TR-I-0014

Genaralizing Unification in Semantic Networks toward Natural Language Understanding

自然言語理解のための意味ネットワーク上のユニフィケーションの一般化

Nadine Lerat and Teruaki Aizawa
ナディヌ・ルラ　相沢 輝昭

November, 1987

Abstract

This paper deals with a generalization of the concept of unification using a semantic network.

Unification is a pattern matching operation of crucial importance in Artificial Intelligence. First, a less restrictive semantic network was introduced using a new relation, called *similarity relation*, on the set of nodes of the semantic network. Next, the usual unification based on "exact" matching was generalized as a *similarity unification* that can treat a kind of "similarity" matching.

As a possible application of this formalism, a more accurate word-to-word correspondence in different languages was investigated by developing a Prolog program for words of both English and French.

This work was done as an internship of the first author at ATR Interpreting Telephony Research Laboratories.

ATR Interpreting Telephony Research Laboratories
ATR 自動翻訳電話研究所

This is an excerpt from the Internship Report

Period:	from April 1, 1987 to November 30, 1987
Company:	ATR Interpreting Telephony Research Laboratories Osaka, Japan
Topic:	Generalizing Unification in Semantic Networks toward Natural Language Understanding
Name:	Nadine Lerat, Student of the ENST (Ecole Nationale Supériere des Télécommunications), Paris, France

INTERNSHIP REPORT

Nadine LERAT

ATR Interpreting Telephone Research Laboratory

ACKNOWLEDGEMENTS

I would like to thank Dr Akira Kurematsu, President of ATR Interpreting Telephony Research Laboratories, for greeting me to stay in his laboratory and for supporting my research.

I am also very grateful to Mr Teruaki Aizawa, Head of the Natural Language Understanding Department, who directed my internship and helped me in my research.

I also wish to thank Dr Hisao Kuwabara for taking care of all the administrative formalities before and during our stay in Japan, and for the very interesting tour of research laboratories that he arranged for us in Tokyo.

Eventually, I would like to thank all the employees of ATR who greeted me in a very friendly way and who made me feel comfortable and at ease. Through this exchange, I learned a lot about Japan and its people and this internship helped me to rule out some common clichés about Japan which are widely spread in European countries.

CONTENTS

INTRODUCTION

I. PRESENTATION OF THE SEMANTIC NETWORK
  I.1 The semantic categories
  I.2 The Semantic Graph
  I.3 The Semantic Network

II. GENERALIZATION OF UNIFICATION USING A SEMANTIC NETWORK
  II.1 Definition of the Similarity Unification
  II.2 Unification of Predicates and Clauses
    II.2.1 S-Unification of Predicates
    II.2.2 S-Unification of Clauses

III. AN APPLICATION OF THE GENERALIZED UNIFICATION TO A BETTER CORRESPONDENCE AMONG WORDS IN DIFFERENT LANGUAGES
  III.1 English-French and French-English word-to-word correspondence
  III.2 English-Japanese and Japanese-English word-to-word correspondence
  III.3 Conclusion

CONCLUSION AND FURTHER RESEARCH PROSPECTS

BIBLIOGRAPHIE

APPENDIX

INTERNSHIP SUMMARY

During my internship, I have been working in ATR Interpreting Telephone Research Laboratory located in Osaka (Japan). I was involved in the Automatic Machine Translation project as part of the Natural Language Understanding department. My research theme was to investigate the field of *Semantic Networks* and to examine to what extent a semantic network could be useful during the translation process.

Roughly speaking, a semantic network models how semantic information is organized within a person's memory. For instance, the grammatical role of each word of the sentence "John loves Mary" – agent, verb, subject – can easily be represented through a semantic network. However, since that kind of representation sometimes lacks generality, I focused on a more powerful representation proposed by Mc Skimin & Minker in [Skim,Mink79] which could handle general statements such as "if an object is a human being, then it is either a man or a woman". Moreover, with that representation we can consider different views of the same set of objects : for instance, the set of human beings can be considered as the union of men and women, children and adults ...etc... depending on the features we are interested in.

