Mili, H. and Pachet, F. "Regularity, document generation and Cyc", in *ExpertMedia - Expert Systems and Hypermedia*, Edited by R. Rada and K. Tochtermann, World Scientific Publishing, pp. 171-206, 1995

Exploiting regularity in Cyc for text generation

François Pachet, Hafedh Mili
Department of Mathematics and Computer Science
University of Quebec at Montreal
P.O. Box 8888, Station A, Montréal, QBC 3P8, CANADA

E-mail : pachet/mili@larc.info.uqam.ca

abstract

In order to provide hypertext with text generation facilities we combine hypertext with knowledge bases and create expertext systems. This paper studies a particular characteristics of hierarchical semantic nets called regularity, which has proved to be a very valuable tool for expertext construction and use. We show how regularity can be transposed in the context of the Cyc knowledge base to provide an operational tool for understanding the knowledge base intimate topology. Using radical and provoking hypothesis i) general if-then rules express *irregularities*, and ii) "the more regular, the less interesting", we associate irregularity to text units relevance, to assist expertext users in a number of way, including finding relevant places to input text, helping formulate reading or writing plans, understand the overall system, and suggesting modifications to the knowledge base designers.

1. From hypertext to expertext

The combination of hypertext technologies with expert systems techniques is called "expertext system" [Rada 89]. Following [Streitz 89] and [Wand & al 91], we think that writing is a complex problem solving activity whose production can be viewed as an external representation of internal knowledge structures of the author. This study is part of a general study on building expertext from existing knowledge bases, aimed at studying the nature of writing plans [Rada & Barlow 89] as hypertext traversal strategies [Mili&Rada 88, 90a]. We use the Cyc knowledge base [Lenat & Guha 90] as a testbed for experimenting with our techniques, and as a basis for the construction of expertext.

This paper is based on two observations resulting from the study of the Cyc system in the context of expertext construction. The Cyc system contains an enormous amount of information and intelligence but uses a variety of knowledge representation techniques which make it hard to understand at a global level. By trying to use the system and manage its complexity, we came to a paradoxical observation concerning the nature of the knowledge embodied in Cyc, i.e., that certain types of knowledge tend to *increase regularity*, in a precise and computable sense, whereas others tend to *break those regularities*. Applying regularity calculus to Cyc, the second observation is that for most relations explicitly present in the system, "the more regular" a relation, the less interesting it is as a potential text unit holder.

The paper is organized as follows. We will first introduce briefly the Cyc system and its main characteristics with regard to expertext construction, and recall the definition and main results of the regularity calculus as defined in hierarchical semantic nets. We will then apply the notion of regularity to the Cyc system, and show how various inference patterns of the system have corresponding regularity properties. Finally we discuss the application of the regularity hypothesis to expertext construction.

2. The Cyc system

The Cyc knowledge base [Lenat&Guha 90] is acknowledged to be one of the most ambitious attempt to model common sense knowledge and reasoning. It contains currently an equivalent of 2 millions assertions, represented as a set of frames (around 50000) with numerous sophisticated inference mechanisms. The Cyc knowledge base contains the description of most common categories such as things, tangible and intangible objects (from tables, lakes and geographical countries to abstract entities such as diseases, financial transactions, beliefs and agent goals). Because of its size and general cohesiveness, it is unquestionably an ideal testbed for experimenting with expertext techniques.

2.1. A classification of inference patterns

Cyc introduces a classification of inference mechanisms that allows the system to take into account syntactic regularities of commonly used inference patterns. These specialized inference mechanisms include inverse slots maintenance, generalized slot inheritance, 'transfer through' relations, transitive closure computations. The most general inference mechanism, the If-then rule is avoided as much as possible. Cyc's distinction between specialized inference features and the general if-then rule is based on technical considerations (efficiency, ease of truth maintenance), and thus is hidden from the end-user, who access the system by a functional interface which includes an automatic translator between requests expressed in first order logic, and the various frame-representations and inference mechanisms. However, we'll see that this distinction may be used to highlight some important properties of the knowledge base with respect to regularity.

2.2. The navigation problem in Cyc

From the point of view of hypertext construction, the Cyc knowledge base has the same navigation problem than most big knowledge bases. This problem has three main sides: formulating questions, controlling the engine, and understanding slot semantics.

