Régularité, génération de documents, et Cyc

Regularity, document generation, and Cyc

Hafedh Mili [1], François Pachet[2]

[1] ***Laboratoire d'Acquisition et de Représentation des Connaissances***
Département d'Informatique, Université du Québec à Montréal
Case postale 8888, Succursale Centre-Ville
Montréal (Québec) H3C 3P8, CANADA
E-mail: Hafedh.Mili@uqam.ca

[2] ***LAFORIA-IBP***
Université Paris 6
4, Place Jussieu
75252 Paris Cedex 05, FRANCE
E-mail: pachet@laforia.ibp.fr

RÉSUMÉ: ***Nous nous intéressons à la modélisation des réseaux hiérarchiques, et avons développé un modèle de hiérarchies sémantiques basé sur la RÉGULARITÉ, une généralisation de l'héritage [MIL 88a]. Nous nous intéressons également à la génération de documents séquentiels structurés à partir de documents hypertextes en utilisant la sémantique des liens hypertextes pour structurer la présentation [MIL 90b]. Nous avons acquis une copie de la base de connaissances CYC [LEN 90a] dans le but de: 1) utiliser le réseau sémantique sous-jacent à CYC pour aider à la génération de textes, et 2) tester l'hypothèse de la régularité. Ironiquement, la taille gigantesque de CYC a forcé ses concepteurs d'adopter des optimisations d'implantation qui la rendent peu adaptée aux explorations logiques profondes requises par la génération de textes. Par ailleurs, l'étude des patrons de régularité dans CYC nous a amené à généraliser la notion de régularité, et à formuler un certain nombre d'hypothèses quant à la structure logique de la base de connaissances.***

ABSTRACT: ***We are interested in the modeling, building, and maintaining of hierarchical semantic nets, and have developed a model of hierarchies based on a generalization of inheritance that we called REGULARITY [MIL 88a]. We are also interested in the generation of sequential documents from hypertexts by using the semantics of the hypertext links to structure the documents [MIL 90b]. Our lab acquired a copy of the CYC knowledge base to: 1) use its underlying semantic net to support the generation of argumentative/explanatory texts, and 2) test regularity patterns within CYC. Ironically, the sheer size of CYC, which we considered to be its major strength, led its designers to a number of implementation optimizations which made it inappropriate for the kind of deep logical explorations required by the generation of argumentative texts. Further, a study of CYC 's knowledge structures led us to generalize the concept of regularity, and to correlate the «interestingness» of CYC rules to explicit violations of regularity.***

MOTS-CLÉ: ***Réseaux sémantiques hiérarchiques; génération de documents; systèmes hypertextes; systèmes experts; héritage; régularité; connaissances de sens commun.***

KEYWORDS: ***Hierarchical semantic networks; document generation; hypertext systems; expert systems; inheritance; regularity; common sense knowledge.***

1. Introduction

Dans cet article, nous présentons quelques travaux de recherche en cours au Laboratoire d'Acquisition et de Représentation des Connaissances (LARC) de l'UQAM. Le LARC regroupe une dizaine de professeur-es du département d'informatique oeuvrant dans divers aspects de l'acquisition, la modélisation, et le traitement des connaissances. Les formalismes étudiés incluent la logique, les réseaux sémantiques, les langages de cadres, et les graphes conceptuels. Parmi les applications explorées au sein de notre laboratoire, on cite la modélisation sémantique en bases de données, les systèmes tutoriels, la recherche de l'information documentaire et logicielle, et les bases de données multi-média. Le laboratoire compte une vingtaine d'étudiant(e)s gradué(e)s, et accueille régulièrement des chercheurs visiteurs et des stagiaires post-doctoraux.

Nos propres travaux en représentation de connaissances sont centrés sur les réseaux sémantiques en général, et sur les réseaux hiérarchiques en particulier. Nous sommes particulièrement intéressés par la représentation et la manipulation des hiérarchies construites *manuellement*. Dans des expériences antérieures, de telles hiérarchies s'étaient avérées utiles pour effectuer certaines tâches d'un système «intelligent» de recherche de l'information (voir par ex. [MIL 88b]). Ces expériences ont également montré le besoin de constamment

maintenir et réviser de telles hiérarchies pour prendre en compte l'évolution de la discipline qu'elles couvrent. Dans [MIL 88a], nous avons présenté un modèle de hiérarchies sémantiques qui généralise les modèles taxinomiques en remplaçant l'*héritage* des propriétés par une caractéristique plus générale que nous appelons *régularité* [MIL 90a]. Ce modèle a été testé sur deux bases de connaissances biomédicales de 100 et de 300 noeuds, respectivement, développées à partir du thésaurus *Medical Subject Headings* (MeSH) [NLM 86]. Des experts ont confirmé que les exceptions à la régularité observées dans les bases étaient dues à des incohérences dans MeSH, et que les inférences basées sur la régularité étaient correctes [MIL 88a]. Depuis, nous avons cherché à tester et valider la régularité sur des bases de connaissances plus complexes, plus volumineuses, et dans des domaines autres que la biomédecine.

Par ailleurs, nous nous intéressons aux applications des réseaux sémantiques aux systèmes hypertextes, en général, et à la génération de documents structurés à partir d'hypertextes. Grosso modo, un hypertexte est un ensemble de blocs de texte connectés par des liens, de sémantique plus ou moins riche [RAD 89]. Il existe plusieurs modèles d'hypertextes, et autant d'architectures de systèmes hypertextes correspondantes [RAD 91]. Nous nous intéressons aux hypertextes où la connexion entre les blocs de texte se fait à travers un réseau *sémantique* (vs. lexical) indépendant; dépendant des modèles, un bloc de texte peut pointer soit vers les liens soit vers les noeuds du réseau [RAD 91]. L'étude d'un ensemble de documents descriptifs (traités en Médecine) nous a permis de formuler et vérifier l'hypothèse que les auteurs structurent leur documents descriptifs selon un parcours *systématique*, *qui leur est propre*, d'un modèle du domaine de discours [MIL 90b]. En Médecine, de tels modèles existent sous forme de réseaux sémantiques de toutes sortes, dont des taxinomies de maladies, des hiérarchies anatomiques ou fonctionnelles, etc [MIL 90b]. Lorsque le modèle du domaine exhibe de la régularité, le parcours peut être exprimé et justifié de façon claire et succincte [MIL 90b]. Nous cherchons à généraliser ce modèle de génération pour le cas de documents argumentatifs. Pour de tels documents, le réseau sémantique sous-jacent contiendrait des assertions (prédicats), et le parcours en question serait basé sur une procédure de preuves [MIL 90b].

Dans le cadre d'une collaboration avec le laboratoire d'IA de la *Microelectronics and Computer Technology Corp.*(MCC), dirigé par Doug Lenat, nous avons acquis une copie de la base de connaissances CYC [Lenat et Guha, 1990] pour nos travaux de recherche. De part son volume et sa richesse, CYC nous semblait offrir une plate-forme idéale pour tester nos modèles et en explorer de nouveaux. Nous avons visé deux applications: 1) utiliser le réseau sémantique sous-jacent à CYC pour supporter la génération de textes argumentatifs et/ou explicatifs, et 2) tester l'hypothèse de la régularité sur diverses relations hiérarchiques dans CYC; nous appelerons *argumentatifs* les documents qui présentent la preuve ou *argumentation* d'une thèse donnée. Paradoxalement, à cause de la taille gigantesque de CYC, et par souci d'efficacité, les concepteurs de CYC ont du utiliser certaines «optimisations» dans leur implantation, qui nous ont rendu la tâche difficile. Premièrement, pour éviter l'explosion combinatoire, certaines aspects de la représentation, couplés avec les procédures d'inférences, font que CYC soit finalement peu adaptée aux explorations logiques profondes requises par la génération de textes argumentatifs. De même, la présence implicite de contraintes d'efficacité dans la représentation, tendant à diminuer l'aspect déclaratif des connaissances représentées, a quelquefois obscurci les patrons de régularité dans CYC. Par ailleurs, l'étude de ces patrons nous a amené à généraliser la notion de régularité, et à formuler un certain nombre d'hypothèses quant à la structure logique de la base de connaissances, montrant une fois de plus que la notion de régularité permet de maîtriser la complexité dans les bases de connaissances volumineuses.

Dans la prochaine section, nous définissons la régularité et décrivons quelques inférences basées sur la régularité. Dans la section 3, nous présentons la problématique de la génération de documents à base d' hypertextes, et expliquons pourquoi CYC nous a semblé, *a-priori*, idéale pour supporter la génération de documents argumentatifs. Dans la section 4, nous introduisons CYC; les choix ontologiques et techniques qui posent problème sont discutés dans la section 5. L'étude des patrons de régularité dans CYC est décrite dans la section 6. Nous discutons les résultats préliminaires et formulons des hypothèses de recherche dans la section 7.

2. Réseaux sémantiques hiérarchiques et régularité

Les réseaux sémantiques sont au coeur d'un grand nombre de systèmes de traitement de l'information dits «*intelligents*». Cependant, leur construction constitue un véritable goulot d'étranglement (*bottleneck*) au développement de tels systèmes. Nous nous sommes longtemps intéressés à la réutilisation de sources de connaissances existantes qui, quoique pauvres en structure par rapport aux bases-jouets[1] soigneusement construites dans les laboratoires d'IA, constituent souvent des squelettes ou échafaudages que l'on peut enrichir de façon plus ou moins automatique [MIL 88a]. Parmi ces sources, on pourra citer les *thésaurus* ou

[1]. À l'exception de CYC; voir plus loin.

***schémas de classification*, développés laborieusement par des spécialistes dans leur domaine, et utilisés pour le catalogage et le repérage de documents bibliographiques. Dans des travaux antérieurs, nous avions développé des méthodes syntaxiques/structurelles pour construire [MIL 87] et maintenir des réseaux sémantiques hiérarchiques [MIL 88b]. La maintenance de réseaux sémantiques consiste essentiellement en le placement de nouveaux concepts dans le réseau. Les résultats ont montré les limitations de telles méthodes, y compris dans des cas *en apparence* simples [MIL 88a]. Depuis, nous avons cherché à utiliser des méthodes d'augmentation/acquisition qui sont basées sur une caractérisation formelle de la sémantique des relations hiérarchiques.**

La littérature en IA regorge de modèles de représentation taxinomiques [SCH 83], [FIS 87], [LEB 86, LEB 87]. Nous distinguons deux types de modèles, que nous appellerons *axiomatiques* et *inductifs*. Le langage KL-ONE et ses dérivées exemplifient l'approche axiomatique [BRA 85]. Dans KL-ONE, la relation taxinomique («subsumption») entre concepts est basée sur un ensemble *prédéfini* de relations primitives sémantiques entre propriétés de concepts; cette définition est à base d'un algorithme de classification qui place un concept dans un réseau KL-ONE en comparant ses propriétés à celles des concepts du réseau [SCH 83]. L'approche inductive peut être illustrée par le système UNIMEM de Lebowitz [LEB 87]. UNIMEM utilise les méthodes de regroupement conceptuel («*conceptual clustering*») pour construire des structures hiérarchiques de classification. Les approches axiomatiques ont l'avantage de la clarté de leur formalismes, et de bases cognitives reconnues [BRA 85]. Cependant, elles n'ont pas la flexibilité requise pour représenter des hiérarchies sémantiques autres que les taxinomies. Les approches inductives ont cette flexibilité, mais souffrent du manque de support théorique, tant sur le plan cognitif, que sur le plan mathématique.

