

Raisonnement temporel dans un graphe de c-intervalles

Khalil BENKOULA (*), Mahieddine DJOUDI (**), Salima BENBERNOU (*)

(*) Université des Sciences et de la Technologie
d'Oran, Institut d'Informatique, B.P. 1505 El
Menouar Oran, Algérie.

(**) Laboratoire SIC-IRCOM CNRS Département
d'Informatique de l'Université de Poitiers,
France.

Résumé

Afin de représenter la granularité des événements une nouvelle classe, nommée c-intervalle, est proposée. Elle est définie en fonction d'un ensemble de macro relations qui permettent aussi d'exprimer les assertions temporelles incomplètes. Un algorithme efficace de vérification de la consistance de ces assertions temporelles est fourni. Déterminer les relations temporelles additionnelles peut être effectué à l'aide de la méthode de propagation des contraintes. Une autre méthode utilisée nécessite l'introduction de la notion de niveau d'un noeud pour guider la recherche des relations temporelles entre les paires d'intervalles.

Mots clés : intelligence artificielle, raisonnement temporel, représentation des connaissances, intervalle de temps, raisonnement, graphe, événement, granularité.

1. Introduction

Le raisonnement temporel est un aspect important dans un grand nombre d'applications de l'intelligence artificielle (IA). Par exemple, en planification l'inconsistance (i.e. incohérence) d'un ensemble d'assertions temporelles signifie qu'une action choisie est non appropriée pour un but donné [17]. En prédiction ce type de raisonnement permet de construire sous certaines hypothèses la ou les histoires possibles correspondant à l'évolution possible de l'univers modélisé [12].

La modélisation du raisonnement temporel a conduit à de différentes propositions. Nous citerons en particulier le modèle de Allen [1, 2] basé sur le concept d'intervalle. Dans lequel, la position relative de deux intervalles de temps est exprimée par treize relations mutuellement exclusives (voir table 1); leur ensemble est nommé A_{13} .

Relation	Relation inverse	Interprétation
I_1 avant ($<$) I_2	après ($>$)	$\frac{I_1}{\quad} \quad I_2$
I_1 touche (m) I_2	touché-par (m^{-1})	$\frac{I_1}{\quad} \quad \frac{I_2}{\quad}$
I_1 recouvre (o) I_2	recouvert-par (o^{-1})	$\frac{I_1}{\quad} \quad \frac{I_2}{\quad}$
I_1 débute (s) I_2	débuté-par (s^{-1})	$\frac{I_1}{\quad} \quad I_2$
I_1 contenu-dans (d) I_2	contient (d^{-1})	$\frac{I_1}{\quad} \quad I_2$
I_1 termine (f) I_2	terminé-par (f^{-1})	$I_2 \quad \frac{I_1}{\quad}$
I_1 égale ($=$) I_2	égale ($=$)	$\frac{I_1}{\quad}$ I_2

Table 1. Les relations de base entre intervalles.

Les relations temporelles entre les paires d'intervalles peuvent ne pas être connues. Ces relations incomplètes, qualifiées de grossières par Freska [5], sont exprimées par la disjonction des treize relations de base. Un algorithme de propagation locale de contraintes a été proposé par Allen [1] pour vérifier la consistance d'un ensemble d'assertions temporelles. En fait, prouver la consistance de cet ensemble dans un tel cadre général de représentation du temps est un problème NP-complet [15]. Vilain et Kautz [15] ont défini une représentation basé sur l'instant. Dans leur approche les points de temps (i.e. instant) sont reliés par trois relations de base possibles: précède ($<$), coïncide ($=$) et succède ($>$). Cependant, un sous ensemble de relations entre les paires d'intervalles peut être translaté en conjonction de relations entre les bornes (i.e. points) inférieures et supérieures des intervalles. Des algorithmes ont été développés par Van Beek [13] qui nécessite un temps en $O(n^2)$ et un espace en $O(n^2)$ pour vérifier la consistance et un temps en $O(n^4)$ pour déterminer les relations temporelles additionnelles où n est le nombre de points.