Having selected that formalism for a semantic network, I had to define a new unification process – called *similarity unification* – for the classical unification algorithm could no longer be used ("Unification" is a pattern matching operation of crucial importance in Artificial Intelligence and is used during the process of query answering). Later on, I investigated a possible application of this formalism to provide for a more accurate word-to-word correspondence in different languages (an example concerning English and French languages has been implemented). In that prospect, the semantic network considered in both languages was related to the vocabulary used by a researcher willing to attend an international conference (involving such topics as accomodation, transportation, schedule ...). Among all English words representing the same general French concept, a unique representative is selected (and conversely for French words) ; hence, the English word *tour* corresponds to two different French concepts *voyage d'affaires* and *excursion.* Since in Natural Language Understanding semantic networks are mainly used for reducing ambiguities, I elaborated a prolog program which – thanks to the similarity unification defined previously – may discard some of the possible translations if some extra semantic knowledge is available. A next step of research would be to integrate that disambiguation device within a set of inference rules inducing some kind of semantic knowledge.

I presented the results of this research during a conference in Hokkaido and a copy of the paper can be found in the appendix.

INTERNSHIP PROJECT :

CONTEXT, GOALS & ACHIEVEMENTS

During my internship period, I belonged to ATR Interpreting Telephony Research Laboratory where fundamental researches in Speech Processing and Natural Language Understanding are conducted in parallel. These researches take place within a wide project aiming at automatic translation and transmission of what is said by two persons from different countries connected via telephone lines. Therefore, both speech recognition, automatic translation and speech synthesis issues have to be tackled.

Within the Natural Language Understanding department to which I belonged, people from different backgrounds – linguists as well as grammar specialists and computer scientists – are investigating various research problems towards automatic machine translation :

1/ elaboration of a model for the discourse (2 persons)

2/ automatic understanding of discourse (3 persons)

3/ elaboration of an automatic translation process for written correspondence (4 persons)

4/ interface between Natural Language Understanding and Speech Processing (2 persons)

Meanwhile, a database which will later be used for testing the automatic translation processes is developped. On the one hand, two researchers are involved in the creation of a dictionnary for that database. For future tests, the vocabulary envisionned here – extracted from samples of telephone or keyboard conversations – deals with all the problems a researcher may have when he goes to an international conference (application, accommodation ...etc ...). On the other hand, two researchers are elaborating a knowledge base.

Lately, that is to say since the beginning of October 1987, people are beginning to investigate possible applications of the connectionnist model and of neural networks in the field of natural language understanding.

As far as I was concerned, the purpose of my internship was to investigate another field of fundamental research that turns out to be very helpful to build a machine translation system : *Semantic Networks* .

Indeed, semantic networks provide for a hierarchical classification of words which, in a sense, models human memory. Hence, semantic networks should be particularly relevant for being integrated into the conception of a more general network (*neural network* for instance) aiming at modeling human brain and the way it functions. In fact, the main idea is that the semantic features described through the semantic network should enable us to disambiguate words during the automatic translation process. For instance, the French word "vol" should clearly be translated in two different ways whether it occurs within a transportation context (the English equivalent being then "flight") or in a police related context (it would then be translated in English as "robbery"). Hence, we have to make the most of the semantic context if we aim at providing for an accurate translation.

Therefore, at the beginning of my internship I gathered information from a lot of papers in the semantic network field. Indeed, semantic networks are variously thought of : diagrams on paper, abstract sets of n-tuples of some sort, data structures in computers and even information structures in brains. Yet, generally speaking, a semantic network is a directed graph whose nodes represent individuals and whose arcs represent relationships between individuals. In such a graph, and arc is labeled by the name of the relation it represents. For instance, in the semantic representation of the sentence :

"John loves Mary"

the arc pointing to "John" will be labeled "agent" whereas the arc pointing to "loves" will be labeled "verb".

However, the specification of semantic networks quoted above – often termed *simple semantic network* – is restricted and cannot handle general statements such as "if an object is a human being, then it is either a man or a woman". Moreover, some of these formalisms have been shown to be highly inadequate or difficult to manage ; therefore, I had to discard them and to choose a more elaborate semantic network that would be more suitable for tackling disambiguation problems.