1. Formulation of questions

As in the INTERNIST system [First & al. 85], one of the first large-scale expert system, the problem of formulating the questions is crucial. Formulating a question in Cyc requires a deep understanding of its internal structures. The user has to formulate the questions exactly in the terms understood by Cyc, or nothing wil happen. Supposing for instance two instances of HumanPerson, John and Paul, and an instance of Car, and wanting to represent the fact that John owns the car, which links do we use between John and the car ? After browsing through the knowledge base, several candidates (such as owns, possesses, legalOwnerOf, buyerOf, actorIn) seem equally tempting. The problem is to find which slot is the most important, i.e. will allow eventually to make inferences so as to fill other similar slots if needed. Which slot means what we mean ? and incidentally what exactly do we mean ?

2. The access-level problem

In Cyc this navigation problem is complicated by the fact that not all knowledge is represented at a given time. The actual knowledge Cyc may infer is potentially infinite: billions of thing may be inferred from even the simplest situations. For instance describing a situation about "John" doing nothing could virtually lead to the creation of an infinite number of utterly uninteresting people (John's parents, John's parent's parents and so on), anatomical objects (such as John's body parts : its cornea, head, lungs, fingers, as well as John's ancestors ' body parts) or performable actions (John can breath, drink, eat, sing, understand). The user is responsible for giving Cyc directions to prevent the system performing uninteresting inferences ad libitum. Cyc introduces the notion of access-level to avoid certain inferences to be systematically performed or objects systematically created. The access-level is presently decided by the user, as well as the number of answers to look for (number of bindings), and the maximum time to spend on a question and various kinds of environment parameters. In many cases, choosing the right access-level and other parameters can only be done rightfully by knowing in advance which inferences will be interesting.

3 The semantics of slots in Cyc

Mili, H. and Pachet, F. "Regularity, document generation and Cyc", in *ExpertMedia - Expert Systems and Hypermedia*, Edited by R. Rada and K. Tochtermann, World Scientific Publishing, pp. 171-206, 1995

As in most frame based systems, the semantics of the knowledge base, is based on the "slot" paradigm. All relations between concepts are represented by slots. Slot values are always collections of other frames. Several kinds of meta information may be represented for a slot such as the type of its values, its arity, or more sophisticated constraints (e.g. a constraint stating that a person's age is always less than its parent's ages). However, slots in themselves do not carry any semantic information. Neither the name of the slot, which is purely arbitrary, or the type of the value of the slot or any other meta information that may be present in the knowledge base is enough to give the slot the semantic wanted by the knowledge base builder. The only way to give a slot some semantics or to understand it is to browse through the tree of possible inferences from a slot given a particular viewpoint. For example, from the viewpoint of financialTransactions, the slot 'owns' is more appropriate than, say, the slot possesses, simply because of a set of rules that state things about the value of this slot before and after a particular transaction.

3. Hierarchical semantics nets and regularity

By examining man-made conceptual hierarchies such as MeSH (Medical Subject Headings) and SNOMED (Systematized NOmenclature of MEDicine) [Lester&Rada 87], Mili & Rada identified mappings between hierarchies of different, but related, conceptual domains such that two hierarchically-related concepts in one hierarchy mapped to two hierarchically-related concepts in another hierarchy. They called this correspondence regularity [Mili & Rada 90a, 90b, 90c].

Hierarchical semantic nets are semantic nets whose links represent hierarchical relationships. A common form of hierarchical relationships is 'is-a'. Roughly speaking, 'is-a' may be used to relate an individual to the class to which it belongs, as in 'John is-a Man', in which case John inherits properties from the class Man. Inheritance is a special case of a more general pattern called *regularity*. Consider the conceptual hierarchies in Figure 1.

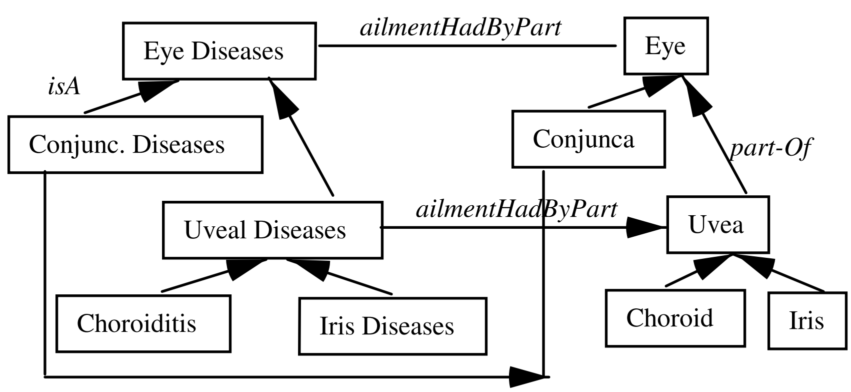

Figure 1: The sub hierarchy of eye diseases and the anatomical hierarchy of the eye.