Dans cette section, nous décrivons un modèle de relations hiérarchiques basé sur l'observation que les relations hiérarchiques entre concepts reflètent des relations, souvent hiérarchiques, entre les propriétés des concepts. Ce phénomène, appelé *régularité*, est une généralisation de la notion d'*héritage*. Nous proposons un modèle de hiérarchies (appelées hiérarchies *Description-Contexte*, ou *DC*) qui est basé sur la notion de régularité de la même façon que les modèles taxinomiques sont basés sur la notion d'héritage. Nous décrirons par la suite certaines inférences basées sur la notion de régularité.

2.1. Exemple de régularité

Considérez la hiérarchie des maladies de l'oeil illustrée par la Figure 1. Cette hiérarchie fait partie du thésaurus *Medical Subject Headings* (MeSH) [MIL 88b]. MeSH est une structure de classification développée au sein de la *National Library of Medicine* américaine, pour les besoins de son système de dépistage de l'information MEDLARS. MeSH contient plus de 15,000 concepts groupés dans 15 catégories, y compris *Anatomical Terms* et *Diseases*. Chaque catégorie est organisée sous la forme d'une forêt (ensemble d'arbres), basée sur la relation *Moins-General-Que* (*Broader-Term*, qui est interprétée comme *Has-Broader-Term*). Cette relation est assez générale, et fait référence aux relations taxinomiques, aussi bien qu'à d'autres types de relations hiérarchiques [COU 88]. La sous-hiérarchie de gauche de la Figure 1 fait partie de la catégorie des maladies (*Diseases*).

Dans une base de connaissances médicales, une maladie peut être décrite par un nombre de propriétés telles que *Localité* ou *Organe* (pour indiquer la partie du corps qui est atteinte par la maladie), *Cause*, *Symptômes*, etc. Les *Localités* des maladies de la Figure 1 sont implicites dans leur noms. Par exemple, la *localité* de *Conjunctival Diseases* est la conjonctive (*Conjunctiva*). Si l'on représente les localités de ces maladies dans un graphe basé sur la relation *Fait-Partie-De*, nous obtenons la hiérarchie de droite de la Figure 1, qui elle-même fait partie de la catégorie *Anatomy* de MeSH. Il semble que lorsqu'une une maladie A est en relation *Moins-General-Que* avec une maladie B (cf. Figure 1-a), la *localité* de A *Fait-Partie-De* la *localité* de B. Nous dirons que la propriété *localité* est *régulière* par rapport à la relation *Fait-Partie-De*. Nous avons identifié [MIL 88a] un certain nombre de manifestations de la régularité, aussi bien dans d'autres catégories de MeSH, que dans d'autres structures de classification.

Figure 1: La sous hiérarchie des maladies des yeux et la sous hiérarchie des parties anatomiques des yeux.

2.2. Définitions

Mathématiquement parlant, la notion de régularité peut être décrite comme suit. Soit N un ensemble de concepts - représentés par des noeuds, et soit *Enfant-De* une relation hiérarchique entre les éléments de N. Une propriété des éléments de N n'est rien d'autre qu'une relation binaire entre les éléments de N et les valeurs acceptables pour cette propriété. Autrement dit, soit F une propriété, et soit P l'ensemble de valeurs acceptables pour F, on a F $\prod$ N ∞ P. Soit r une relation binaire sur P. On a r $\prod$ P ∞ P. Quand F est une fonction -- il y a une seule valeur acceptable de F pour chaque élément de N, la régularité de F peut s'exprimer par la règle suivante:

(1) (∀n1, n2 Domaine (F)) (n1 Enfant-De n2 -> (F(n1), F(n2)) r

En général, F n'est pas une fonction, auquel cas F(n) dénote le sous-ensemble de P donné par:

F(n) = { p P tel que (n, p) F }

Par exemple, si F est *Cause*, on exprime le fait que *Conjunctivitis* puisse être causée par une inflammation ou une infection, par:

Cause(Conjunctivitis) = {Inflammation, Infection }

Par conséquent, à chaque relation r ∏ P ∞ P, on associe une relation entre sous-ensembles, R ∏ 2^P ∞ 2^P, définie comme suit:

A,B ∏ P et (A, B) R + ∀ a A, ∃ b B tel que (a, b) r.

R est appelée la *relation d'ensembles* associée à r. On définit la régularité *en général* comme suit:

Définition

Soit F une propriété avec des valeurs dans P, et r une relation binaire définie sur P. F est régulière par rapport à r si, et seulement si, ∀ n1, n2 Dom(F),

(2) n1 Enfant-De n2 -> (F(n1), F(n2)) R

où R est la relation d'ensembles associée à r. F définit un homomorphisme de graphes entre (Dom(F), Enfant-De) et (U / {n Dom(F)} {F(n)}).

Quand F est une fonction, la règle (2) ci-haut est équivalente à la règle (1). Dans la suite, nous dirons que F est régulière par rapport à r ou à R, et nous dirons que r (ou R) est une *relation de régularité* de la propriété F.

2.3. Propriétés

Dans cette section, nous examinons certaines propriétés de la régularité. Premièrement, nous montrons que l'héritage est un cas particulier de la régularité. Généralement parlant, une propriété F est *héritable* si, et seulement si, pour tout n1, n2, on a:

n1 is-a n2 -> F(n1) ∏ F(n2)

où "is-a" est la notation générique pour les relations taxinomiques, et F(n) a la même signification que dans 2.2. En d'autres termes, F est héritable si elle est régulière par rapport à l'inclusion. Notez que l'inclusion peut être définie comme la relation d'ensemble associé a l'égalité, c'est à dire:

A ∏ B ∅ (∀ a A) (∃ b B) tel que a = b

Dans KL-ONE, si A «englobe» (*subsumes*) B, les rôles fonctionnels respectifs de A et B doivent satisfaire une relation parmi quatre relations prédéfinies de *modification* (modification relationships [BRA 85]). Donc, on a:

n1 is-a n2 -> (F(n1), F(n2)) R_i

où *is-a* est l'appellation standard pour les relations taxinomiques, et R_i l'une des relations de modification.

La régularité offre un mécanisme fiable pour calculer les valeurs par défaut des propriétés, de la même façon que l'héritage permet de calculer ces valeurs pour le cas des taxinomies. Ce mécanisme est appelé *expansion*. Simplement dit, expansion ajoute des paires (n,F(n)) de sorte à *conserver* la régularité de F. Formellement, étant donné une propriété F régulière par rapport à une relation R, et supposant que l'on ne connaît pas la (les) valeur(s) de F pour un concept n, expansion associe à n un ensemble S de valeurs par défaut tel que:

(3.a) (∀ g) n Enfant-De g -> (S , F(g)) R

et

(3.b) (∀ g) g Enfant-De n -> (F(g) , S) R

Si un tel ensemble existe, en prenant F(n)= S, on conserve la régularité de F par rapport à R [MIL 88a]. On peut montrer que si *n* n'a pas de descendants, ou si aucun descendant n'a de valeurs pour F, un ensemble S

satisfait (3.a) s'il est inclus dans $\leftrightarrow$ / (n Enfant-De g) R^{-1} (F(g)) où R^{-1} (F(g)) est l'ensemble {p P | $\exists$ q F(g) tel que (p, q) r} [MIL 88a]. Ce type d'expansion est appelé *expansion descendante*. Notez que l'héritage multiple dans les taxinomies peut être exprimé par une expansion descendante par rapport à la relation d'inclusion (R = $\prod$). Dans ce cas, on prendrait simplement S = $\leftrightarrow$/ (n Enfant-De g) F(g), c'est à dire, l'ensemble de valeurs par défaut est égal à l'intersection des ensembles de valeurs des ancêtres.

Contrairement à l'héritage, l'expansion nous permet de calculer les valeurs par défaut dans les deux directions. En effet, on peut montrer que si "n" n'a pas d'ancêtres, ou si aucun de ses ancêtres n'a de valeurs pour F, alors un ensemble S satisfait (3.b) si S contient $\approx$ / g Enfant-De n R(F(g)). Ce type d'expansion est appelé *expansion ascendante* [MIL 88a]. Lorsque *n* a des ancêtres *et* des descendants, un ensemble S qui satisfait (3.a) et 3.b) doit être tel que:

$$\approx_{g \text{ Enfant-De } n} R(F(g)) \prod S \prod \leftrightarrow_{n \text{ Enfant-De } g} R^{-1}(F(g))$$

Naturellement, un tel ensemble existerait si:

$$\approx_{g \text{ Enfant-De } n} R(F(g)) \prod \leftrightarrow_{n \text{ Enfant-De } g} R^{-1}(F(g))$$

L'*expansion bidirectionelle* affecterait à *n* le plus petit ensemble qui satisfait ces conditions, notamment:

$$S = \approx_{g \text{ Enfant-De } n} R(F(g))$$

L'expansion est fiable parce que les relations de régularité sont souvent plus précises que les relations de modifications de KL-ONE, par exemple, principalement parce que nous ne tentons pas de prédéfinir un ensemble de relations qui caractériseraient toutes les hiérarchies que l'on pourrait considérer. Au contraire, nous utilisons des relations qui sont spécifiques à la hiérarchie en question. Cette dépendance des données (*data dependence*) a l'avantage simultané d'une plus grande flexibilité, et d'une précision accrue.

D'un point de vue sémantique, l'expansion est justifiée dans la mesure où la régularité correspond à une propriété *fondamentale* des hiérarchies. Nous soutenons que c'est le cas: dans une hiérarchie, les relations de régularité *généralement* expriment certains aspects de la relation hiérarchique sous-jacente. En d'autres termes, les relations de régularité sont souvent des *primitives sémantiques* des relations hiérarchiques, de la même façon que les relations de modification de KL-ONE sont les primitives sémantiques des relations taxinomiques. Les hiérarchies dont les relations sous-jacentes sont exprimées par la régularité d'une seule propriété sont dites *élémentaires*. Soit F cette propriété, et R la relation de régularité de F, la condition (F(n),F(g)) R représente alors une condition à la fois nécessaire et suffisante pour que n soit un Enfant-De g. Dans ce cas, F définirait un isomorphisme de graphes entre (N,Enfant-De) et ($\approx_{n \quad Dom(F)}$ {F(n)} , R).