Hence, in a second step, I focused on the formalism of a semantic network presented by Mc Skimin and Minker in [Skim,Mink79]. Given that representation, the semantic network has the full power of the predicate calculus and is thus able to answer queries involving semantic information in a rather easy and straightforward way.

During the query retrieval process, the semantic network is used first of all to narrow the search space by selecting only the relevant information and also helps rejecting meaningless queries such as "Find all the individuals who are *simultaneously* the *father* of an individual *a* and the *mother* of an individual *b*". Moreover, it controls data input by the user and rejects it whenever it conflicts with the current semantic knowledge.

One of the main features of that formalism is that it provides for several views of the same semantic category. For instance, the semantic category *animate* could be viewed as the union of the categories *human-being* and *non-human* or as the union of *male* and *female* , depending on what features we are interested in.

Yet, this modification which seems quite intuitive entails some important changes. For instance, the usual unification algorithm – an operation matching certain subexpressions extremely important and widely used in Artificial Intelligence – has to be redefined.

Therefore, a third step of my research consisted in finding a new unification algorithm – called *similarity unification* – for the type of network considered above. Two different views of the same semantic category being connected by a "similarity" link in the semantic network, this new algorithm is no longer based on "exact" matching but on a more general "similar" matching. Eventually, I implemented a prolog program achieving similarity unification on a small example.

Afterwards, considering samples of telephone and keyboard conversations dealing with various problems a researcher wishing to attend an international conference may encounter, I could list the main occurring words and elaborate an appropriate semantic network in that specific context.

However, this network being too cumbersome to manipulate, the scope of further applications was reduced to transportation matters.

Indeed, the final step of my internship enlightens the interest of the "similarity unification" for word to word correspondence in different languages.

Familiar examples of word correspondence show that a word of the source language may have different possible translations in the target language, depending on the meaning it conveys. For instance, the English verb "know" may be conveyed either by "savoir" or "connaître" in French.

Thus, having defined two semantic network for English and French related to the conference domain, English (*resp.* French) words corresponding to an identical French (*resp.* English) general concept are grouped together and one of these English (*resp.* French) words is chosen to represent that French (*resp.* English) concept.

Hence, the word *tour* as a synonym of "business-trip" would be translated in French by *voyage* whereas, as a synonym of "leisure-trip", it would be translated by *excursion.*

In that prospect, the similarity unification algorithm allows us to take semantic knowledge into account and may lead to discarding some possible translations, thus completely or partially disambiguating the translation process.

Eventually, for the transportation matters domain, I implemented a prolog program providing for better English-French correspondence. For instance, given the English word *tour*, the program in a first step provides for the set of possible French equivalents *voyage* and *excursion*. Then in a second step, if some extra semantic knowledge is available – for instance, if we are in a leisure trip context – , the French word "voyage" will be discarded and eventually we get the unique French translation *excursion* for the English word *tour* in a leisure trip context.

The results of this research have been presented during a conference held in Hokkaido on 28-30 September 1987. A sample of the article presented is included in the appendix of the technical report.

INTRODUCTION

Since Quillian first introduced the idea of a "semantic network" in [Quil68], this topic has been the subject of extensive research. Semantic networks were introduced in the literature as a means of modeling how semantic information is organized within a person's memory. Initially and now onwards, they have been used for disambiguating and understanding natural language.

Broadly speaking, a semantic network is a directed graph whose nodes represent individuals and whose arcs represent relationships between individuals. In such a graph, an arc is labeled by the name of the relation it represents. For instance, in the semantic representation of the sentence :

"John loves Mary"

the arc pointing to "John" will be labeled "agent" whereas the arc pointing to "loves" will be labeled "verb".

Indeed, semantic networks are variously thought of : diagrams on paper, abstract sets of n-tuples of some sort, data structures in computers and even information structures in brains.

However, the specification of semantic networks quoted above – often termed *simple semantic network* – is restricted and cannot handle general statements such as "if an object is a mammal, then it is either a male or a female". Moreover, Woods in [Woo75] has analysed semantic networks as they relate to the representation of natural language meanings and has also denoted some of the inadequacies of simple semantic networks.