Le modèle basé sur l'intervalle est général et formel. Cependant, la représentation des assertions temporelles sous différents niveaux de détails ou granularité permet de mieux rendre compte des descriptions d'événements. Montanari [7] a introduit la notion de macro événement pour formaliser l'opération de décomposition d'un événement en d'autres événements. La granularité a été aussi utilisée par Nokel [9] pour reconnaître d'une manière automatique les événements.

Dans cet article, on propose en premier une nouvelle classe d'intervalles, nommée c-intervalle, pour représenter la granularité des événements. Après avoir représenté les assertions temporelles dans un graphe de c-intervalles, on définit un algorithme qui nécessite un temps en $O(\max(n, m))$, où n et m sont respectivement le nombre de noeuds et d'arcs, pour vérifier leur consistance. Deux approches de raisonnement sont ensuite examinées. La première est basée sur la méthode de propagation des contraintes. Enfin, et avant de conclure, la deuxième méthode qui nécessite l'introduction de la notion de niveau d'un noeud dans le graphe est abordée.

2. Présentation du modèle

Considérer la même réalité sous différents niveaux de détails est une démarche souvent utilisée pour structurer hiérarchiquement un domaine. Deux types de granularité sont distinguées: des événements et du temps. La granularité des événements est en fonction des descriptions données. Quant à la granularité du temps, elle dépend des unités ou des échelles utilisées.

Le présent travail se concentre sur la granularité des événements qui considère à la fois les distinctions temporelles qu'on voudrait faire et les distinctions qu'on peut laisser inspecifiées. Prenons par exemple l'événement "Bénaïl prend son déjeuner" qui est décomposé en événements suivants:

manger,
parler et
boire du café.

En regardant de près chacun de ces événements composants, ils peuvent aussi être raffinés en d'autres événements. Ceci dépend de la pertinence de l'événement par rapport à notre point de vue.

Puisqu'un événement est supposé apparaître pendant un intervalle dans l'axe des réels, on peut toujours décomposer les période de temps en sous intervalles. Chaque sous intervalle correspondra à un événement composant. On obtient ainsi plusieurs niveaux de détail des assertions temporelles.

Dans l'exemple précédent, on a seulement le lien événement et événement composant, mais on n'a aucune information sur les relations temporelles entre les paires des trois derniers événements. Afin de représenter de telles assertions temporelles, un ensemble de macro relations est défini. Une macro relation est un sous ensemble des arize relations de base et elle est considérée comme la disjonction de ses relations constituantes. La plus importante macro relation est la relation "inclus", notée \subset , entre un intervalle et ses sous intervalles. Elle caractérise la fonction de décomposition d'un événement en d'autres événements. En effet, un événement composant doit apparaître durant son intervalle composé. En plus, elle exprime un lien de plus fine granularité entre les événements car elle permet de décrire un événement à un certain niveau de précision puis de raffiner cette description pour introduire des détails supplémentaires. En considérant la description de l'événement déjeuner nous avons "Manger \subset Déjeuner", "Parler \subset Déjeuner" et "Café \subset Déjeuner". Allen [1] a mentionné qu'une telle relation structure souvent les connaissances temporelles.

Les macro relations introduites comme suit :

$$\begin{aligned} \subset &= \{s, f, d, =\} & \subset^{-1} &= \{s^{-1}, f^{-1}, d^{-1}, =\} \\ \alpha &= \{<, m\} & \alpha^{-1} &= \{>, m^{-1}\} \\ \beta &= \{<, >, m, m^{-1}, o, o^{-1}, s, s^{-1}, f, f^{-1}, d, d^{-1}, =\} \end{aligned}$$

permettent de définir un nouveau ensemble noté par $A_8 = \{<, >, \alpha, \alpha^{-1}, \subset, \subset^{-1}, \beta, =\}$ dont ses éléments sont appelés des relations atomiques. La relation α (resp. α^{-1}) signifie qu'un intervalle est avant ou touche (resp. après ou touché-par) un autre intervalle. D'autre part, la relation β signifie qu'aucune relation n'est spécifiée entre les paires d'intervalles: "Parler β Café", "Manger β Café" et "Parler β Manger". Lorsque plusieurs événements composants sont temporellement inclus dans le même événement peuvent ainsi se succéder (i.e. avant, avant ou touche) ou apparaître simultanément (i.e. égale).