Pratiquement, étant donné une hiérarchie H dont les concepts sont décrits par k propriétés F_1,...,F_k régulières par rapport aux relations R_1,...,R_k, respectivement, on est amené à se poser deux questions majeures. Premièrement, quelles sont les régularités qui correspondent à de vraies primitives sémantiques? En effet, il pourrait y avoir des cas où la régularité d'une propriété F par rapport à une relation R est *fortuite*. Par exemple, si l'on représente la structure administrative d'une organisation par une hiérarchie, il sera souvent le cas que l'âge d'un fonctionnaire soit plus petit que celui de son supérieur. Auquel cas, la propriété *Age* sera régulière par rapport à la relation *Plus-Petit-Que* (<). Cependant, ceci est une simple conséquence du fait que les positions administratives sont affectées selon des capacités qui croissent avec l'age (telles que l'expérience). Mais les décisions d'engagement ne sont pas basées sur l'âge en tant que tel!.

Deuxièmement, supposons que l'on ait identifié les relations qui correspondent à de «vraies» primitives sémantiques, quelle est *l'expression exacte* de la relation hiérarchique sous-jacente, fonction de ces primitives? Nous avons proposé un modèle de hiérarchies *Description-Contexte* ou DC, qui permet de répondre à cette deuxième question [MIL 88a]. Un contexte décrit un certain point de vue sur l'ensemble des concepts de la iérarchie. Pour les fins de cet article, nous nous contenterons de mentionner que le modèle supporte un algorithme de classification de concepts qui maintient ce contexte/point de vue [MIL 88a]. Nous verrons aussi dans la section 3.2 un exemple qui illustre la *précision* du modèle DC, relativement aux modèles taxinomiques.

3 La génération de documents à partir de systèmes hypertextes

3.1 Les systèmes hypertexte

Dans son article «As we may think» [BUS 45], Vannevar Bush a proposé un engin mécanique permettant d'agencer *physiquement* plusieurs documents de sorte à pouvoir les présenter dans des séquences différentes reflétant différents liens sémantiques; cette machine était supposé imiter les processus cognitifs touchant à la mémoire à long-terme. À la différence près que les systèmes hypertextes d'aujourd'hui sont électroniques plutôt que mécaniques, son modèle et sa rationalisation ont survécu plus ou moins intacts. Simplement

parlant, un *document* hypertexte est un ensemble de blocs de texte connectés par des liens *explicites*, de sémantique plus ou moins riche [RAD 91]. Un *système* hypertexte est un logiciel permettant de visualiser, ou autrement manipuler, un document hypertexte en naviguant à travers les liens reliant ses composantes. Il existe plusieurs modèles d'hypertextes, et autant d'architectures de systèmes hypertextes correspondantes [RAD 91]. Nous nous intéressons aux hypertextes où la connexion entre les blocs de texte se fait à travers un réseau *sémantique* (vs. lexique) indépendant; dépendant des modèles, un bloc de texte peut pointer soit vers les liens soit vers les noeuds du réseau [RAD 91].

Un système hypertexte permet à ses usagers d'utiliser plusieurs chemins de navigation à travers les composantes d'un document, autres que celui imposé par la structure séquentielle des copies papier ou des fichiers électroniques séquentiels. Cependant, cette flexibilité a toujours été une arme à double tranchant: les usagers ont tendance à se perdre très rapidement. Selon Van Dam, le co-créateur du «Hypertext Editing System», l'un des premiers systèmes hypertextes:

> **«We already started getting the notion that the richer the hypertext, the greater the navigational problem. But we arranged careful demos in which we knew exactly where we had to go, and people were impressed...» [Van Dam, 1988]**

Nous soutenons que, malgré son attrait, la présumée vraisemblance cognitive du processus de navigation a été utilisée de façon abusive dans le cas des systèmes hypertextes. En effet, même en admettant le modèle réseau pour la «mémoire sémantique» [QUI 68], et la théorie navigationelle des processus d'associations/comparaisons en mémoire [COL 75], cette dernière est supposée être parallèle, subconsciente (*subconscious)*, et auto-régulée dans le sens que l' «attention» (*spreading activation*) s'atténue au fur et à mesure que l'on s'éloigne du chemin sémantique le plus court entre les deux concepts à comparer [COL 75]. Or, avec un système hypertexte, la navigation est *séquentielle, consciente/attentive*, et rien n'indique à l'usager qu'il/elle s'est égaré(e) trop loin pour le/la rappeler à l'ordre!

Ainsi, en plus des fonctionalités de navigation atomique offertes par les systèmes hypertextes, plusieurs chercheurs ont tenté d'offrir des mécanismes de navigation plus structurée où l'usager peut choisir une *stratégie* de navigation et la poursuivre dans ses explorations [RAD 91]. Ces fonctionalités n'affectent pas la flexibilité sous-jacente aux systèmes hypertextes, dans la mesure où: 1) l'usager est libre de choisir la stratégie de navigation qui lui convient, et 2) les deux types de navigation peuvent co-habiter, i.e. on permet à l'usager de «digresser», en quelque sorte, en cas de besoin. Pour savoir quelles stratégies offrir à l'usager, nous avons tenté d'identifier les stratégies utilisées par les *auteurs-mêmes* pour organiser et présenter leur matière. Ceci fait l'objet de la prochaine section.

3.2 Un modèle de documents

Les (bons) documents traditionnels sont généralement structurés de sorte à offrir une vue cohérente et logique de l'objet du discours, tant localement, e.g. au sein d'une page ou d'une sous-section, que globalement, e.g. à travers l'agencement des chapitres. Nous avons donc cherché à caractériser de telles structures de sorte à en systématiser la génération. Nous nous sommes intéressés aux types de relations présentes dans une table des matières: 1) les relations hiérarchiques entre sections et sous-sections du même documents, que nous appellerons *inclusion textuelle*, et 2) les relations qui existent entre deux sections successives, de même niveau, que nous appellerons relations de *précédence textuelle*. Pour ce faire, nous avons étudié les tables de matières de plusieurs livres et traités en médecine [MIL 90b]. Le choix du domaine de la médecine était motivé tant par l'expertise de l'un des participants, que par la maturité de la médecine en tant que discipline scientifique, et la stabilité de ses modèles.

Pour le cas des *relations de précédence textuelle*, nous avons identifié quatre types de relations: 1) relations fortuites/arbitraires, 2) relations temporelles, et 3) relations logiques. Les relations temporelles, prévalentes dans les documents descriptifs, se retrouvent lorsque l'objet d'une section précède dans le temps (ou succède à) l'objet de la section qui suit. Ceci est illustré par un classique (semble t-il!) en médecine familiale intitulé *Clinical Obstetrics*, dont les chapitres sont organisés de la manière suivante:

1. Early development of the fertilized egg
2. Physiological reactions of the mother
3. **Antepartum care**
4. **Labor**
5. **Puerperal management of the mother**
6. **Care for the neonate**

Pour ce qui est des relations logiques, on les retrouve dans les documents argumentatifs et elles seront discutées dans la section 3.3.

Pour les *relations d'inclusion textuelle*, on distingue deux grandes catégories: 1) les relations binaires, et 2) les relations n-aires. Les relations binaires relient une section à sa super-section, indépendamment des autres sous-sections du même niveau. Dans de tels cas, la relation de précédence textuelle entre sous-sections du même niveau est souvent arbitraire. Un exemple de relation binaire est la relation Fait-Partie-De ou Est-Un (is-a). Dans le *Harrison's Internal Medicine*, on retrouve l'organisation suivante pour le chapitre 8:

8. Diseases of the organ systems
8.1. Diseases of the respiratory system
8.1.1. Diseases of the upper respiratory tract
8.1.2. Diseases of the Pleura, Mediastinum, and Diaphragm
8.2. Diseases of the hepatobiliary system

Il est intéressant de noter que la relation entre «respiratory system» (ou «hepatobiliary system») et «organ system» est du type «is-a», alors que la relation entre «Upper respiratory tract» (ou «pleura, mediastinum, and diaphragm») et «respiratory system» est du type «part-of», ou, «Fait-Partie-De». En d'autres termes, la propriété «Localité» (voir section 2.1) des sections est régulière par rapport à «is-a» pour le premier niveau (8.1.∅ 8. et 8.2. ∅ 8.), et régulière par rapport à «Fait-Partie-De» pour le deuxième niveau (8.1.1. ∅ 8.1. et 8.1.2. ∅ 8.1.). Selon un modèle taxinomique, la relation entre sections et sous-sections est, indépendamment du niveau, du type «is-a». Selon notre modèle de hiérarchies DC, il s'agit là de deux relations hiérarchiques distinctes. Pour se convaincre de la différence, le lecteur pourra considérer la structure suivante où l'on a ajouté une troisième sous-section à 8.1. (8.1.3.), et constater que la relation entre 8.1.3. et 8.1. est *perceptiblement différente*[2] *de celles entre 8.1.2. (ou 8.1.1.) et 8.1.*

8. Diseases of the organ systems
8.1. Diseases of the respiratory system
8.1.1. Diseases of the upper respiratory tract
8.1.2. Diseases of the Pleura, Mediastinum, and Diaphragm
8.1.3. Diseases of the fetal respiratory system
8.2. Diseases of the hepatobiliary system

D'autres types de relations binaires relient un concept (ou sa description) à (la description de) ses propriétés, tel qu'illustré dans l'exemple suivant, extrait du même chapitre, quelques niveaux plus bas:

8.1.1.2. Diffuse infiltrative diseases of the lung
8.1.1.2.1. Pathogenesis
8.1.1.2.2. Clinical manifestations
8.1.1.2.3. Diagnosis
8.1.1.2.4. Treatment
8.1.1.2.5. Prognosis

Dans le cas des relations binaires, la relation de précédence textuelle est souvent arbitraire. Par exemple, dans le cas où les relations sont du type «is-a», l'ordre des sous-sections peut être, tout simplement, alphabétique, ou bien maintient une certaine balance entre les longueurs des sections, etc. Pour le cas des relations n-aires, la précédence textuelle entre les sous-sections est souvent *sous-jacente* à la relation n-aire. Prenons l'exemple d'un traité d'histoire. On peut s'imaginer un chapitre qui parle de l'histoire de l'Afrique du Nord, organisé comme suit:

3. L'(histoire de l') Afrique du Nord
3.1. Période pré-islamique
3.2. Conquête islamique
3.3. Colonisation Française
3.4. Mouvements d'indépendance

La relation entre une section et le chapitre est celle d'inclusion temporelle, et cette relation peut être perçue comme binaire. Cependant, si l'on impose la contrainte que l'on doit couvrir dans les sections *toute la période* implicite dans le chapitre, la relation devient pentanaire. Nous observons un phénomène semblable pour le cas

---

[2]. En fait, notre modèle d'hiérarchies DC accommode le cas où les relations hiérarchiques au sein d'une même hiérarchie sont «légèrement» différentes; les différences acceptables correspondent au cas où les relations d'un niveau sont une *spécialisation* des relations du niveau supérieur [MIL 88a]. Dans le présent exemple, cela reviendrait à dire que la relation «Fait-Partie-De» (entre «upper respiratory tract» et «respiratory system») est une spécialisation de la relation «is-a» (entre «respiratory system» et «organ system»), ce qui n'est pas le cas, d'où l'impression que «Diseases of the fetal respiratory system» n'est pas à sa place.

de relations de précédence textuelle logiques: un chapitre peut représenter une affirmation/thèse à prouver, et les différentes sections élaborent les différentes étapes de la preuve, et la table des matières peut ressembler au parcours en profondeur d'une arbre ET/OU.