Therefore, it seems necessary to look for extensions of simple semantic networks based on predicate logic which would allow to represent quantified information. Quite a bunch of various extensions have been proposed ; however, discarding those who avoid extensional quantifiers by introducing functions, we will focus on the definition of an *extended semantic network* proposed by Mc Skimin and Minker in [Skim,Mink79].

Indeed, their representation of a semantic network turns out to be very useful during the deductive phase of a problem solving system. For instance, the primary use of that semantic network is to narrow the search space during a deductive search.

In the following section, we will explain in details the definition of semantic network chosen and we will point out its main advantages.

I. PRESENTATION OF THE SEMANTIC NETWORK

In [Skim,Mink79], Mc Skimin and Minker define an extended semantic network which seems particularly appropriate for applications in Natural Language Understanding. Therefore, in this chapter, we will present this network and underline its main advantages.

The semantic network we describe will be used during the deductive phase of an inferential system. Its original framework – the first-order predicate calculus – has been slightly modified in order to accomodate semantic information concerning the domain of application. Therefore, unlike previous deductive searching using the predicate calculus and relying only on syntactic pattern matching, here both syntactic pattern matching and semantic information about the domain can be used during the retrieval process.

In the semantic network, four types of information are stored :

1/ the *data base* contains facts and general inference axioms which are represented by predicate calculus clauses.

2/ the *semantic form space* provides for semantic constraints on arguments of predicates or functions.

3/ the *dictionary* defines for each predicate, function and element its semantic category.

4/ the *semantic graph* defines the set-theoretic relationships between semantic categories.

Each argument of a predicate is constrained to belong to a named sort which is a member of the semantic network. This semantic network can then be used in three fundamental ways :

(1) for rejecting irrelevant queries, assertions or inference rules whenever the arguments of the predicates are not consistent with the semantic constraints specified in the semantic network.

(2) for selecting only semantically relevant assertions and rules when answering some part of a query, thus reducing the search space.

(3) for detecting when all the semantically possible answers for a query have been found, thus avoiding further deductive search.

I.1 The semantic categories

We would like to avoid two problems arising in deductive systems : on the one hand, using irrelevant data or general rules during the deductive process, on the other hand attempting to solve semantically meaningless problems.

For instance, in order to illustrate the first kind of problem, if we want to find x such that ATTEND(Minker, x) – i.e. Minker is part of the attendance of the conference x – the axiom ¬LISTENER(u,v) ∨ ATTEND(u,v) may be applied during the resolution of the query. However, in an environment where Minker is known to be a *speaker*, since the first argument of the predicate *LISTENER* is semantically constrained to be a *listener* (whose set is disjoint from the set of speakers of the conference), then – as soon as semantic constraints are taken into account – the axiom quoted above will clearly be irrelevant for answering the original query.

To understand the second kind of problem, just consider the query "Is there a person who is the father of the individual *a* and the mother of the individual *b* ? ". Obviously enough, in a domain where the first argument of FATHER(x,y) must be *male*, where the first argument of MOTHER(x,y) must be *female* and where the two sets male and female are known to be disjoint, this query is semantically meaningless and there is no need to start a deductive process for answering it.

Thus, the semantic network should achieve two purposes : first of all, filter semantically irrelevant axioms and rules, then reject ill-formed problems. Therefore, we should provide for a precise definition of the domain of discourse D – i.e. give explicitly its content and the relationships between the different subsets. These subsets are called semantic categories and constitute a particular structuring of D. This hierarchy has to be specified according to the specificities of the domain of interest, in the case of topics related to the conference domain, a hierarchy is proposed in Annex 1.

Set-membership relations are represented as semantic categories instead of unary predicates and interaction relations between objects are denoted by n-ary predicates. That is to say that the fact that John is male will not be represented by MALE(John) where "MALE" would be a unary predicate but by the fact that the object "John" will belong to the semantic category *male* ; meanwhile FATHER(Terry,John) will denote that Terry is the father of John. Since set-interrelationships and interaction relations are of different flavor, they should be represented in different ways and treated separately.

Indeed, for solving problems encountered in inferential systems, it is particularly interesting to use a special semantic category representation for set-interrelationships *separate* from the representation used for interaction relations. Otherwise, whenever we want to specify set restrictions on arguments of an interactionrelation, a list of set relations has to be appended and checked whenever a substitution of the arguments is made.