The 'ailmentHadByPart' links associate each eye disease with the organ it affects. Figure 1 shows that whenever a disease A isA disease B, the 'ailmentHadByPart' of disease A is partOf the 'ailmentHadByPart' of disease B. This is a prototypical case of regularity : we say that the 'ailmentHadByPart' property is regular with respect to the Part-Whole relation.

Mathematically, regularity can be characterized as follows. Let N be the set of nodes in a hierarchy, and ``Lower-Than'' a hierarchical relationship between elements of N. The name 'Lower-Than' may refer to any kind of hierarchical relationship, be it 'Broader-Term', 'is-a', or any other relationship. A property (or attribute) of the concepts in N can be seen as a binary relation between the elements of N and the permissible values for that property.

Let F be a property, and P the set of permissible values for F. We have $F \subseteq N * P$. Let r be a binary relation defined in P. We have $r \subseteq P * P$. A property F that is a function from N to P is said to be

single-valued. It is often the case that a node has more than one value for a given property. We generalize regularity to the case of multi-valued properties as follows:
With each relation $r \subseteq P * P$, we associate a relation $R \subseteq 2^P * 2^P$, which is defined as:

$$A, B \subseteq P \text{ and } (A, B) \in R \equiv \forall a \in A, \exists b \in B \text{ such that } (a, b) \in r.$$

R is called the set relation associated with r.

Definition.
Let F be a multi-valued property, $r \subseteq P * P$. F is regular with respect to r iff, $\forall$ (n1, n2) $\in$ Domain (F),

$$\text{n1 LowerThan n2} \rightarrow (F\ (n1), F(n2)) \in R \qquad \text{(Reg)}$$

where R is the set relation associated with r. F defines a graph homomorphism between (Domain(F), Lower-Than) and $(\bigcup_{n \in N} \{F(n)\} R)$.

This definition of regularity may be easily represented by a *ratio* of "regular" links by total number of links [Mili&Rada 90b]. This gives the user a practical measure of regularity that is easy to manipulate.

The study of scientific book outlines, particularly in the field of medicine [Mili&Rada 90a], [Mili&Rada 90b] confirmed the hypothesis that regularity is an important mechanism that can be used to uncover the underlying domain models used by the writer. Such models are reflected in the structure of their books. Inherent to the outline of a book, the studies showed that there are two kinds of relationships: i) precedence relationships as reflected by the sequence of subsections and ii) hierarchical relationships of textual containment between a section of text and its subsections. These book outlines studies show that these relationships exhibit very strong regularity with respect to classifications of concepts of the model of the world.

Moreover, the regularity pattern may be used to make inferences to maintain the semantic net used to navigate through hypertext. Two kinds of inference have been detected : classification and expansion. Primarily, these methods complete the knowledge embodied in the semantic net in a way that is consistent with the established regularity patterns. However, as a by-product, these methods may be used to critique and suggest corrections for the existing semantic net. In classification, a node whose property values are known is hierarchically connected to nodes from the same conceptual domain. Expansion infers the property values of a node based on those of its neighbors in a given conceptual hierarchy. The classification algorithm was used to reclassify nodes from hierarchical semantic nets [Mili & Rada 90]. Most of the failures of classification could be explained and corrected. In all cases, the authors were able to take meaningful and semantically justifiable corrective actions that improved the performance of classification to a success rate near 90%. The remaining failures were due to the lack of explicit relationships in the hierarchies. In turn, the expansion procedure was applied to hierarchical classifications to infer missing properties, and the inferred values were compared to the actual values. Similarly, the authors obtained a high percentage of correct values.