Comme les exemples précédents le montrent, la structure des documents traditionnels reflète la structure du domaine du discours. La nature séquentielle des média traditionnels exige une certaine linéarisation de la structure du domaine. Comme l'ont montré les exemples des «Diseases of the organ systems» plus haut, il semblerait que plus la linéarisation est systématique, plus «cohérente» est la présentation; nous soutenons qu'une présentation cohérente facilite l'assimilation parce qu'elle crée des attentes chez le lecteur, facilitant l'intégration des nouvelles informations acquises durant la lecture. Finalement, comme illustré par les divers exemples du livre *Harrison's Internal Medicine*, les auteurs peuvent choisir de traverser/parcourir différentes relations du modèle du domaine, pour différents niveaux hiérarchiques de la structure du documents. Par exemple, la structure du chapitre 8 du *Harrison's Internal Medicine* peut s'exprimer par les trois règles suivantes:

- **La relation d'inclusion du premier niveau est basée sur la régularité de la propriété «Localité» par rapport à la relation «is-a»; la relation de précédence est alphabétique.**
- **La relation d'inclusion du deuxième niveau est basée sur la régularité de la propriété «Localité» par rapport à la relation «Fait-Partie-De»; la relation de précédence est alphabétique.**
- **La relation d'inclusion du troisième niveau est celle de concept à propriété («slot»); lorsque applicable, utiliser une relation de précédence temporelle.**

On peut s'imaginer une représentation hypertexte du *Harrison's Internal Medicine* où, finalement, juste les sous-sections de plus bas niveaux sont représentées par des blocs de texte attachés aux propriétés des concepts (ici, les maladies internes). Etant donné un langage de description/spécification de parcours/tables de matière permettant d'exprimer ceci de façon symbolique, on peut soit naviguer à l'intérieur du système hypertexte à l'aide de ce parcours, soit générer une copie «papier» (ou autre version séquentielle). Ce mode de parcours peut co-habiter avec d'autres liens tels que les références croisées, les réferences bibliographiques, etc.

3.3. Génération de documents argumentatifs

Nous appelons argumentatifs les documents qui soutiennent une thèse, contrairement aux documents descriptifs tels que les traités en médecine discutés dans la section précédente. L'aspect argumentatif vs. descriptif est *global* dans le sens qu'un document argumentatif peut (doit!) contenir des éléments descriptifs, et vice-versa. Pour reprendre l'un des exemples précédents, on peut s'attendre à trouver des justifications/explications dans les sections «Pathogenesis», ou «Diagnosis» propres aux différentes maladies, surtout que le livre s'adresse à des spécialistes. *Nous soutenons que la structure-type pour un document argumentatif résulte de la hiérarchisation, suivie de la linéarisation, d'une procédure de preuve de la thèse centrale du document.* Nous illustrons l'hypothèse par un exemple, avant d'examiner les problèmes qui peuvent se poser dans un cas pratique, et la mesure dans laquelle une base de connaissances telle que CYC peut supporter la génération de tels documents.

Considérons les expressions logiques suivantes, et supposons qu'elles soient toutes vraies:

A ∧ B ∅ C, D ∅ E, E ∅ A, F V G ∅ B, F, D

Supposons que l'on ait six blocs de texte, expliquant (ou énoncant) chacun une expression (six en tout); nous dirons que chaque bloc est *indexé* ou *catalogué* par une expression logique. Supposons que l'on ait à «dire» ou plutôt «écrire» pourquoi l'énoncé *C* est vrai. Une structure possible serait:

«C» est vrai
1. **«A» est vrai**
1.1. **«D» est vrai**
1.2. «D ∅ E» est vrai
1.3. **«E ∅ A» est vrai**
2. **«B» est vrai**
2.1. **«F» est vrai**
2.2. «F V G ∅ B»
3. **«A ∧ B ∅ C» est vrai**

Une telle structure peut être générée à partir de l'arbre de preuve de C, en utilisant quelques conventions supplémentaires. Par exemple, relativement aux relations de précédence, il a été décidé dans ce cas de procéder des prémisses aux conclusions; une autre possibilité aurait été de faire l'inverse. Pour ce qui est des relations d'inclusion, la règle générale semble dire que les sous-sections d'une section S donnée représentent:

1) la règle ayant S comme conséquent, et 2) les antécédents de la règle. On a fait exception à cette règle pour le cas de la section 1 («A»), pour ne pas avoir un arbre étroit et profond; ceci peut s'exprimer par une condition portant sur le nombre d'antécédents de la règle dont «A» est un conséquent: si le nombre est inférieur à un certain seuil, on «aplanit» l'arbre d'un niveau. Sans cette règle d'exception, la section 1. aurait eu la structure suivante:

1. **«A» est vrai**
1.1. **«E» est vrai**
1.1.1. **«D» est vrai**
1.1.2. «D ∅ E» est vrai
1.2. **«E ∅ A» est vrai**

Cet exemple montre comment des considérations esthétiques et ergonomiques peuvent entrer en jeu dans la génération de tels documents. On peut aussi envisager d'ajouter une section d'introduction à tous les niveaux pour décrire les étapes de la preuve à ce niveau. Cependant, dans des cas pratiques, les défis sont de toute autre nature. Tout d'abord, pour qu'un système soit d'une utilité pratique quelconque, il doit disposer d'une base logique considérable. Un «système argumentatif» dans un domaine spécialisé n'est rien d'autre qu'un système expert accompagné d'un module d'explication, et en tant que tel, souffre des même problèmes que les systèmes experts en ce qui concerne la génération des explications [CLA 83]. Notons, entre autres, la longueur des chaînes d'inférence. Pour le cas de INTERNIST, l'aboutissement d'un diagnostic peut impliquer le déclenchement de centaines de règles. Le nombre considérable est du, en partie, au fait que beaucoup de règles sont, finalement, très «peu intéressantes» pour des fins d'explication, mais sont indispensables pour relier les prémisses aux conclusions [CLA 83]. Les chercheurs ont tenté de développer des heuristiques permettant de filtrer certaines règles et regrouper/synthétiser le reste [CLA 83]. Notons aussi le fait que les explications-type générées par les systèmes experts sont en réalité des *justifications*, et sont d'une valeur quasi-nulle pour les non-initiés. Il est intéressant de noter que lorsque INTERNIST a été utilisé comme outil d'enseignement, il a fallu lui ajouter une composante hypertexte, qui *explique* le bien-fondé des règles [FIR 85].

Le cas des domaines non-spécialisés cause des problèmes encore plus épineux. Pour l'anecdote, l'un des auteurs a observé une étudiante en journalisme consolider les informations receuillies pour une enquête d'intérêt politico-sociologique[3]. L'étudiante avait interviewé une quarantaine de personnes/personnalités, et recueilli certaines de leur citations. Le moment venu pour rédiger, en plus de la difficulté de dégager une thèse unificatrice malgré les divergences des opinions receuillies, il fallait également retrouver, parmi les centaines de citations, celles qui illustraient les propos avancés. En simplifiant à l'extrême, si on avait indexé chaque citation par une assertion, et si on disposait d'une base de règles et d'un démonstrateur de théorèmes, il aurait été possible de soumettre la thèse centrale de l'étude comme énoncé à prouver, et laisser le démonstrateur prouver cette thèse en receuillant les citations au passage.

Nonobstant les problèmes mentionnés ci-haut (longueur des chaînes d'inférences), et ceux d'ordre linguistique/stylistique (verbaliser les règles et raccorder les différents bouts), il nous reste deux problèmes quasi-insurmontables. Premièrement, contrairement aux systèmes argumentatifs spécialisés où le domaine de discours est restreint, et les règles sont, malgré tout, essentiellement synthétiques, le système qu'il nous faut doit avoir des connaissances de sens commun pour pouvoir relier les différentes citations. Un système spécialisé peut s'en sortir avec «peu» de règles de grosse (*coarse*) granularité. Un système d'intérêt général doit avoir «beaucoup» de règles de granularité fine. Le deuxième problème est d'ordre formel, et a trait à la monotonicité du système logique sous-jacent. Tout d'abord, les citations en question se contredisent, reflétant des divergences d'opinions; un démonstrateur de théorème «monotonique» ne fonctionnerait pas. De plus, qui dit connaissances de sens commun dit logique non-monotonique.

Il nous a semblé que la base de connaissances CYC répondrait adéquatement aux deux problèmes mentionnés ci-haut: 1) CYC représente des connaissances du sens commun, et les connaissances qui y sont représentées sont de granularité assez fine, et 2) CYC utilise des méthodes d'inférences par défaut, recherchant toujours la conclusion la plus «plausible» [LEN 90]. En tant que telle, CYC nous permettrait d'explorer d'autres aspects de la génération de textes argumentatifs. Par exemple, pour l'application journalistique qui nous préoccupe, l'objectivité requiert que le document en question soit représentatif des différentes opinions exprimées. Donc, si dans le processus de prouver une thèse T, on arrive aussi à prouver ¬T, mais que la preuve de T est plus plausible/fiable que la preuve de ¬T, on peut quand même retenir T comme conclusion, tout en nuançant la conclusion par l'opinion dissonante. Autre cas de figure intéressant, que les être humains (et surtout les

---

[3]. L'enquête portait sur le rôle de l'église dans la vie sociale et politique du Québec contemporain

journalistes et politiciens) semblent traiter sans difficultés, est le cas des «preuves incomplètes» où l'on passe outre certains antécédents[4], mais où l'on nuance les conclusions.