For instance, if we want to find a person attending the conference of Artificial Intelligence IJCAI who is male, foreigner and speaker at that conference, the resolution process will start with the negation of the query, i.e. :

$$\neg\mathrm{ATTEND}(x,\mathrm{IJCAI}) \vee \neg\mathrm{MALE}(x) \vee \neg\mathrm{FOREIGNER}(x) \vee \neg\mathrm{SPEAKER}(x)$$

This expression is very cumbersome and may require separate peripheral storage for each set relation retrieved. Moreover, if inference rules have to be used to retrieve ATTEND(x,IJCAI) some of them may be inconsistent with the semantic restrictions that follows and in that case work will be done in vain. On the other hand, the semantic category representation has the advantage that semantic conflicts will be detected before applying an axiom or an inference rule, thus avoiding useless efforts. Furthermore, set-membership relations can be retrieved easily either directly from data base facts or through a straightforward set inference (such as transitive superset relations) mechanism. Eventually, the unary predicate representation has no efficient way of denoting exclusion from a given set.

For avoiding the problems stated above, the semantic category graph is proposed as an efficient way of representing set-membership relations.

I.2 The Semantic Graph

The representation developped for set-interrelationships is using the concept of semantic category whose advantages have been underlined in the previous section. A semantic category is a *name* regrouping a collection of *known* elements of the domain. For instance, *foreigner* is the semantic category consisting of all the foreigners listed in the database. In the conference related area in which we are more specifically interested, other examples of semantic categories are : male, female, attendance, speaker, transportation...etc...An example of a semantic graph for the conference domain is presented in Annex 1.

The semantic graph is a finite graph without cycles whose nodes are semantic categories. All these semantic categories are subsets of a universal category representing the universe of discourse – i.e. the domain. In this semantic graph, there are three types of set-theoretic relations (denoted by labeled arcs interconnecting the nodes of the graph) between semantic categories :

(1) if category c is a superset of category d – i.e. every element of the domain belonging to d also belongs to c – then the arc c $\longrightarrow$ d in the semantic graph is labeled by g and is called a *g-link*.

(2) if category c equals category d – i.e. c and d are two names for the same set of objects – then c and d are linked by a *s-link* (*s* standing for *similarity*) which is a non-oriented link.

(3) category c and d may also be disjoint – i.e. there is no object of the domain belonging simultaneously to c and d.

In practice, disjunction is not represented in the semantic graph since the latter is constrained as follows : If a semantic category c is linked by g-links to the categories $c_1,.., c_n$ (i.e. $c_1,.., c_n$ are subsets of c) then $c_1,.., c_n$ constitute a partition of the category c ; that is to say that the union of the c_i is the category c and for i$\neq$j c_i and c_j are disjoined semantic categories. For instance, in the example given in Annex 1, the universal category is partitionned in *animate* and *inanimate* and these categories are disjoined from one another.

Categories linked by s-link contain exactly the same objects of the domain. Hence, they are different names corresponding to different partitions of the same set of objects. In fact, considering the semantic graph described in Annex 1, the use of s-link allows to view the attendance of a conference in different ways : divided in speakers and listeners in the first view ; in company staff, university staff and independents in the second view and in foreigners and locals in the last view. This is indeed the most important feature of that semantic graph since it enables us to choose the partition which corresponds to what we are interested in.

Besides, more complex expressions built up from categories – that is to say combinations of semantic categories with the operators union ($\cup$), intersection ($\cap$) and complement (-) – will be used for representing data base facts or general rules.

I.3 The Semantic Network

In order to accelerate the retrieval process, all the semantic information available is regrouped in the semantic network. It makes explicit within the computer model the usually implicit knowledge about facts, thus leading to more efficient query answering.

The semantic network has four components :

1/ the data base of assertions and general rules.

2/ the semantic form space, defining the semantics of each argument of n-tuples.

3/ the dictionnary, containing information about predicates, functions, constants and semantic categories.

4/ the semantic graph specifying the set-theoretic relations between semantic categories.

I.3.1 Data Base Assertions and General Rules

Assertions are explicit facts about elements of the domain and general rules are clauses that may be used to derive assertions. To handle semantic categories, extended Π-clause notation is introduced. An extended Π-clause is a (n+1)-tuple where the first argument is a n-ary predicate and where the following arguments are bound to given sets of objects.