Mili, H. and Pachet, F. "Regularity, document generation and Cyc", in *ExpertMedia - Expert Systems and Hypermedia*, Edited by R. Rada and K. Tochtermann, World Scientific Publishing, pp. 171-206, 1995

4. Regularity in Cyc

Given the gigantic size of the Cyc system, the question of having adequate tools to understand and navigate in the system is more important than ever. The concept of regularity is apparently well adapted to help handle the complexity because the system is based on a large number of independent hierarchies of concepts. However, the problem of detecting or exploiting regularities in Cyc is slightly different from the case of hierarchical semantic nets for two reasons:

- The notion of regularity as defined here is too strict to be applied directly to knowledge bases such as Cyc, because the relations that may hold between two independent hierarchies of concepts are usually not represented as a single slot but rather as a "path" or a composition of slots,

- Many regularities are already encoded *as such* in the system. In fact one of the main goals of the Cyc team is indeed to find regularities in the world that can simplify its representation. Regularities are therefore not interesting to detect when the rules that "make this regularity happen" are already available. On the contrary, regularity is to be seen as a high-level abstraction of local characteristics of the knowledge base.

In this section we will firstly give a definition of extended generality that is usable in the context of Cyc, then show that certain specialized inference mechanisms intrinsically exhibit regularity, whereas the most general inference pattern, the If-Then rule, tend to break regularity.

4.1. Extended regularity

In this section we extend the notion of regularity to describe cases of regularities between several hierarchies of concepts that are not simply embodied by a single slot.

As an example, let us consider the concept of an Automobile (the collection of all automobiles) and its various specializations : FrenchCar (itself specialized in PeugeotCar, Peugeot304Car), SouthKoreanCar (specialized in HyundaiCar and so on). Cyc proposes the slot "countryOfManufacturer", associated with any instanceOf Automobile (actually of any ManufacturedProduct), and which points to the frame representing the "home country of the manufacturer of the product".
For instance, all instances of FrenchCar have their slot "countryOfManufacturer" filled with "France" (the frame representing the country of France). Similarly (or rather, "regularly") instances of SouthKoreanCar have countryOfManufacturer pointing to SouthKorea, and so forth.
We can therefore detect a simple regularity between the two hierarchies (Automobiles and its specializations) and Country (and its various geographical sub regions). However, this regularity is not directly a case of regularity, because of the instanceOf relationship that exists between a single car and its collection. In other words, the slot countryOfManufacturer describes instancesOf Automobile but not the collection itself. Therefore no *direct* link exists between the various concepts of the two hierarchies. The "regular" link that we want to uncover is a composition of "instanceOf" and "countryOfManufacturer" (see Fig. 2).

In order to generalize this notion of regularity, we simply have to substitute "binary relations" by a more general notion, that we will call *access path*. An access path is simply a composition of slots that provides a link between a concept of source hierarchy to a concept in the target hierarchy. We will also extend this notion to include in its definition the specifications of both hierarchies.

Definition. *Extended regularity* is defined as a 6-ary predicate having the following parameters :

- a hierarchy A (e.g. Automobile.allspecializations),
- a relation that makes sense for concepts of A (for instance subBrandOf or generalization),
- a hierarchy B (e.g. Country.allInstances),
- a relation that makes sense for concepts of B (e.g. geographicalSubRegionOf),

- an "access-path" that provides a link between concepts of the two hierarchies (e.g. allInstances°countryOfManufacturer), and
- a ratio, which is a numerical measure of the actual regularity, computed as the ratio of "correct" couples by the total number of couples in the knowledge base, according to the same formal definition than (Reg).

This definition subsumes the definition given by (Reg), as standard cases of regularity correspond to access-path reduced to single slots. Regularity is computed from a given "state" of the knowledge base, and not directly from existing axioms or inference rules. In the case of rules having exceptions, the regularity calculus will depend on the number of actually instantiated classes. If no exceptional instance exists at the time of computation, the relation will be assumed perfectly regular. Regularity does not subsume the rules that serve to *establish* the regularity: it is a high-level view of a particular property of the combinations of several relations.

