graphe de voisinage entre les relations sur les intervalles temporels. Les nœuds représentent les relations et les arcs du graphes représentent le voisinage entre ces relations suivant que les intervalles sont déformés en (A) bougeant graduellement l'une des extrémités, (B) bougeant un intervalle tout en maintenant sa durée fixe et (C) en fixant le milieu d'un intervalle tout en autorisant la variation de sa durée[1].

Un *voisinage conceptuel* est définit comme un ensemble de relations de Γ dont les éléments constituent un sous-graphe connexe du graphe de voisinage de Γ.

Freska remarque que $\times_{13}$ donne des voisinages conceptuels (pour A). Indépendement, le graphe de voisinage (voir figure 13) pour A_8 est établi dans [RAN 92a, b]. Ils ont remarqué que l'application de la transitivité à deux relations retournait un ensemble de relations qui constituait un voisinage conceptuel. Cette remarque a été utilisée pour prouver la table de A_8 [RAN 92b]: quand une sortie de la table contient deux relations, on sait alors qu'elle contient au moins un chemin (dans le graphe ci-dessus) entre ces relations.

---

[1] Christian Freksa considère que les propriétés établies dans [FRE92] sont valides pour les trois types de définitions. En fait, il semble qu'elles ne le soient toutes que pour A.

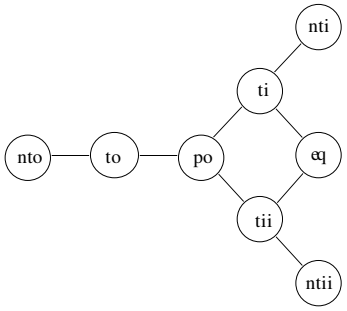


Figure 13 (d'après [RAN 92b]). Graphe de voisinage entre les relations sur les régions spatiales. Les auteurs n'ont envisagé qu'une transformation: celle correspondant à déformer continuement l'enveloppe convexe d'une région connexe. On peut remarquer que ce graphe est homomorphe au graphe A de la figure précédente (avec pour homomorphisme la fonction associant aux nœuds de A_{13} leurs équivalents en A_8 et traitant de même les arcs).

La connaissance incomplète d'une relation entre deux objets est appelée *connaissance grossière* si l'ensemble de relations en disjonction représentant cette connaissance forme un voisinage conceptuel.

La notion de connaissance grossière peut sembler voisine de celle de granularité. Cependant, elle n'est ici utilisée que pour représenter des disjonctions de relations voisines sans faire appel à une notion explicite de granularité. La notion de voisinage conceptuel a été utilisée intuitivement pour produire les tables de monté et de descente présentée ci-dessus. La méthode utilisée a consisté à prendre les deux éléments en relation et à les déformer autand que faire ce peux sans violer l'une des contraintes posées au §3.3. Dés lors, il est normal que les ensembles obtenus constituent un voisinage conceptuel. La table 17 et la figure 14 donnent deux représentations de ces voisinages.

r	b	m	o	s	f	d	e	d^{-1}	f^{-1}	s^{-1}	o^{-1}	m^{-1}	b^{-1}
${}_g\uparrow^{g'}r$													
${}^g\downarrow_{g'}r$													

Table 17. Conversion ascendante et descendante des relations entre intervalles exprimées dans le formalisme de [FRE 92] qui consiste à faire figurer les éléments de l'ensemble sur le graphe de voisinage.

Les résultats ci-dessus sont immédiatement vérifiés pour la granularité spatiale en utilisant simplement l'appauvrissement de l'algèbre qui se traduit par l'homomorphisme sur le graphe.

Freksa a transformé le système de Allen en un système muni de 10 relations qui représentent des voisinages. Ces 10 relations munies de l'opération de conjonction permettent d'exprimer tout ce qui est exprimable avec les relations de Allen munies de la disjonction. Cet ensemble de relations est clos par transitivité, il donne donc une table de transitivité s'exprimant uniquement

en fonction de ces voisinages. L'efficacité d'une telle table par rapport à la manipulation de disjonctions est évident (et aurait pu être utilisé au sein de l'annexe A).

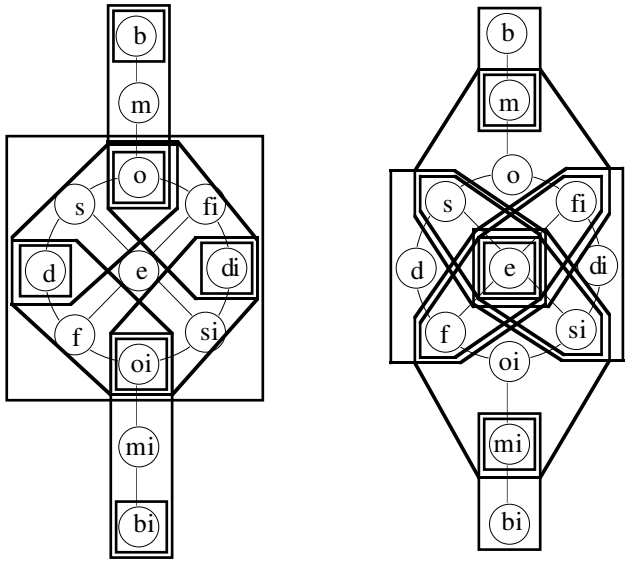


Figure 14. Le graphe de voisinage est représenté avec les ensembles correspondants aux voisinages engendrés par les opérations de conversion ascendante (à gauche) et descendante (à droite).

Lorsque les conversions entre granularités sont prises en compte, les voisinages obtenus ne sont pas les mêmes que ceux considérés comme primitifs par Christian Freksa (ils sont uniquement choisis pour des propriétés de la structure qu'ils composent). On dispose en effet des 13 relations de Allen et de 17 voisinages (disjonctions de relations primitives) dont 8 ne font pas partie de ces relations [FRE 92, fig. 7]. Par ailleurs, si cet ensemble est clos par réciprocité il ne l'est ni par conversion (ascendante/descendante), ni par transitivité.

Cependant, ceci ne permet pas de dire s'il existe ou non un ensemble de voisinages ayant de bonnes propriétés par rapport aux opérations de conversion (clos par transitivité et conversion). L'existance d'un tel ensemble devrait permettre de construire un système d'inférence temporelles entre plusieurs granularités de manière relativement efficace.