For instance, (α,x,y) { [FATHER]/α, [John]/x, [Mary,Ted]/y } is a more compact way of expressing the two first-order predicate calculus clauses FATHER(John,Mary) and FATHER(John,Ted).

In order to take advantage of semantic categories, the definition above is modified to allow restriction of arguments to Boolean category expressions (i.e. boolean combination of simple semantic categories) ; besides, quantification over subsets of the domain is allowed. Hence, in the most general case a variable x quantifies over a subset S = ($[D^E,(D^X)]$, D^I) of the universe where :

* D^E is the set of constants explicitly included in S

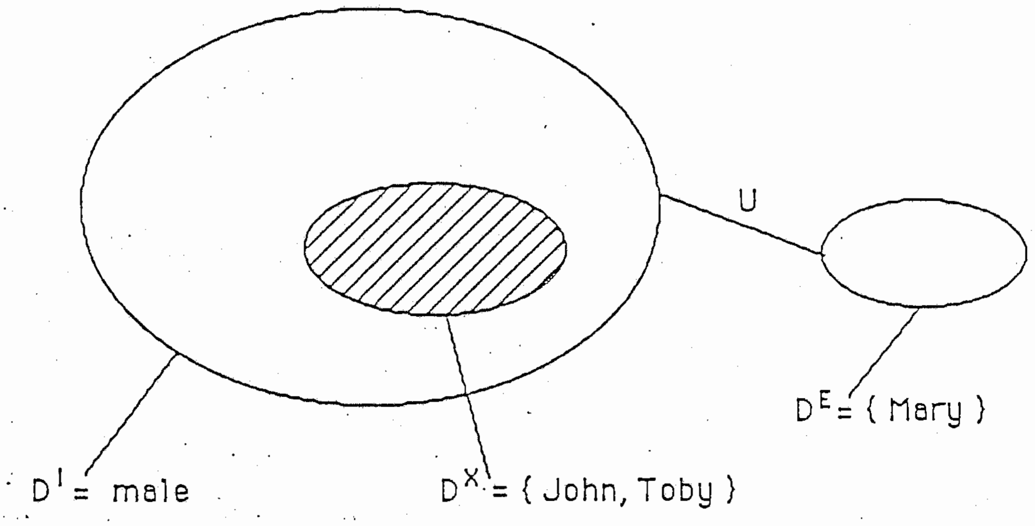
* D^X is the set of constants explicitly excluded from S

* D^I is a set of constants represented as a Boolean category expression implicitly included in S.

Moreover, in order to avoid redundant information that would necessitate cumbersome treatment, the following conditions should be satisfied :

$$D^E \cap D^I = \emptyset \quad \text{and} \quad D^X \subseteq D^I$$

For instance, if x belongs to S = ([{Mary},({John,Toby})], male) then x is either *male* but is not John nor Toby or x is Mary. The set relationships are shown in the following figure :



Indeed, the extended Π-clause representation combined with the use of such subsets of the universe constraining variables provides for a very compact and convenient way for representing information. In fact, the definition of subsets explained above is also used as semantic constraint in general rules and queries.

I.3.2 The Semantic Form Space

In the semantic form space, semantic category restrictions for arguments of predicates are stored. The semantic forms make the semantic of predicates explicit by stating to which subclass each of its arguments should belong.

Via semantic unification algorithms, it can be used to perform well-formedness tests on queries or data base clauses input by the system user. Hence, on the one hand, it guarantees that no assertion inconsistent with the present semantical knowledge of the data base could possibly be added. For instance, if the first argument of the predicate FATHER must belong to the semantic category *male*, then the assertion FATHER(Mary,John) – inconsistent with the semantic knowledge – will not be added to the data base. On the other hand, checking well-formedness of expressions can avoid processing meaningless queries and is therefore systematically performed before starting the query answering process.

I.3.3 The dictionary

The dictionary defines the semantic categories assigned to constants, function domains and ranges, and predicates. Besides, it defines the structure of the semantic graph.

That is to say that the way the universe is partitionned is recorded in the dictionary entries for semantic categories.