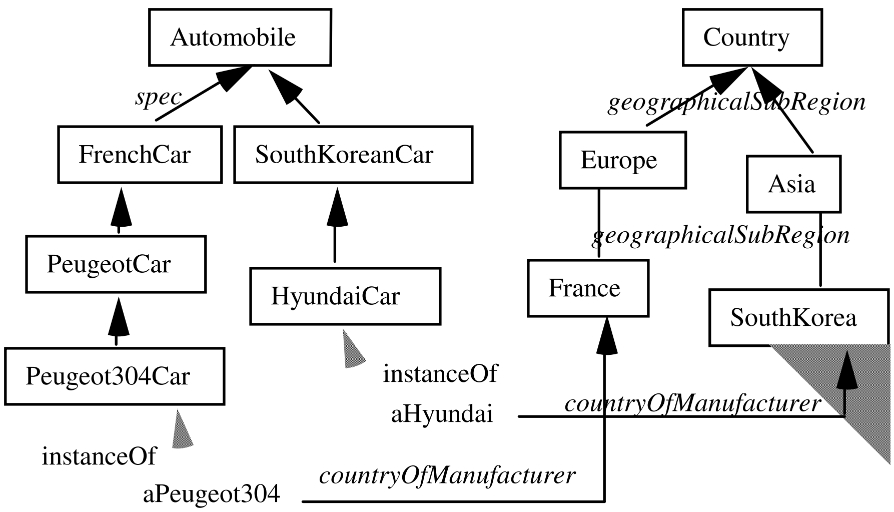

Figure 2: Extended regularity between car types and countries.

An other example of regularity that exemplifies the idea of indirect relationship can be found by substituting the access-path "allInstanceOf°countryOfManufacturer" by the more complex "allInstancesOf°madeBy°countryOfMainActivity", where:

"madeBy" is a slot that relates a product to a ManufacturingOrganization (here an AutomobileManufacturer), and
"countryOfMainActivity" is a slot that relates a manufacturer to its country of activity.

This regularity is the composition of two simple regularities : the preceding one, and the regularity that maps manufacturers to their country of activity :

Hierarchy A: Manufacturer.allInstances, relation A : branchOf
Hierarchy B: Country.allInstances, relation B : geographicalSubRegionOf
access-path: mainCountryOfActivity

Indeed, FrenchCars are "made by" (instancesOf) FrenchManufacturers which "operate" in France, SouthKoreanCars by southKorean manufacturers operating in SouthKorea and so forth. These examples show that extended regularity can account for a large number of regularities holding between two independent hierarchies.

4.2. Regularities of specialized inference patterns

Mili, H. and Pachet, F. "Regularity, document generation and Cyc", in *ExpertMedia - Expert Systems and Hypermedia*, Edited by R. Rada and K. Tochtermann, World Scientific Publishing, pp. 171-206, 1995

Almost all Cyc specialized inheritance patterns have a corresponding regularity definition. We give here two examples of very frequent inference patterns and their representation as regularity patterns : inheritance and transferThrough inferences.

Inheritance correspond to reduced forms of extended regularity, because the inherited property is usually fixed. Consider the rule "all birds fly", which is expressed as an inheritance pattern between all instancesOf Bird, that add to their slot "performsProcessType" the frame "Flying-Locomotion". This is a reduced form of regularity since all instances of Bird have exactly the same value for their slot performsProcessType (and not a kindOf "Flying-Locomotion"). However, this inheritance scheme can be described with the following regularity pattern :

Hierarchy A : Bird.allSpecializations, relation A : generalizations
Hierarchy B : Flying-Locomotion.allSpecializations, relation B : generalizations
access-Path : allInstances°performsProcessType
ratio : 93 % (there are exceptional non-flying birds, such as chickens in the knowledge base)

Actually this representation is a little bit more general than the "all birds fly" rule described because it allows specializations of "Flying-Locomotion" to be associated with instances of Bird : one could imagine different types of Flying-Locomotion that could be "regular" with allInstances°performsProcessType (e.g. Eagles could performsProcessType EagleLikeFlyingLocomotion).

Another more sophisticated and powerful inheritance pattern is the so-called *transferThrough* pattern. This pattern represents inferences of the form :
Let (xy, y, z) be frames and (s1 and s2 slots), s1 **transferThrough** s2 iff ((x s1 y) and (x s2 z) => (z s1 y)).

For instance, inferences such as "owning a thing implies owning all its parts" is represented in Cyc by the statement (transferThrough owns partOf) (see Fig. 3). This is a singular form of regularity where the relation in the first hierarchy is empty, and which applies only to one particular concept (here an instanceOf Person).