Les intervalles de temps utilisés vont être notés par I_1, I_2, \dots, I_n avec \mathcal{I} est l'ensemble des intervalles et \mathcal{I}_c est l'ensemble des c-intervalles.

Après l'introduction de la notion de macro relation, on peut définir maintenant d'une manière formelle la nouvelle classe de c-intervalles comme suit:

Définition : c-intervalle

Un c-intervalle I est un intervalle décomposé en n sous intervalles I_i tels que :

$$\forall i \ 1 \leq i \leq n \ I_i \subset I$$

et $\forall i, j \ 1 \leq i \leq n$ et $1 \leq j \leq n$ tel que $i \neq j$ une des relations suivantes est vérifiée:

$$\begin{aligned} &I_i < I_j \\ &I_i \alpha I_j \\ &I_i \beta I_j \\ &I_i = I_j \end{aligned}$$

En d'autres termes un c-intervalle est une période de temps composée par d'autres c-intervalles ou intervalles qui peuvent être reliés par des relations de l'ensemble A_8 . La notion de c-intervalle introduite nous permet de raffiner les événements et de représenter alors les assertions temporelles sous différents niveaux de détails.

Il est important de mentionner que les relations avant ($<$), avant ou touche (α), inclus (\subset), inconnu (β) et égale ($=$) sont transitives.

Pour indiquer qu'un ou deux intervalles composent un autre intervalle deux prédicats sont introduit comme suit:

$$\text{COMP}(I_1, I) \Leftrightarrow I_1 \text{ compose } I$$

$$\text{MEMB}_I(I_1, I_2) \Leftrightarrow \text{COMP}(I_1, I) \wedge \text{COMP}(I_2, I)$$

Formellement une décomposition temporelle peut être vue comme une fonction qui associe à chaque c-intervalle un ensemble d'intervalles de temps. Elle permet ainsi d'approcher le temps continu en des représentations de plus en plus fine sans jamais l'atteindre. Par exemple, l'action "aller chercher le livre" peut être décomposée en "aller à la bibliothèque", "faire une demande de prêt" et "prendre le livre". L'action "faire une demande de prêt" peut à son tour être décomposée en "prendre une feuille et un stylo", "écrire la demande" et "déposer la demande".

3. Représentation des assertions temporelles

Puisque les relations temporelles sont binaires, nous représentons de telles assertions dans un graphe, appelé graphe de c-intervalles, $\text{GC} = \langle \mathcal{I} \cup \mathcal{I}_c, R \rangle$ qui consiste d'un ensemble fini $(\mathcal{I} \cup \mathcal{I}_c)$ de noeuds et un ensemble R d'arcs étiquetés par les relations de l'ensemble noté par $A_S = \{<, \alpha, \beta, \subset, =\}$ tels que les deux relations suivantes sont vérifiées :

$$\forall I_1 \in \mathcal{I}_c, \forall I_2 \in (\mathcal{I} \cup \mathcal{I}_c) \quad \text{COMP}(I_2, I_1) \Leftrightarrow (I_2, \subset, I_1) \in R$$

$$\forall I_1, I_2 \in (\mathcal{I} \cup \mathcal{I}_c), \exists r \in A_S \quad (I_1, r, I_2) \in R \Rightarrow \exists I \text{ MEMB}_I(I_1, I_2)$$

Un chemin de longueur n dans le graphe est une séquence de n triplets $(I_0, r_1, I_1) \dots (I_{n-1}, r_n, I_n)$ où I_i ($0 \leq i \leq n$) sont des noeuds (i.e. c-intervalle ou intervalle) et r_j ($1 \leq j \leq n$) sont les étiquettes des arcs (i.e. relation temporelle). En particulier le chemin est un circuit lorsque I_0 et I_n sont identiques. Si toute paire de noeuds du graphe est connectée par un arc alors le graphe est dit complet.

En reprenant l'exemple du déjeuner et en ajoutant la description "Bénali mangeait en même temps qu'il parlait, après il a bu son café", on obtient le graphe donné par la figure 2.