Malheureusement, comme nous le verrons dans les sections 4, et surtout 5, CYC s'est avérée peu adaptée à ce genre d'utilisations, et les questions soulevées ci-haut ne pourront être explorées pour le moment. L'inadéquation de CYC n'est pas épistémologique; elle est principalement due à des choix d'implantation assez contraignants, mais peut être inévitables à cause de sa taille et de sa complexité.

_4 Le système Cyc

4.1. Cyc, état de l'art

Le système Cyc, développé par l'équipe de Douglas Lenat à MCC, est un système ambitieux, tentant de représenter la connaissance de «sens commun» (*common sense*). Par «sens commun», Lenat entend les connaissances en général non-explicitées, mais nécessaires à la compréhension de dictionnaires, d'encyclopédies, d'articles du *Wall Street Journal*, ou *France-Soir*, mais aussi les connaissances qui nous permettent de ne pas croire ce qu'on lit dans *Allo-Police*, ou le «tabloid» sensationnel américain *National Enquirer*. On y retrouve, ou on est capable d'y prouver, des règles du genre «les choses tombent quand on les lache», «les enfants sont moins âgés que leurs parents», «arracher un membre fait souffrir», «les être humains veulent être heureux»[5], et «après un acte de vente, l'acheteur possède l'objet». Le projet CYC, étalé sur 10 ans, a commencé en 1984. Peu de documents sont à ce jour disponibles sur l'état effectif du système. Signalons le livre [LEN 90a], qui décrit les principaux mécanismes de représentation et d'inférence, ainsi que le rapport de mi parcours [LEN 90b], qui donnent une idée de la complexité du système. Le système fait l'objet de nombreux débats dans la communauté d'intelligence artificielle. Étrangement, ces débats portent la plupart du temps sur des questions philosophiques, et se situent à un niveau abstrait. Cyc est basée sur l'hypothèse que l'intelligence ne peut exister sans une masse *critique* de connaissances, que Lenat se propose de construire *manuellement*. Ces hypothèses, discutées en détail dans [LEN 91], ont été vivement critiquées, en particulier dans le même numéro d'IA [CAN 91]. Le livre sur Cyc a aussi fait l'objet de critiques vives - et tardives - [CYC 93], auxquelles ont répondu les auteurs de la même manière, c'est à dire avec un point de vue d'ingénieur. Signalons enfin le livre [CON 94] (chapitre 4) transcrivant un atelier au cours duquel Lenat eut à défendre ses idées devant un public peu facile (McDermott, Steels, Chandrasekaran, Clancey, Mitchell, Cohen). Si le débat est le plus souvent passionnant, les problèmes techniques restent, là encore, peu abordés et les débats glissent rapidement vers des questions philosophiques touchant à la nature de la connaissance, la définition de l'intelligence, etc.

Les descriptions initiales de Cyc étaient fortement orientées vers les langages de cadres [LEN 90a]. Celles-ci ont maintenant disparu, au moins superficiellement, et Cyc est aujourd'hui décrit de manière entièrement logique, i.e. en terme *d'assertions*, portant sur des *constantes*, par des *prédicats* ou relations [GUH 94]. Bien qu'il soit difficile de mesurer la taille de la base, celle-ci est estimée à environ deux millions d'assertions (Cf. réponse de Guha et Lenat dans [CYC 93]), mentionnant environ 50,000 «constantes» et 8,000 «collections», à l'aide de 5,000 «prédicats». Ces assertions décrivent les catégories d'objets les plus fréquentes des objets concrets («tangibles») courants (chaises, lacs et voitures) aux objets abstraits («intangibles») comme les maladies, les transactions financières, et les différents buts et croyances des agents.

4.2. Niveaux de représentation et «patrons d'inférence»

Cyc utilise deux niveaux de représentation que Lenat appelle *niveau épistémologique* et *niveau heuristique,* répondant à différentes exigences du système [LEN 90b]. Le niveau épistémologique doit supporter une sémantique simple, cohérente et stable; il est essentiellement à base de logique du premier ordre, augmentée de réification et de réflexion [LEN 90b]. Le niveau heuristique, quant à lui, doit représenter les connaissances de façon à optimiser les inférences, ou toutes autres requêtes, les plus courantes, et pour ce faire, «tous les moyens» sont bons. Ainsi, l'équipe Cyc a développé une classification des inférences les plus courantes du sens commun, qui met en évidence des régularités syntaxiques pouvant être exploitées pour fins d'optimisation. On propose dans cette classification des inférences simples du type «slot inverse», ou héritage généralisé de slot, mais aussi des patrons comme le «transferThrough», qui exprime le type d'inférence suivant:

> **Si (x R y) et (y R' z) et (transferThrough R' R) alors (x R z).**

---

4. Les chercheurs intègrent ces antécédents dans les donnée/hypothèses du problème.

5. Le droit à la «... pursuit of happiness...» est ancré dans la constitution américaine.

Par exemple: «owns» transfersThrough «partOf» signifie que la propriété d'appartenance se propage aux parties de ce que l'on possède (si l'on possède une voiture, on en possède aussi toutes les parties). Une vingtaine de patrons d'inférence sont ainsi proposés, du plus simple au plus complexe. En haut de cette hiérarchie on trouve la règle Si-Alors généralisée, avec laquelle on peut tout dire ou presque, mais qui est à éviter autant que possible, car engendrant la pire des complexités.

Dans les toutes premières versions de CYC, les «acquéreur-es de connaissances» (*knowledge enterers*) devaient choisir la forme syntaxique la plus efficace pour rentrer les connaissances. Depuis, on a développé un traducteur qui saisit les règles/connaissances sous forme logique du premier ordre, mais qui les traduit vers les patrons d'inférence les plus efficaces. De même, l'interface à l'usager, appelée *interface fonctionnelle*, saisit les requêtes sous-forme de logique du premier-ordre, et se chargera de les traduire en des inférences spécialisées.

En fin de compte, le niveau épistémologique constitue l'interface à l'usager, alors que le niveau heuristique est relégué au rang d'optimisation cachée du système.

4.3. Que faire de Cyc ?

À notre connaissance, Cyc n'est toujours pas utilisée dans une application commerciale. Actuellement, l'équipe de Lenat entretient deux types de collaborations pouvant mener à des applications concrètes: 1) des ententes avec des laboratoires de recherche universitaire tels que le nôtre, et 2) des projets de transfert de technologie ou recherche et développement plus ciblés, auprès des partenaires/pourvoyeurs de fonds du projet CYC. Dans la première catégorie, on note des projets en traitement du langage naturel dans lesquels on utilise les connaissances de sens commun contenues dans CYC pour lever l 'ambiguïté de certaines constructions ayant différentes interprétations syntaxiques (voir par ex. [BAR 90]). Dans la deuxième catégorie, on note l'exemple de systèmes conseillers. Lenat cite l'exemple d'un système conseiller pour l'achat d'automobiles qui, à partir de l'âge et de la profession de l'utilisateur, est capable de déduire sa situation de famille, et le nombre probable d'enfants, et de lui proposer une break de grande capacité, etc., ou, à partir de son lieu de résidence, recommander un traitement anti-rouille (proximité de l'océan) ou une quatre-roues motrices. Selon Lenat[6], la *Central Intelligence Agency* (CIA), l'un des organismes pourvoyeurs de fonds pour le projet CYC, compte utiliser CYC pour analyser les événements politiques et sociaux dans les pays sous son observation, en tenant compte des motivations des intervenants, de leur historique, et de leur idéologie. Sans vouloir sombrer dans le macabre, un exemple de situation où de telles analyses s'avèrent utiles est le cas de violence politique non-revendiquée. Par exemple, à l'aube d'élections présidentielles, quel parti bénéficierait le plus de l'assassinat du chef de l'un des partis d'une opposition fragmentée.

Dans [LEN 91], Lenat et Guha suggèrent une dizaine d'applications types pour lesquelles Cyc devrait être particulièrement utile, y compris:

Le nettoyage de bases de données: Repérer des inconsistances dans les bases de données en associant les champs et clé à des prédicats Cyc.
Le courtage on-line: Permettant de relier vendeurs et acheteurs en fonction de leurs intérêts mutuels
La gestion automatique des savoir-faire au sein d'entreprises: Une sorte de «pages jaunes» intelligentes et informatisées
Les tableurs intelligents: Permettant d'expliquer les sens des lignes et colonnes, repérer les erreurs (typiquement, des inconsistances entre âge et emploi)
Le «marketing dirigé»: Tenant compte de l'historique des clients pour leur proposer des produits susceptibles de les intéresser,
L'intégration de bases de données: Utilisant les propriétés de «couplage» ou de «colle sémantique» de Cyc.
Les interfaces intelligentes: Ajustements automatiques divers comme le remplissage automatique de champs par défaut, etc.,
La traduction automatique de textes techniques.
La réalité artificielle «enrichie»,
Etc.

Sur le plan théorique, mis à part l'ontologie fort développée de Cyc, il nous semble que l'un des apports qui survivront à l'éventuel échec du projet, est la formalisation des contextes en Cyc [GUH 91], qui permet d'organiser la base de connaissances en *micro-théories* en proposant un certain nombre de mécanismes (dont les *lifting rules*) permettant la composition et l'héritage des micro-théories.

---

6. Communication personnelle

5. Cyc du point de vue d'un utilisateur

En tant qu'utilisateurs de ce système, nous avons rencontré plusieurs problèmes dûs soit à des omissions, soit à des choix délibérés des concepteurs de CYC. Nous commencerons par discuter des problèmes liés à la représentation. Dans la section 5.2, nous discuterons des problèmes liés aux inférences.

5.1. La vision du monde par constantes: le principe de minimalité

L'un des choix primordiaux faits dans Cyc réside dans le fait que tous les concepts représentés le sont par des constantes *explicites* du système. Ainsi on a une constante `FrenchCar` qui représente la collection[7] des voitures françaises, une autre `KoreanCar`, ainsi que `EuropeanCar`, et une constante pour chacune des voitures effectives que le système aurait à manipuler. Pour limiter le nombre de ces constantes, Lenat et Guha proposent d'appliquer un certain nombre de principes de bon sens tels que le principe de n'introduire que les constantes dont on a effectivement quelque chose à dire, autre que les «généralités» que l'on peut déduire du reste des connaissances consensuelles [LEN 90a]. Cette limitation est quasiment inévitable, vu la quantité d'informations (et donc, constantes générées) que CYC peut déduire à partir d'une affirmation simple. Par exemple, si l'on affirme que `John` est un `HumanPerson`, on ne crée pas nécessairement les constantes représentant son père, sa mère, leurs parents, son bras gauche, son lieu de résidence, etc., à moins d'avoir quelque chose de spécifique à dire la dessus, tel que «l'`Age` du `Father` de `John` est de `45`». Lenat donne par ailleurs quelques directions méthodologiques permettant de décider dans les cas «limites», de l'intérêt de réifier ou non les constantes.