Hierarchy A : Person.allInstances
 relation A : empty
Hierarchy B : Product.allInstances
 relation B : partOf
access-path : owns

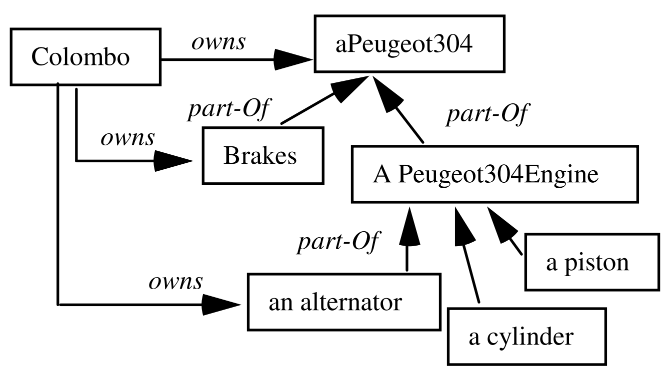

Figure 3 : TransferThrough relations as regularities.

4.3. Regularity of temporal sub abstraction relations

The notion of temporal sub abstraction in Cyc is used to represent various pieces of time of the same concept. For instance, LieutenantColombo may have several sub abstractions for each interesting episode of its life (ColomboDuring1stEpisode, ColomboDuring23thEpisode and so on). Each sub abstraction can in turn have several sub abstractions, so sub abstractions form a natural hierarchy. This relation exhibits natural regularity when coupled with other hierarchies. For example, the statement that Colombo owns a Peugeot304Car has to be seriously refined to take time into account : at each moment of an episode, Colombo does not actually owns his Peugeot304, but the corresponding temporal sub abstraction of the car for the time the sub abstraction is referring :

Mili, H. and Pachet, F. "Regularity, document generation and Cyc", in *ExpertMedia - Expert Systems and Hypermedia*, Edited by R. Rada and K. Tochtermann, World Scientific Publishing, pp. 171-206, 1995

ColomboAfterTheCrime owns ColombosCarAfterTheCrime, and so on (see figure 4). Similarly there is a regularity between sub abstractions and 'parent' relationships, as well as a (surprisingly big) number of other relations.

Hierarchy A : Person.allInstances, relation A : subAbstraction
Hierarchy B : TangibleThing.allInstances, relation B : subAbstraction
access-Path : owns.

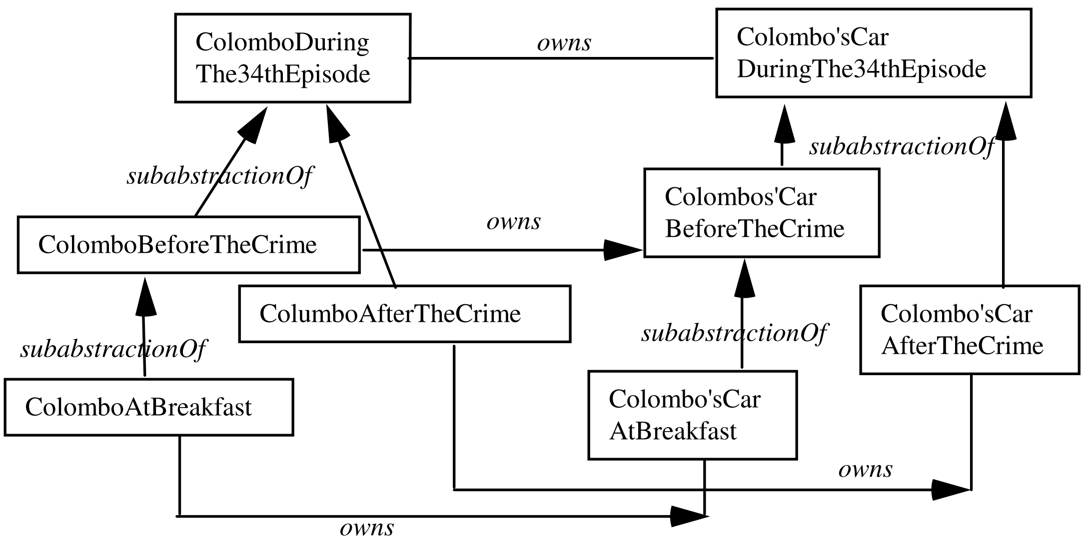

Figure 4 : temporal sub abstraction as extended regularity.

Although this correspondence between the hierarchy of temporal sub abstractions of an entity and the sub abstractions of any entity owned is a good example of extended regularity, the computed ratio for this regularity may not be as high as expected. The reason is that human beings are often involved in buying transactions whose effect is a "transfer of property". This is precisely what the if-Then rules are all about : describing complex relationships between concepts. We will show now that if-Then rules may be seen, unexpectedly, as expressions of, or justifications for *irregularities* in the knowledge base.