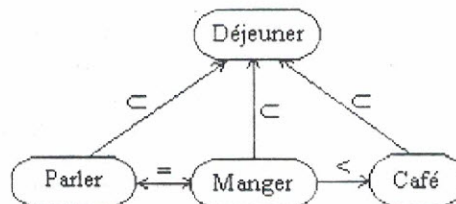


Figure 2. Représentation d'un événement composé.

Puisque la relation d'égalité est symétrique, on l'a représentée par un arc bidirectionnel. De plus, les arcs étiquetés par la relation "inconnu", qui n'apporte aucune contrainte temporelle, ont été supprimés. On peut aussi supprimer la relation d'égalité dans le graphe en identifiant toutes les paires d'intervalles qui sont nécessairement égales et les condenser dans un seul noeud. En condensant les noeuds "Parler" et "Manger", on obtient le graphe réduit donné par la figure 3.

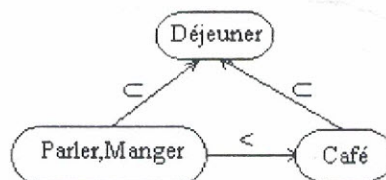


Figure 3. Graphe réduit.

Par conséquent, le graphe de c-intervalles est un graphe incomplet car les propriétés de transitivité des relations temporelles sont utilisées pour réduire le nombre des arcs.

4. Consistance des assertions temporelles

Comme dans tout système de représentation des connaissances, il est nécessaire de vérifier la consistance des assertions temporelles spécifiées. Pour cela on utilise le graphe de c-intervalles.

Considérons l'exemple de l'événement déjeuner, en insérant maintenant la description "Pendant que Bénali buvait son café, il prenait son déjeuner puis parlait". La représentation de ces informations dans un graphe est donnée par la figure 4.

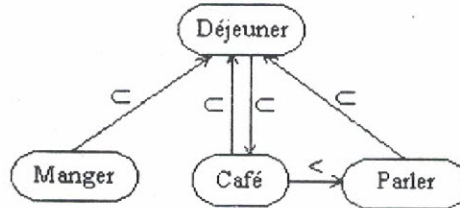


Figure 4. Un exemple de graphe inconsistant.

En analysant seulement les énoncés on peut affirmer qu'elles sont inconsistantes: l'événement déjeuner est décomposé en événement boire du café et la réciproque est fausse. Afin de vérifier la consistance d'un ensemble d'assertions temporelles deux types de chemins sont définis comme suit:

Définition : \subset -chemin

Dans un graphe de c-intervalles un chemin qui contient des arcs étiquetés uniquement par la relation d'inclusion \subset est appelé un \subset -chemin.

Définition : α -chemin

Dans un graphe de c-intervalle un chemin qui contient au moins un arc étiqueté par une des relations de précédence α ou $<$ est appelé un α -chemin.

S'il existe un \subset -chemin de I_0 à I_n alors on obtient $I_0 \subset I_n$ par la propriété de transitivité de la relation "inclus" et si de plus le \subset -chemin est un circuit alors on a $I_n \subset I_0$. Cependant, la relation "inclus" est symétrique dans le cas seulement où elle est restreinte en relation d'égalité. D'une manière similaire l'absence de α -chemin qui sont des circuits garantit la consistance du graphe car les relations α et $<$ sont irréflexives.

Propriété 1

Le graphe de c-intervalle est consistant si et seulement si

- i) un \subset -chemin de longueur n est un circuit alors $I_0 = I_n$.
- ii) il n'existe pas dans le graphe un α -chemin qui est un circuit.

La description précédente est bien inconsistante car le graphe de c-intervalle correspondant contient un circuit dont un de ses arcs est étiqueté par la relation $<$.

La détermination de la consistance d'un ensemble d'assertions temporelles nécessite en premier l'identification des composantes fortement connexes du graphe de c-intervalles. Pour cette première étape, l'algorithme de Baase [3] est utilisé. Il requiert un temps en $O(\max(n, m))$ où n et m sont respectivement le nombre de noeuds et d'arcs. Ensuite, l'existence d'une composante fortement connexe contenant deux noeuds connectés par la relation α ou $<$ est vérifiée. L'algorithme qui effectue les deux étapes citées auparavant est proposé comme suit:

ENTREE: un graphe $GC = \langle I, \cup, \cap, R \rangle$ de c-intervalles.