Indeed, a semantic category entry in the dictionary includes its synonyms (i.e. categories linked by *s-link* to this semantic category) and the various partitions of the set of objects named by that semantic category.

For instance, if we look at the semantic graph of Annex 1, the dictionary entry for the semantic category *human* would contain the two synonyms *human1* and *human2* and the two partitions they represent (*male, female*) and (*confmemb, outconf*).

The dictionary also contains entries for constants and predicates which state to which semantic category constants belong and enforce semantic constraints on arguments of predicates. The semantic category of every constant should be defined within the semantic network before attempting to answer a query. Moreover, in order to avoid problems of decidability of set membership, that semantic category should be a *primitive category*, that is to say a terminal node of the semantic graph (a category who has no subsets).

Similarly, function entries in the dictionary store the number of arguments of the function and the semantic category to which each argument should belong. Unlike predicates whose semantic well-formedness is checked by the unification algorithm, here a recursive procedure is used to verify the compatibility of the instantiations of the arguments with the semantic forms. Skolem constants (representing existential quantifiers) can be restricted to range over semantic categories.

I.3.4 The Semantic Graph

For better computing efficiency, Mc Skimin & Minker decided to store the relations between all categories of the semantic graph in a special matrix in the computer. Since this was a simple trick for speeding up the computing process, I discarded it in the theoretical study exposed in the following chapter. However, it will certainly turn out to be very useful as soon as practical applications will be investigated.

II. GENERALIZATION OF UNIFICATION USING A SEMANTIC NETWORK

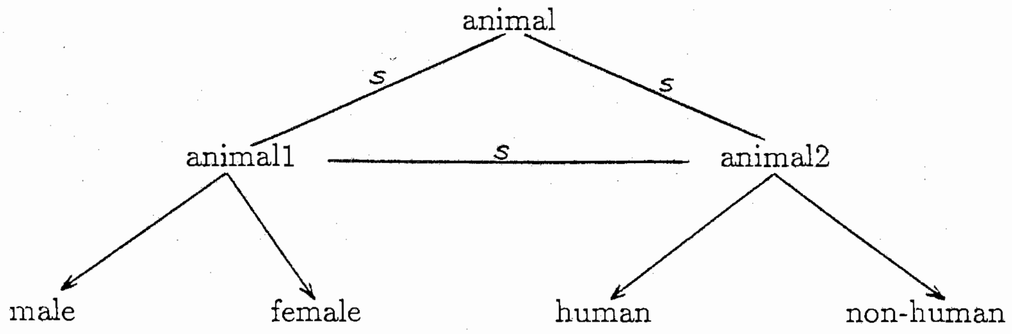
In proving theorems involving quantified formulas, it is often necessary to "match" certain subexpressions. For example, in order to produce $W_2(A)$ from the formulas $(\forall x)[\ W_1(x) \Rightarrow W_2(x)]$ and $W_1(A)$, it is necessary to find the substitution " A for x " that makes $W_1(A)$ and $W_1(x)$ identical. Finding substitutions of terms for variables to make expressions identical is an extremely important process in Artificial Intelligence and is called *unification*.

However, the usual unification process is based just on an "exact" matching. In order to take full advantage of the semantic network defined in the previous chapter during the query retrieval process, it is necessary to conceive a more general type of unification corresponding to a "similar" matching.

Indeed, in what follows we will use the semantic network to describe a similarity relation among terms on which unification is operated. Thus, by taking into account the similarity relation in the semantic network, we define a *similarity unification* (denoted *s-unification* for short) which reflects certain aspects of a similar matching more general than an exact matching.

II.1 Definition of the Similarity Unification

Let's consider as an example of a semantic network described in the previous chapter, the semantic network represented by the following graph :



This semantic network provides for two different views (linked by s-links) – i.e. two different partitions – "animal1" and "animal2" of the same set of objects regrouped under the concept "animal". In fact, "animal1" is partitionned (by g-links) into the two semantic categories "male" and "female" whereas the second view "animal2" allows to distinguish between the semantic categories "human" and "non-human".

One should notice that here, unlike in [Skim,Mink79], an element of the extension of "animal" has not to be classified either as male or female, thus being more flexible for updating the extensions of concepts.