4.4. General If-Then rules express irregularities

The preceding section has shown that specialized inference mechanisms have corresponding descriptions in terms of extended regularity. This description of the regularity is not equivalent to the inference mechanism itself (it does not allow to make inferences as precisely for instance) but gives a high level view of the state of the knowledge base with regards to the relation between two independent hierarchies.
The most powerful inference mechanism in Cyc is called the if-Then rule. It allows to express if-then rules that do not exhibit particular syntactic properties. This is the case for instance for rules involving more than two objects. For instance, there are rules that describe what happens when a buying transaction occurs between two agents, such as : transfer of property (one of the agent owns the product before, and the other one after), transfer of legal rights, various constraints on the availability of the product being transferred (e.g. the seller must own the product before), the money transfer that parallels the transfer itself, and various constraints such as " buyer, seller and product should be cotemporal", and so on.

Here is a rule (in a simplified syntax) that states that if anAgent performs aBuyingTransaction, then he will own the transaction object after the transaction :

OwnWhatYouBuy :

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

Of course this rule states a kind of regularity (hence the word 'rule' which share a common etymological root with 'regular') : this transfer of ownership happens each time there is a buying transaction. But the important point to note here is that rules like this one can also be seen, paradoxically, as *expressions of - or justifications for - irregularities* in the knowledge base, with respect to the extended definition we give in this paper. Indeed, this rule explains, among other things, why the relation 'own' is not regular with regards to temporal sub abstractions : the ideal situation sketched in Figure 4, where every the sub abstractions of Colombo owns a corresponding sub abstraction of the car does not hold any longer if, say, Colombo sells his car, or before Colombo had bought his car (situations which we carefully avoided in Figure 4 by specifying ColomboDuring34thEpisode as a root of hierarchy A, knowing that no episode ever showed him sell his car).

In summary, inference mechanisms in Cyc are strongly related to regularity, both as regularity "generators" (specialized inference mechanisms) and regularity "breakers" (general If Then rules). We will now apply this notion to expertext construction.

5. Exploiting regularity and irregularity in Cyc

5.1. "The more regular the less interesting" hypothesis

By studying examples of regularities and irregularities in Cyc, we came to the intuitive conclusion that the more regular a relation is, be it a simple slot or a combination of slots, the less interesting it is with regard to textualization. More precisely, given two hierarchies and a relation between them, three cases may occur:

- The relation is very regular. This is the case for instance with the relation between various brands of cars and their respective countries of manufacturer or between genealogies of people and their age hierarchies. Although very regular relations in hierarchical semantic nets may be classified as either 'incidental' or 'essential' [Mili&Rada 90a], the transposition of these notions in knowledge bases such as Cyc does not seem to make much sense because no relation may emphatically be assigned such property a priori. In this case, the hypothesis is that there is probably not much to say about it, so one should avoid associating text to these relations or relying on them in expertext traversal.

- The relation is very irregular. This is the case if the relation is purely arbitrary. For instance, the relation between the brand of car of a person, and the kind of cancer he may have. This case is similar to the preceding one in terms of expertext relevance.

- Intermediary cases where the relation is not regular, but this irregularity is caused by the structure of the knowledge base itself are the most interesting as far as textualization is concerned. This is the case with owns/temporal sub abstraction, which is not regular because of the existence of rules that define changes during transactions for instance. Those changes are extremely important with regards to the own/temporal sub abstraction relation, and so would be i) very interesting to textualize or comment and ii) should be used as a central spot in the knowledge base/expertext for navigation. For example, the relation caused-By may be shown to be quite regular with regard to, say, classifications of diseases and classifications of causes. [Mili&Rada 90a] showed that portions of the MeSH thesaurus could exhibit regularity between these two taxonomies : for instance, conjunctivitis is causedBy Infection, and the various kinds of conjunctivitis are caused by specializations of Infections (Viral conjunctivitis is causedBy viral infection and so on). However, this regularity is not perfectly convincing, as we can see for instance in the cancer taxonomy : LungCancer may be causedBy ExcessiveSmoking, whereas SkinCancer may be causedBy ExcessiveExpositionToSun, but the relation between those two types of cancer may have nothing to do with the relation between the causes. This is explained either by:

- Poor taxonomies or missing intermediary nodes. This can be corrected by suggesting the knowledge base builder modifications,
- Explicitly represented irregularities (such as general If-Then rules). In this case those irregularities are to be given strong emphasis in the text generation strategies.