SORTIE: une réponse sur la consistance de GC.

Vérifier_Consistance(GC)

début

 identifier les composantes fortement connexes de GC.

 pour chaque arc (I_i, r, I_j) appartenant à R faire

 si $composante(i) = composante(j)$ et $(r = '<'$ ou $r = '\alpha')$

 alors

 retourner ("Graphe inconsistant")

 fsi

 retourner ("Graphe consistant")

fin

L'identification des composantes fortement connexes permet d'associer à chaque noeud I_i un nombre noté par *composante(i)* correspondant à sa composante fortement connexe. Nous avons m arcs qui sont partitionnés dans les différentes composantes fortement connexe. Pour chacun de ses arcs, on vérifie s'il est étiqueté par la relation α ou $<$. Ce test nécessite alors un temps en $O(m)$. Par conséquent, le temps total requis pour vérifier la consistance d'un graphe de c -intervalles est en $O(\max(n, m))$.

Propriété 2

La détermination de la consistance d'un graphe de c -intervalles nécessite un temps en $O(\max(n, m))$, où n et m sont respectivement le nombre de noeuds et d'arcs.

Une autre approche de détermination de la consistance d'un ensemble d'assertions temporelles consiste à utiliser l'algorithme de propagation des contraintes de Allen [1] ou celui de Van Beek [13]. La dernière alternative est possible dans la mesure où toutes les relations de l'ensemble A_8 peuvent être traduites en conjonction de relations entre les paires de points.

5. La déduction des relations implicites

Etant données d'une manière explicite certaines relations temporelles représentées dans un graphe de c -intervalles, on voudrait avoir la possibilité de déduire les relations additionnelles qui sont implicites dans celles données. Par exemple, si un intervalle I_1 est inclus dans I_2 , et I_2 est avant I_3 , alors I_1 doit être avant I_3 . Cette dernière relation insérée dans le graphe peut permettre de déduire d'autres relations. Afin d'effectuer cette tâche de raisonnement deux opérations l'intersection et la composition sont définies sur l'ensemble A_8 .

a) L'intersection

L'opération d'intersection notée ' \cap ' fournit la relation restreinte que deux relations atomiques entre la même paire d'intervalles vérifient. Par exemple, la relation $<$ est vérifiée si les deux relations $<$ et α sont explicitement données entre une paire d'intervalles. Ainsi, un extrait de la table de l'opération d'intersection est représenté par la figure 5.

\cap	$<$	α	$=$	β
α	$<$	α	\emptyset	α
\subset	\emptyset	\emptyset	$=$	\subset
β	$<$	α	$=$	β

Figure 5. Un extrait de la table de l'opération d'intersection.

D'une manière algorithmique, l'intersection de deux macro relations est déterminée en trouvant leurs relations constituantes communes.

b) La composition

D'autre part, l'opération de composition notée ' \cdot ' est définie entre les paires de relations qui lient trois intervalles I_1 , I_2 et I_3 . Elle est utilisée pour déterminer la relation entre I_1 et I_2 , étant données les relations entre chacun avec I_3 . La table de l'opération de composition dont un extrait est donné par la figure 6 est déterminée en utilisant la méthode de Allen [1] de distributivité de la composition par rapport à la disjonction.

\cdot	$<$	α	$=$	β
α	$<$	$<$	α	β
\subset	$<$	α	\subset	β
$=$	$<$	α	$=$	β

figure 6. Un extrait de la table de l'opération de composition.

La déduction des relations implicites peut être vue comme la détermination de la fermeture transitive (i.e. graphe minimal dans la terminologie de Van Beek) des relations entre les paires d'intervalles, en utilisant l'opération d'intersection et les règles de la table de composition. On considère pour chaque couple d'intervalles (I_1, I_2) les restrictions apportées sur les relations de I_1 et I_2 avec les autres intervalles jusqu'à aucune relation ne peut encore être restreinte dans le graphe.