Cette stratégie peut poser problème vu que: 1) CYC ne peut réifier tout seul, que dans des cas bien particuliers, et 2) les procédures d'inférences dépendent de la réification d'un certain nombre de constantes pour aboutir. Une des critiques de Elkan & Greiner [CYC 93], p. 45 consistait justement à remettre en cause la pertinence de cette réification systématique (*widespread reification*), et à comparer cette approche avec d'autres, fonctionnelles, ou par composition (*compound terms*). La réponse de Guha à cette critique ([CYC 93], p. 158), consiste simplement à affirmer que Cyc peut effectivement représenter des expressions non atomiques, comme `(mayor (Capital(Texas)))`. Le problème de fond n'est cependant pas abordé: n'existent dans Cyc que les constantes explicitement rentrées dans le système, et toute décision de non-réification de la part des concepteurs, même dûment argumentée (la constante en question est jugé inintéressante) est irrévocable. Nous verrons comment l'*expansion*, l'une des inférences supportées par la régularité, permet de réifier certaines constantes «tardivement», si celles -ci s'avèrent utiles.

5.2. La navigation en Cyc

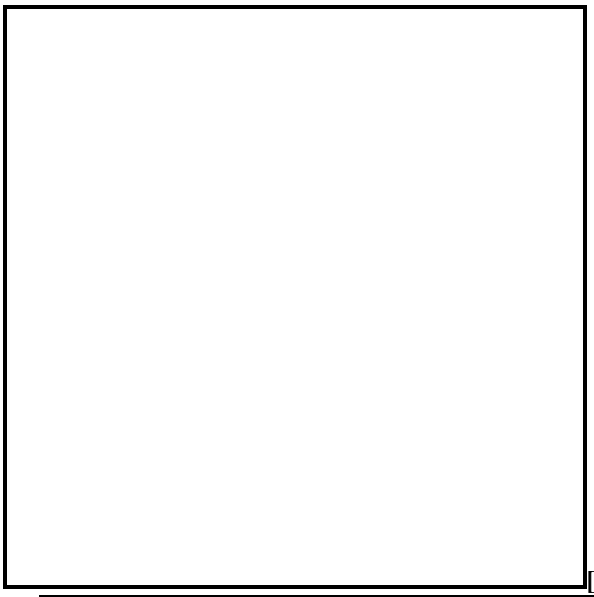
Comme toute base de connaissances de grande taille, le système Cyc souffre d'un problème de navigation, du à la grande quantité de concepts disponibles. Ce problème revêt trois aspects primordiaux: la formulation de questions, le contrôle des inférences, et la séparation de slots aux sémantiques proches.

5.2.1. Formulation des questions

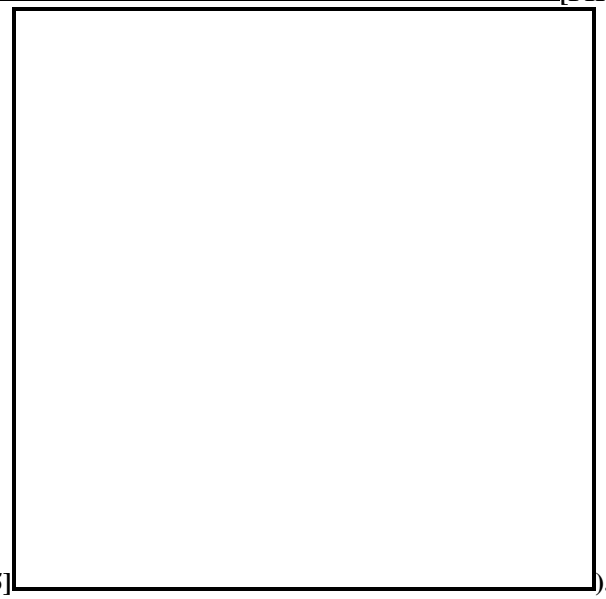
Depuis le système pionnier INTERNIST, un des premiers systèmes experts de grande taille, le problème de la formulation adéquate des questions à été identifié comme central (voir par ex.

---

[7]. **Dans Cyc, les *classes* sont représentées par des *collections* qui représentent l'ensemble de leurs instances. En d'autres termes, les classes sont représentées par leur *extensions* plutôt que leurs *intensions.* Ceci n'est sans poser des problèmes, tant épistémologiques (voir par ex. [MAI 82]) que pratiques.**



[FIR



85]). L'utilisateur d'un tel système doit formuler les questions en utilisant les termes exacts manipulés par le système, sans pouvoir bénéficier d'aucune flexibilité, et doit donc connaître les détails des structures profondes des concepts utilisés. Cette question survient dès les premières utilisations du système. Supposons par exemple que l'on ait deux instances de `HumanPerson`, `John` et `Paul`, et une instance de `Car`, et que l'on veuille dire au système que John *possède* cette instance de `Car`. Après avoir parcouru la base (par des moyens syntaxiques), on trouvera plusieurs candidats possibles pour représenter cette relation: `owns`, `possesses`, `legalOwnerOf`, `buyerOf`, `actorIn`, etc. Le problème est alors de choisir parmi ces slots possibles, lequel il nous faut utiliser. Se pose alors inévitablement la question: À quelle fin? Ceci est fonction des autres questions quel'on voudra, par la suite, poser au système. Évidemment, un système idéal devrait pouvoir se débrouiller et retrouver ses petits, mais dans notre cas, la puissance du système venant justement de sa capacité à distinguer des concepts aussi fondamentaux, il est peu raisonnable de lui demander de tout mélanger.

La solution ad hoc, pour l'instant, consiste à parcourir la base en cherchant les inférences déclenchables à partir de chacun de ces candidats, pour, *in fine*, arriver aux slots dont on est plus sûrs. Ainsi, dans cet exemple, si l'on désire faire parler au système de la relation entre posséder et l'action d'acheter/vendre, on finira par choisir le slot `ownerOf`, plutôt qu'un autre, ayant identifié des inférences permettent de lier celui-ci aux concepts représentant les processus de vente (`Selling`) et d'achat (`Buying`). En particulier, le choix de `owns` sera influencé par notre découverte d'une règle disant que le fait d'acheter entraîne la possession, et qui s'exprime justement par le slot `owns` (et non un autre). Ainsi, pour poser la bonne question (ou bien faire

la bonne assertion), nous faut-il pratiquement refaire, en chemin inverse, les inférences que l'on voudra, par la suite, éventuellement déclencher.

5.2.2. Le problème de la profondeur d'inférence

Le problème de la navigation est compliqué par le fait que les questions doivent être assorties d'indications de *profondeur*, destinées au moteur d'inférences. Toutes les inférences ne sont pas déclenchables systématiquement [LEN 90a]. En effet, Cyc peut virtuellement inférer une infinité d'informations, à partir des situations les plus simples. Pour reprendre l'exemple de `John` en tant que Human`Person`, Cyc pourrait créer (et considérer) ses parents, les parents de ses parents, etc., ainsi que ses parties anatomiques, les parties anatomiques de ses ancêtres, ou les actions typiques qu'il peut effectuer (respirer, manger, dormir, etc.). L'utilisateur est alors responsable de donner au système des directives pour l'empêcher d'errer *ad-nauseam* dans des considérations inutiles. Pour ce faire, Cyc introduit la notion de *profondeur d'accès* (access-level), associée à chaque type d'inférence (cf. section 4.2). Ainsi, l'utilisateur doit qualifier ses requêtes d'un niveau d'accès donné, et sa requête ne déclenchera que les inférences ayant ce niveau d'accès, ou des niveaux inférieurs. De plus, l'utilisateur peut préciser le *nombre* maximal de réponses souhaitées (*number of bindings*), ainsi qu'un *temps* de traitement maximal à ne pas dépasser. Souvent, le choix de ces paramètres nécessite, une fois de plus, de connaître *à l'avance* les inférences par lesquelles le système doit passer pour donner les réponses que l'on attend.

5.2.3. La sémantique des slots

Comme tout langage à base de cadres, la notion de *slot* est centrale: toutes les relations entre concepts sont représentées par des slots dont les valeurs sont des collections d'autres concepts. À chaque slot, on peut associer des propriétés à caractère "méta" dont, sa multiplicité (arité), les types (collections d'objets) permis, etc. Cependant, la sémantique même des slots n'est pas définie; un slot n'a de sens que par rapport au reste de la base. Cette vision des slots est cohérente avec les modèles cognitifs des réseaux sémantiques (voir par ex. [COL 75]): le *nom* ou *symbole* que l'on attribue à un concept ne veut rien dire en soi; c'est plutôt le parcours ou navigation que l'on peut effectuer à partir d'un noeud, ou les inférences que l'on peut tirer d'une relation, qui attribue, à un élément du réseau, sa sémantique.

En pratique, il est difficile de produire le graphe d'inférences *complet*, tel quel ou sous forme d'arbre, que l'on peut générer à partir d'un slot donné. Par contre, pour des fins de comparaison, les «implications» immédiates, ou proches, permettent de distinguer deux slots que l'on considère, ou suspecte, d'être sémantiquement proches. Pour reprendre l'exemple de la section 5.2.1., la différence entre les slots «`owns`» et «`possesses`» ne nous est apparu que lorsque nous avons comparé, par inspection, les inférences que l'on pouvait effectuer à partir de ces slots/relations. C'est une direction de recherche que nous avons explorée, en proposant des outils de visualisation, déclenchables à partir de paires de slots quelconques de la base: pour chaque slot, on identifie les règles ayant les slots comme antécédents, et on ne montre que les règles qui sont déclenchables par l'un ou l'autre, mais pas les deux. Pour le cas de «`owns`» et «`possesses`», il ne fallait pas aller loin pour voir la distinction: le slot «`owner-of`», l'inverse de «`owns`», intervenait dans une règle décrivant l'acte de vente, contrairement à «`possesses`» qui, vérification faite, parle de *détention* (*custody*) plutôt que de *propriété*.

6. Régularité dans Cyc

Cyc nous a semblé être une plate-forme idéale pour tester la notion de régularité vu qu'elle contient un très grand nombre de concepts, appartenant à un grand nombre de hiérarchies distinctes et a priori indépendantes. L'un des buts du projet Cyc n'est il pas, précisément, de trouver et d'exploiter les régularités-- au sens étymologique du terme-- du monde consensuel, afin de les codifier. En fait, beaucoup de régularités, au sens étymologique et au sens de la section 2, sont déjà représentées explicitement dans le système. En revanche, d'autres le sont implicitement, et leur explicitation permet de répondre à certaines des questions posées dans les sections 4 et 5. Comme nous le verrons, la notion de régularité permet de parler de la base de connaissances à un niveau d'abstraction élevé.