And indeed these causality relations are worth textualizing : explaining why SkinCancer is causedBy ExcessiveExpositionToSun is intuitively more interesting than explaining why FrenchCars are madeIn France. Rather than relying on semantic a priori on the nature of causation, this interestingness may here be related to purely formal properties of the underlying classifications, and detected by formal procedures.

6. Discussion

The main idea of this argumentation is to use the regularity calculus to find irregularities, and then analyze these irregularities, rather than exploring the knowledge base ad libitum. In this respect regularity is a valuable tool for navigation in big knowledge bases.

As a side-effect, regularity provides a way of looking at a slot semantics for discrimination purposes. We claim that slot semantics is defined by the tree of possible inferences that relies on the slot values. For instance, the only difference between the slot owns, possesses, or isLegalOwnerOf are to be found in the respective trees of possible inferences, e.g., own is the only slot that will be used to trigger the rules that talk about buyingTransactions. Since those trees may be potentially very big, regularity can help by providing a high level view of the behavior of a slot in a given context. Slot comparisons can then be based on the comparisons of their respective regularities. Here, the slot own is irregular with regard to temporal sub abstraction, because of the existence of a set of rules that describe these changes. Possesses is less interesting with regards to temporal sub abstractions because no rule says any thing about the violation of regularity of possesses/temporallySubsumes. And indeed, the slot possesses (at least in the present state of the system) is more or less a "ghost" slot which does not seem to be used any more. For representing situations of money transfers, the slot owns is more interesting because of its irregularity properties.

By implementing regularity calculus in the Cyc knowledge base, we provide a operational definition of the interestingness of arbitrary relations with respect to text units associations. Based on a seemingly paradoxical result ("rules express irregularities"), the regularity approach to expertext building and manipulation provides a conceptual tool to use the system as well as a measure of the system's intimate topology that helps its maintenance and extension.

7. References

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

[Lenat&Guha 90] D. Lenat, R.V. Guha. Building large knowledge-Based Systems. Representation and Inference in the Cyc project. Addison Wesley, 1990.

[Lester & Rada 87] S. Lester and R. Rada. Method of Building Medical Knowledge Bases. Methods of Information in Med. 26 (1) (1987), 31-39.

Mili, H. and Pachet, F. "Regularity, document generation and Cyc", in *ExpertMedia - Expert Systems and Hypermedia*, Edited by R. Rada and K. Tochtermann, World Scientific Publishing, pp. 171-206, 1995

[Mili 88] H. Mili. Building and Maintaining Hierarchical Semantic Nets. August Doctoral Dissertation Washington, D.C. Department of Computer Science, George Washington University.

[Mili 91] H. Mili. Hypertext: Managing software documentation within an Object-Oriented toll for Software reuse. Proc. of the World Congress on Expert Systems, Orlanda, florida, Dec. 16-19, pp. 2785-2794, 1991.

[Mili&Rada 90a] H. Mili and Roy Rada. Regularity: Structuring Paradigm for Hierarchical Knowledge. Proc. Hawaii International Conference on Systems Sciences, Vol III (January 1990) 413-421.

[Mili&Rada 90b] H. Mili, R. Rada. Medical expertext as regularity in semantic nets. Artificial Intelligence in Medicine 2 (1990) pp. 217-229.

[Mili&Rada 90c] H. Mili and R. Rada. Fuzzy Regularity. IEEE Transactions on Systems, Man, and Cybernetics (1990).

[Rada 89] R. Rada. Writing and Reading Hypertext: An Overview. Journal American Society Information Science, 40 (3) (1989) 164-171.

[Rada & Barlow 89] R. Rada and J. Barlow. Expert Systems and Hypertext. Knowledge Engineering Review, 3 (4) (1989) 285-301.

[Streitz 89], N. A. Streitz, J. Hannemann, M. Thüring. From Ideas and Arguments to Hyperdocuments : Travelling through Activity Spaces. Proceedings of the ACM conference HYPERTEXT '89, pp. 343-364, Pittsburgh, PA. USA, Nov. 5-8, 1989.

[Wand & al 91] W. Wang, R. Rada, C. Ghaoui. An Expertext Aid for Writing. Proceedings on the World Congress on Expert Systems, Orlanda, florida, Dec. 16-19, pp. 2767-2775, 1991.