Partons d'une description simple: "Dans la nuit, il avait plu mais le sol est vite redevenu sec". Dans cet énoncé trois expressions à valeur temporelle apparaissent: "Dans la nuit", "il avait plu" et "le sol est vite redevenu sec". La traduction de ces informations sous forme d'un graphe de c-intervalles est donnée par la figure 7.

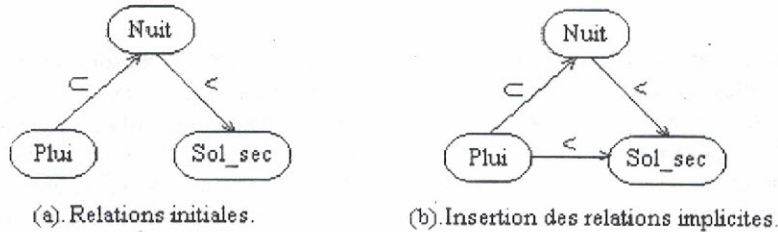


Figure 7. Représentation de la déduction des relations.

On se rend compte de l'importance de la déduction que nous avons présentée dans la figure précédente, pour insérer des relations temporelles dans le graphe alors qu'elles n'étaient pas présentes explicitement dans la description. Ceci permet de relier toute paire de noeuds du graphe par la relation restreinte. Si un intervalle de temps est durant un intervalle de référence, alors on peut ne pas le connecter avec un autre intervalle qui est avant ou après son intervalle de référence. Cette propriété qui a été décrite par Dorn [4] peut être vérifiée à l'aide de la table de l'opération de composition.

6. Requête pour les relations temporelles

Supposons que l'ensemble des relations temporelles représentées dans un graphe de c-intervalles est consistant. Nous nous intéressons maintenant aux requêtes du type "Quelle est la relation temporelle restreinte qui lie les deux intervalles I_1 et I_2 ?".

La stratégie précédente consiste à déterminer la fermeture transitive des assertions temporelles et extraire du graphe la relation restreinte entre la paire d'intervalles. Cependant, une autre approche plus efficace en espace est basée sur la partition de l'ensemble des noeuds en chaîne [4, 6]. D'une manière générale, une chaîne est un ensemble totalement ordonné par une relation. Cette approche permet de guider la recherche de la relation entre une paire d'intervalles sans déduire toutes les relations implicites inutilement.

Nous proposons de décomposer l'ensemble des intervalles en sous ensembles totalement ordonnés par la relation "inclus". Une telle chaîne d'intervalles est ainsi un \subset -chemin. Pour se faire, un nombre appelé niveau est associé à chaque noeud du graphe. Le niveau d'un noeud est égale à la longueur du plus long \subset -chemin à partir du noeud qui n'est relié à aucun autre noeud par un arc étiqueté par la relation "inclus".

En considérant l'exemple de l'événement déjeuner où la description "Avant qu'il bois son café, il a mangé" est ajoutée, les niveaux des noeuds sont donnés par la figure 8.

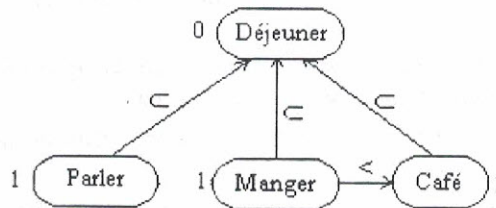


Figure 8. Les niveaux des noeuds d'un graphe de c-intervalles.

Puisque l'événement "Déjeuner" n'apparaît durant aucun autre événement alors son niveau est nul. Par contre, les événements "Parler", "Manger" et "Café" qui sont reliés à l'événement par "Déjeuner" par la relation "inclus" ont un niveau égale à 1.