Nous commencerons par généraliser la définition de la régularité d'une caractéristique de propriétés/slots, à une caractéristique de *chemins* de propriétés. Dans la section 6.2, nous illustrons des applications de cette nouvelle définition à différents patrons d'inférence. Dans la section 6.3, nous interpréterons certaines règles d'inférences comme énonçant des exceptions, ou *cassures* à la régularité.

6.1. La régularité étendue

Dans cette section, nous introduisons une extension de la notion de régularité qui permet de parler de couples de hiérarchies liées par des chemins de propriétés quelconques, et non plus simplement par des propriétés élémentaires.

Prenons comme exemple le concept `Automobile` représentant la collection des automobiles, et ses différentes spécialisations, dont `FrenchCar`, elle-même spécialisée en `PeugeotCar` laquelle est spécialisée en `Peugeot403Car`, et `SouthKoreanCar`, spécialisée en `HyundaiCar`. Cyc propose le slot `ManufacturingOrganization`, qui associe à toute instance d' `Automobile`, le constructeur automobile correspondant, qui doit être une instance de `AutomobileManufacturer`. De même, Cyc associe à chaque constructeur, une zone géopolitique (instance de `Region`) qui correspond à la base de fabrication, ou, la «nationalité» du constructeur, par le slot `RegionOfMainActivity`. La figure 2 montre les hiérarchies (`Automobile`, `is-a`) et (`Region`, `sub-region`). La correspondance entre les deux hiérarchies pourrait être exprimée par la régularité d'une propriété que l'on exprimerait par «`RegionOfMainActivity-Of-ManufactoringOrganization`», ou, la concaténation des deux relations.

Pour généraliser la notion de régularité, nous substituons simplement les propriétés binaires par une notion plus générale de *chemin d'accès*. Un chemin d'accès est une composition de slots qui constitue un lien indirect entre un concept d'une hiérarchie *source*, et un concept d'une hiérarchie *destination*. Par ailleurs, nous allons étendre cette notion pour prendre en compte explicitement les racines des deux hiérarchies.

Figure 2. Régularité étendue. Les traits fins relient les marques aux constructeurs. Les traits épais relient les constructeurs à leur régions d'activité principales.

Définition: régularité étendue

Soient
- un ensemble de concepts A (p. ex. les spécialisations de `Automobile`),
- une relation r sur les éléments de A (par exemple *`subBrandOf` ou `is-a`*),
- un ensemble de concepts B (p. ex. les instances de *`Region`*),
- une relation r' sur les éléments de B (p. ex. *`Sub-Region`*),
- un chemin d'accès c qui fournit un lien entre les éléments de A et les éléments de B (p. ex. *`ManufacturingOrganization•RegionOfMainActivity`*),

On dira que le chemin d'accès c est régulier par rapport à A, B, r et r', ssi :
Pour tout n1, n2 **A*:***

$$(n1, n2) \; r \varnothing (c(n1), c(n2)) \quad R'$$

où R' est la relation d'ensembles associée à r'.

6.2. Régularité des patrons d'inférence spécialisés

Dans cette section, nous allons montrer comment certains patrons d'inférence spécialisés de Cyc peuvent avoir une correspondance en terme de régularité. Nous prenons deux exemples simples: le mécanisme d'héritage de slots, et les inférences temporelles.

6.2.1. Héritage de slots

Considérons la fameuse règle qui dit que tous les oiseaux volent. En Cyc, cette règle sera exprimée (au niveau heuristique), par un héritage de slot: Toutes les sous-classes de `Bird` hériteront sur leur slot `performsProcessType`, de la valeur `Flying-Locomotion`. Comme nous l'avons montré dans la section 2.2., ceci est une forme dégénérée de régularité, puisque les sous-classes de `Bird` vont se retrouver avec la même valeur. On peut cependant décrire cette assertion par le schéma de régularité suivant, où `Bird.allSpecializations` représente toutes les sous-classes de `Bird`:

Hiérarchie A : `Bird.allSpecializations`, **relation A :** `generalizations`
Hiérarchie B : `Flying-Locomotion.allSpecializations`, **relation B :** `generalizations`
Chemin d'accès : `performsProcessType`

Nous ne prenons pas en compte ici le problème des exceptions (les oiseaux qui ne volent pas); ceux-ci sont gérés par des règles explicites de défaut.

Nous pouvons exploiter cette relation de régularité dans les cas où certains concepts manquants viendraient à devenir utiles. Ce cas risque d'être fréquent étant donné que dans Cyc, on ne crée que les constantes sur lesquelles on a quelque chose à dire au moment de la création/saisie des connaissances (voir section 5.1). Le mécanisme d'expansion peut alors être utilisé pour inférer des valeurs par défaut. Par exemple le concept de `AutomobileManufacturer` peut se voir attribuer la valeur par défaut `World` pour son slot `RegionOfMainActivity`[8]. Rappelons, tout de même, que l'expansion n'est fiable que dans le cas de l'expansion bi-directionnelle (voir section 2.3), i.e. lorsque les valeurs à déduire sont bornées/limitées à la fois du «bas» et du «haut». En général, pour que l'expansion soit fiable, il faut qu'il y ait une masse critique de données correctes déjà connues [MIL 88a].

6.3. Régularité de relations entre sous abstractions temporelles

La notion de *sous-abstraction temporelle* est introduite en Cyc pour représenter plusieurs tranches temporelles d'un même concept au cours du temps. Par exemple, `LieutenantColombo` peut se voir attribuer plusieurs sous-abstractions temporelles le représentant au cours de moments caractéristiques de son existence, comme: `Colombo1erEpisode`, `Colombo2emeEpisode`. Cette relation de sous-abstraction temporelle est hiérarchique: toute sous-abstraction temporelle peut elle-même être découpée en tranches, comme: `ColomboEntrantDansLaPiece`, ou `ColomboSeGrattantLaTeteDansLaPiece`, etc. Il se trouve que cette relation de sous-abstraction temporelle possède certaines régularités lorsqu'elle est couplée à d'autres relations concernant d'autres hiérarchies de la base. Par exemple, le fait que `Colombo` possède une `Peugeot304Car` est en fait représenté en Cyc de manière plus fine: chaque sous-abstraction de Colombo possède non pas l'instance de `Peugeot304Car`, mais la tranche temporelle équivalente. Ainsi, `ColomboApresLeCrime` owns `Peugeot304ApresLeCrime`, etc. (Cf. Figure 3). Ceci est naturel, et obéit à un principe de co-extentionalité, représenté en Cyc par un certain nombres de règles, qui créent ces objets sur demande.

Ceci peut être représenté par une relation de régularité entre:

Hiérarchie A : `Person.allInstances`, **relation A :** `subAbstraction`
Hiérarchie B : `TangibleThing.allInstances`, **relation B :** `subAbstraction`
Chemin d'accès : `owns`.

Figure 4 : sous abstractions temporelles comme régularités

De même, on peut exhiber des régularités entre les relations de sous abstraction temporelle et bien d'autres relations de la base, comme la relation `parents`, `livesIn`, etc. Notons que ce n'est pas le cas pour toutes les autres relations. Par exemple, la quantité d'argent disponible n'est pas régulière par rapport au temps. La

---

8. Pas très utile ici, mais vous voyez l'idée.

encore, l'expression de ce phénomène par une relation de régularité permet de simplifier la compréhension de la base.

Il est intéressant de noter que la relation de régularité que nous venons d'exhiber n'est en effet pas toujours régulière. En particulier, c'est le cas lorsque Colombo «vend» sa voiture. Il y a alors cassure de la régularité. Nous traitons de tels cas dans la prochaine section.

6.4. Règles exprimant des irrégularités

La notion de régularité peut aussi être utilisée pour exprimer des cassures, ou des événements qui font que certaines règles ne tiennent plus. C'est justement le cas pour l'acte de vente, qui exprime le fait qu'une vente d'objet provoque nécessairement un transfert de propriété: l'objet vendu change de propriétaire par la vente. Notons que cet acte de vente nécessite plusieurs règles pour être entièrement décrit: il faut dire que lors d'une vente, le vendeur possède l'objet à vendre, qu'il ne le possède plus après, et inversement pour l'acheteur. Par ailleurs, certaines règles parlent du transfert de droit légal, et des contraintes sur la co-temporalité des objets et des acteurs. Par exemple, l'acheteur, le vendeur et l'objet se trouvent nécessairement au même endroit, et au même moment, et ainsi de suite.

Voici la règle (en syntaxe simplifiée) qui exprime le transfert de propriété d'un objet au cours d'un acte de vente: Si `anAgent performs aBuyingTransaction`, alors il possédera l'objet après la transaction:

```
Rule: OwnWhatYouBuy
IF    (allInstanceOf aTransaction Buying)
      (occursIn aTransaction aSituation)
      (nextSituation aSituation aSituation2)
      (transactionObject aTransaction anObject)
      (performedBy aTransaction anAgent)
THEN  (holdsDuring anAgent owns anObject aSituation2)
```

Bien sûr, cette règle exprime une *régularité*, au sens étymologique du terme: elle décrit, voire même *définit,* les situations normales d' actes de ventes. Le point important ici est que l'on peut interpréter cette règle comme une expression ou une *justification* d'une certaine irrégularité, celle liant la relation de sous-abstraction temporelle à la relation de possession.

7. Résumé et discussion

Nous nous intéressons à la construction et à l'entretien de réseaux sémantiques hiérarchiques, et nous avons développé un modèle de hiérarchies qui généralise les modèles taxinomiques, et qui supporte des inférences plus fiables que celles rendues possibles par l'héritage (voir section 2). Nous nous intéressons également à la génération de documents argumentatifs en général, et plus particulièrement, dans des domaines non-spécialisés qui nécessitent énormément de connaissances de bas niveau (voir section 3.3). Pour ces deux types d'explorations, CYC nous a semblé comme la plate-forme idéale pour tester nos théories et en explorer de nouvelles.

En ce qui concerne la génération de documents, nous nous sommes heurtés à un certain nombre de problèmes, dès les explorations préliminaires. Certains problèmes ne sont pas spécifiques à Cyc, et caractérisent la plupart des systèmes à base de connaissances de taille, notamment, le problème du vocabulaire (section 5.2.1): il faut savoir formuler ses questions comme il faut pour avoir les réponses que l'on cherche. La friabilité (*brittleness* [LEN 90b]) sémantique, typique des systèmes experts traditionnels caractérisés par Lenat comme idiots-savants, a été remplacée par une friabilité/fragilité lexicale. Cette dernière est, en principe, remédiable, du moment que CYC est munie d'une interface en langage naturel, qui se chargera elle même de traduire les requêtes et assertions des usagers dans la terminologie appropriée. Il est intéressant de noter que l'une des principales applications de CYC est le traitement de la langue naturelle [BAR 90]; espérons que cette dépendance mutuelle entre Cyc et le langage naturel ne soit pas une *étreinte fatale.* De par sa taille, CYC a aussi soulevé des problèmes qui lui sont spécifiques, et qui ont nécessité des optimisations au niveau de l'implantation. En particulier, la réification sélective (voir section 5.1), combinée avec la notion de niveau d'accès pour les inférences (section 5.2.2), font que l'usager doit avoir une assez bonne idée sur le résultat, et sur *la façon de l'obtenir*, pour être assuré que sa requête sera proprement traitée. Cela étant, nous ne sommes pas prêts encore à dire que la génération de documents argumentatifs, telle que décrite dans la section 3.3, n'est pas possible avec CYC; nous continuons d'expérimenter.