L'algorithme qui détermine le niveau d'un noeud dans un graphe de c-intervalles est basé sur celui présenté par Gervini et Schubert [6]. Ils ont décomposé l'ensemble des points en chaînes où la relation d'ordre est précède ou égale (i.e. \leq). Cependant, il faut d'abord éliminer tous les circuits du graphe en identifiant toutes les paires d'intervalles qui sont nécessairement égales et les condenser dans un seul noeud. Ainsi, la relation entre une paire d'intervalles qui appartiennent à la même chaîne est déterminée en un temps constant, en comparant seulement leurs niveau respectif. Si les deux intervalles n'appartiennent pas à la même chaîne et sont inclus dans le

même c-intervalle alors une relation peut être déduite en appliquant la transitivité des relations du α -chemin qui les relie. Dans le cas où ils ne sont pas inclus dans le même c-intervalle, la relation est la même en remplaçant l'intervalle de niveau minimal par son c-intervalle jusqu'à aboutir au cas précédent. Ceci est une conséquence de la propriété d'héritage d'assertions temporelles dans la hiérarchie d'intervalles définie par la relation "inclus" [1].

7. Conclusion

Dans cet article on s'intéresse à la granularité des événements dont nous proposons un modèle basé sur une nouvelle classe d'intervalles nommée c-intervalle. Celle-ci est définie en fonction de macro relations et de sous intervalles. On peut aussi constater que la représentation des assertions temporelles dans un graphe de c-intervalles est justifiée. En effet, l'algorithme qui a été défini pour vérifier la consistance d'un ensemble d'assertions temporelles nécessite un temps linéaire. De plus, la notion de niveau d'un noeud introduite permet de déterminer la relation restreinte entre une paire d'intervalles sans déduire toutes les relations additionnelles inutilement, d'où un gain en espace considérable.

Références

- [1] James F. Allen, Maintaining Knowledge about Temporal Intervals. Communications of the ACM 1983, Vol 26, n 11, pages 832-843.
- [2] James F. Allen. Towards a General Theory of Action and Time. Artificial Intelligence 1984, pages 123-153.
- [3] Sara Baase. Computer Algorithms. Introductions to Design and Analysis. Second Edition, 1989, Addison-Wesley Publishing Company.
- [4] Jürgen Dorn. Temporal Reasoning in Sequence Graphs. 3AI 1992, pages 735-740.
- [5] Christian Freska. Temporal Reasoning based on semi-intervals. Artificial Intelligence 1992, 54, pages 199-227.
- [6] Alfonso Gervini et Lenhart Schubert. Efficient Temporal Reasoning through Timegraphs. IJCAI 1993, pages 648-654.
- [7] Angelo Montanari, Enrico Majn, Emanuele Ciapessoni, Elena Ratto. Dealing with Time Granularity in the Event Calculus. Proceeding of the international conference on the fifth generation computer systems 1992, pages 702-712.
- [8] Rober A. Morris et William D. Shaff et Linda Khatib. Path Consistency in a Network of Non-Convex Intervals. IJCAI 1993, pages 655-660.
- [9] Klaus Nökel. Temporal Matching: Recognizing Dynamic Situations from Discret Measurements. In Proceeding of the 11th IJCAI, Detroit, 1255-1260.
- [10] Wojciech Penczek. A Temporal Logic for Event Structures. Fundamenta Informaticae XI 1988, pages 297-326.
- [11] Yoav Shoham. Reified Temporal Logics : Semantical and Ontological Considerations. ECAI 1980.
- [12] Yoav Shoham et Drew McDermott. Problems in Formal Temporal Reasoning. Artificial Intelligence 1988, 36, pages 49-61.
- [13] Peter Van Beek. Exact and Approximate Reasoning about Qualitative Temporal Relations. Technical Report TR 90-29, August 1990, University of Alberta, Edmonton, Alberta, Canada.
- [14] Peter Van Beek. Reasoning about Qualitative Temporal Information. Artificial Intelligence 1992, 58, pages 297-326.
- [15] Marc Vilain et Henry Kautz. Constraint Propagation Algorithm for Temporal Reasoning. Proceeding AAAI-1986, philadelphia, pages 377-382.
- [16] R. Wylie et M. Kamel. Model Based Knowledge Organisation. Proceeding of the seventh international conference on application of artificial intelligence in engineering 1992, page 161-178.
- [17] R. Wylie et M. Kamel. Model Based Knowledge Organisation : A Framework for Constructing High-level Control System. Expert systems with applications 1992, pages 285-296.