Pour ce qui est de la régularité, nos explorations nous ont menés vers des observations et des pistes intéressantes, quoique quelque peu différentes de celles escomptées. À la question «Cyc exhibe t-elle de la régularité?», la réponse est «oui», et probablement que chaque fois que les concepteurs de Cyc ont détecté une régularité, ils se sont débarrassés des valeurs, et l'ont codifiée par des règles. En effet, une relation de régularité exprime d'une façon concise et explicite ce qu'auraient exprimé, de manière implicite, incomplète, et diffuse, un ensemble d'assertions Cyc. Dans ce cadre, nous travaillons sur des extensions du mécanisme d'expansion (voir section 2.3) qui permettent de faire apparaître des concepts manquants, non réifiés, mais satisfaisant la régularité. Par exemple, il est possible de faire apparaître les concepts tels que "EagleLikeFlying-Locomotion" ou "SeagullLikeFlyingLocomotion", spécifiques aux catégories d'oiseaux concernées, et n'ayant pas été explicitement entrés dans la base, suivant les préceptes de minimalité (Cf. section 5.1). Par ailleurs, l'analyse des causes de régularité et d'irrégularité dans Cyc nous conduit à penser que la notion de régularité peut être utilisée de manière plus générale, dans les deux sens: 1) pour caractériser les règles de la base qui l'*enforcent* (e.g. règles gérant les constructions des sous-abstractions temporelles, règles liées à la possession en général), mais aussi 2) pour caractériser les règles de la base qui la *détruisent* (règles gouvernant les actes de vente).

Cette dernière observation semble suggérer une réponse possible à un problème lié à la génération de documents argumentatifs, notamment, le problème de savoir quels concepts, relations (slots) ou règles d'inférence, dans la trace d'une preuve, sont intéressants, et doivent être verbalisés. Nous proposons une règle simple basée sur la régularité: plus une relation est régulière, moins elle est intéressante à décorer de texte. Plus précisément, étant données deux hiérarchies et une relation (propriété ou chemin les reliant), trois cas peuvent se présenter:

- **La relation est parfaitement régulière. C'est le cas entre les différentes classes de voiture et les différents pays de fabrication. Dans le cas où cette régularité est considérée essentielle (par opposition à fortuite), la relation en question est probablement peu intéressante. Elle exprime une trivialité («les voitures françaises sont fabriquées en France»). Si elle est fortuite, elle est alors intéressante dans la mesure où elle peut révéler des caractéristiques extrinsèques des hiérarchies étudiées.**
- **La relation est «parfaitement irrégulière»: c'est le cas lorsque les hiérarchies ne sont pas du tout liées par la relation. Par exemple, il n'y a aucun rapport entre les types de voiture et les types de maladies qu'ont leurs conducteurs. Il n'y a alors rien à exploiter a priori.**
- **Les cas intermédiaires sont les plus intéressants: ceux pour lesquels une certaine régularité est à la fois manifestée et contrecarrée par des règles différentes de la base. C'est le cas de la relation de sous-abstraction temporelle et de la possession. En particulier, les irrégularités peuvent révéler soit des *incomplétudes* de la base, que le mécanisme d'*expansion* peut alors aider à combler, soit des propriétés essentielles du monde (par exemple liant l'achat et la possession). Dans ce dernier cas, il paraît intéressant de textualiser ces relations.**

Nous étudions à présent comment exploiter ces règles simples pour proposer des stratégies de parcours intelligentes, en rapport avec les directions discutées dans la section 3.3.

Remerciements**: Le travail rapporté dans cet article s'est étalé sur plusieurs années, et représente le concours de plusieurs personnes. Les travaux sur la régularité et les hiérarchies DC ont bénéficié de la supervision et direction du Prof. Roy Rada, présentement à l'Université Liverpool. Rachel Vincent a contribué l'example de l'application journalistique. Les professeur-es Gilles Gauthier, Robert Godin, Brigitte Kerhervé, Bernard Lefebvre, et Rokia Missaoui, tous/toutes membres du laboratoire LARC, ont: 1) chaleureusement accueilli François Pachet durant son stage post-doctoral, financé par l'INRIA, 2) tous, plus le Professeur Jean-François Perrot du LAFORIA, ont, inlassablement redressé la barre du projet durant ses multiples dérives, de par leur questions, critiques, et suggestions, et 3) relu et commenté différentes versions de cet article; ils ne sont toutefois pas responsables pour les erreurs, omissions, ou idées fantaisistes qui auraient survécu.**

Bibliographie

[BAR 90] Barnett J., Knight K., Mani I., et Rich E., **«Knowledge and natural language processing,»** ***Communications of the ACM*****, vol. 33, no: 8, pp. 50-71, August 1990.**

[BRA 85] Brachman R J. et Schmolze J. G., **"An Overview of the KL-One Knowledge Representation System,"** ***Cognitive Science*****, vol. 9, pp. 171-216, 1985.**

[BUS 45] Bush, V., **«As we may think,»** ***Atlantic Monthly*****, (July 1945), pp. 101-108.**

[CAN 91] Cantwell Smith B., **"The owl and the electric encyclopedia,"** ***Artificial Intelligence*****, vol. 47, pp. 251-288, 1991.**

[CLA 83] Clancey, W., **«The Epistemology of a rule-based expert system: a framework for explanation,»** ***Artificial Intelligence*****, vol. 20, No 3 (1983), pp. 215-251.**

[COL 75] Collins A. M., Loftus E. F., **«A spreading activation theory of semantic processing,»,** ***Psychological review*****, vol. 82, 1975, pp. 407-428.**

[CON 94] **«The Knowledge Level in Expert Systems: Conversations and Commentary,» in** ***Perspectives in Artificial Intelligence*****, Eds L. Steels and J. McDermott, Academic Press, 1994.**

[COU 88] Council National Library and Information**,** ***Guidelines for Thesaurus Structure, Construction and Use*****, Technical Report American National Standards Institute, 1988.**

[CYC 93] «**Book reviews and response from Lenat & Guha,»** ***Artificial Intelligence*****, vol. 61, pp. 37-181, 1993.**

[FIR 85] First M. B., Soffer L. J. et Miller R.A., **"QUICK Index of Caduceus Knowledge: Using the Internist-1/Caduceus Knowledge Base as an Electronic Textbook of Medicine,"** ***Computers and Biomedical Research*****, vol. 18, pp. 137-165, 1985.**

[FIS 87] Fisher D. H., **"Knowledge Acquisition via Incremental Concept Formation,"** ***Machine Learning*****, vol. 2, pp. 139-172, 1987.**

[GUH 91] Guha R.V., ***Contexts: A formalization and some applications.*****, Technical Report ACT-CYC-423-91, MCC technical report, 1991.**

[GUH 94] Guha R. V. et Lenat D. B**, "Enabling Agents to Work Together,"** ***Communications of the ACM*****, vol. 37, pp. 127-142, 1994.**

[LEN 90b] Lenat D. B, Guha R. V., Pittman K. , Pratt D. et Shepherd M., **"Cyc: Towards programs with common sense,"** ***Communications of the ACM*****, vol. 33, no: 8, pp. 30-49, August 1990.**

[LEB 86] Lebowitz M., **"An Experiment in Intelligent Formation Systems: RESEARCHER," in** ***Intelligent Information Systems: Progress and Prospects*****, E. H. Limited, Ed., pp. 127-150, 1986.**

[LEB 87] Lebowitz M., **"Experiments with Incremental Concept Formation: UNIMEM,"** ***Machine Learning*****, vol. 2, pp. 103-138, 1987.**

[LEN 91] Lenat D. et Guha R.V., ***Ideas for Appling Cyc*****, Technical Report ACT-CYC-407-91, MCC technical report, 1991.**

[LEN 91] Lenat D.B. et Feigenbaum E.A., **"On the thresholds of knowledge,"** ***Artificial Intelligence*****, vol. 47, pp. 185-250, 1991.**

[LEN 90a] Lenat D.B. et Guha R.V., ***Building large knowledge-based systems. Representation and Inference in the Cyc project*****. Addison-Wesley, 1990.**

[MAI 82] Maida A. S., Shapiro S. C.**, «Intensional concepts in propositional semantic networks,»** ***Cognitive Science*****, vol. 6, 1982, pp. 291-330.**

[MAY 93] Mayfield J. and Nicholas C., **"SNITCH: augmenting hypertext documents with a semantic net,"** ***International Journal of Intelligent and Cooperative Information Systems*****, vol. 2, pp. 335-351, 1993.**

[MIL 88a] Mili H., ***Building and Maintaining Hierarchical Semantic Nets*****, Ph.D. thesis, George Washington University, Washington, D.C., 1988.**

[MIL 87] Mili H. et Rada R., **"Building a Knowledge Base for Information Retrieval," in** ***Proceedings Third Annual Expert Systems in Government Conference*****, pp. 12-18, 1987.**

[MIL 88b] Mili H. et Rada R., **«Merging Thesauri: Principles and Evaluation,»** ***IEEE Transactions on Pattern Analysis and Machine Intelligence*****, vol. 10, pp. 204-220, 1988.**

[MIL 90a] Mili H. et Rada R., **"Generalizing Inheritance to Fuzzy Regularity,"** ***IEEE Transactions on Systems, Man, and Cybernetics,*** **vol 20, No 5, Sept/Oct 1990, pp. 1184-1198.**

[MIL 90b] Mili H. et Rada, R., **"Medical Expertext as Regularity in semantic nets,"** ***Artifical Intelligence in Medicine*****, vol. 2, pp. 217-229, 1990**

[MIL 92] Mili H. et Rada R., **"A Model of hierarchies Based on Graph Homomorphisms,"** ***Computers and Mathematics with Applications*****, vol. 23, pp. 343-361, 1992.**

[QUI 68] Quillian J. R., **«Semantic memory,» in** ***Semantic Information Processing*****, Eds Marvin Minsky, MIT Press, 1968.**

[RAD 89] Rada R., **"Writing and Reading Hypertext: An Overview,"** ***Journal of the American Society of Information Science*****, vol. 40, pp. 164-171, 1989.**