In order to generalize the usual concept of unification to the semantic network defined above, we proceed as follows.

First of all, we define an equivalence relation on the set of semantic categories (nodes of the semantic graph represented above) such that two nodes linked by a *s-link* belong to the same equivalence class.

In each equivalence class C, we select a unique node of the semantic network $rep(C)$ that will represent that class. For instance, for the semantic graph above, the semantic categories *animal*, *animal1*, *animal2* belong to the same equivalence class and we can choose for instance the semantic category *animal* in order to represent that class.

Henceforth, we define the *similarity unification* (s-unification) for ordinary semantic categories at the level of the semantic network as follows :

1/ Two variables x and y are *s-unifiable* as $x' = y'$ where $x' = rep(Cl(x))$ (resp. $y' = rep(Cl(y))$) denotes the unique representative of the equivalence class containing x (resp. y).

2/ A constant "male" and a variable x are *s-unifiable* as $x' = rep(Cl(\text{male}))$ where x' is defined as before and where $rep(Cl(\text{male}))$ denotes the unique representative of the equivalence class containing male.

3/ For two constants c and d, nodes of the semantic network :

   a) if c and d are equivalent, then they are s-unifiable as $rep(Cl(c))$

   b) otherwise, there exists c' (resp. d') such that there is a path in the semantic network from c (resp. d) to c' (resp. d') involving either a similarity link or one or more generalization links and such that c' and d' belong to the same equivalence class. Then, $rep(Cl(c'))$ s-unifies c and d.

Thus, for instance, in the semantic network described previously, if we assume that animal = rep ({ animal, animal1, animal2 }), then the two semantic categories *male* and *human* will be s-unifiable as the semantic category *animal* whereas they would not have been unifiable by the usual unification process.

Moreover, this s-unification can be extended quite easily to the more complex type of semantic category that we have described in the first chapter in section I.3.1.

Let's recall here that a general formulation of a semantic category is of the form :
S = $([D^E,(D^X)],D^I)$ where D^E is the set of constants *explicitly included* in S, D^X is the set of constants *explicitly excluded* from S and D^I is the set of constants represented as a Boolean category expression and *implicitly included* in S.

Moreover, the following conditions :

$$D^E \cap D^I = \emptyset \quad \text{and} \quad D^X \subseteq D^I$$

should be satisfied.

Hence, we would like to extend our definition of s-unification to the general categories described above.

Definition Let's consider the two general semantic categories :

$$S_1 = ([D^{E_1},(D^{X_1})],D^{I_1})$$

$$S_2 = ([D^{E_2},(D^{X_2})],D^{I_2})$$

We have defined previously the s-unification of the semantic categories D^{I_1} and D^{I_2}. Let's denote it $D^I = S-UNIF(D^{I_1},D^{I_2})$.

Then, let's consider

$$D^X = D^{X_1} \cup D^{X_2}$$
$$D^E = D^{E_1'} \cup D^{E_2'}$$

where $D^{E_1'} = D^{E_1} - (D^{E_1} \cap D^I)$
and $D^{E_2'} = D^{E_2} - (D^{E_2} \cap D^I)$

Then, S = ([D^E, (D^X)], D^I) is a general semantic category which s-unifies the two general semantic categories considered above.

N.B. : We must point out here that the category S = ([D^E, (D^X)], D^I) satisfies the constraints forced on general semantic categories since
$D^X \subseteq D^I$ [$D^{X_1} \subseteq D^{I_1}$, $D^{X_2} \subseteq D^{I_2}$ and by definition of s-unification for usual semantic categories, D^I is a superset of both D^{I_1} and D^{I_2}]
and $D^E \cap D^I = \emptyset$ [by considering $D^{E_1'}$ and $D^{E_2'}$, we discard elements which may appear simultaneously in D^{E_1} and D^I or in D^{E_2} and D^I, thus avoiding redundancies].

For instance, if we want to s-unify the two general semantic categories

([{Mary}, ({ John, Toby})], male)

i.e. an element of that category is either Mary or male but is not John nor Toby (Toby is supposed to be a male dog).

and ([{Toby, chair}, ({Helen})], human)

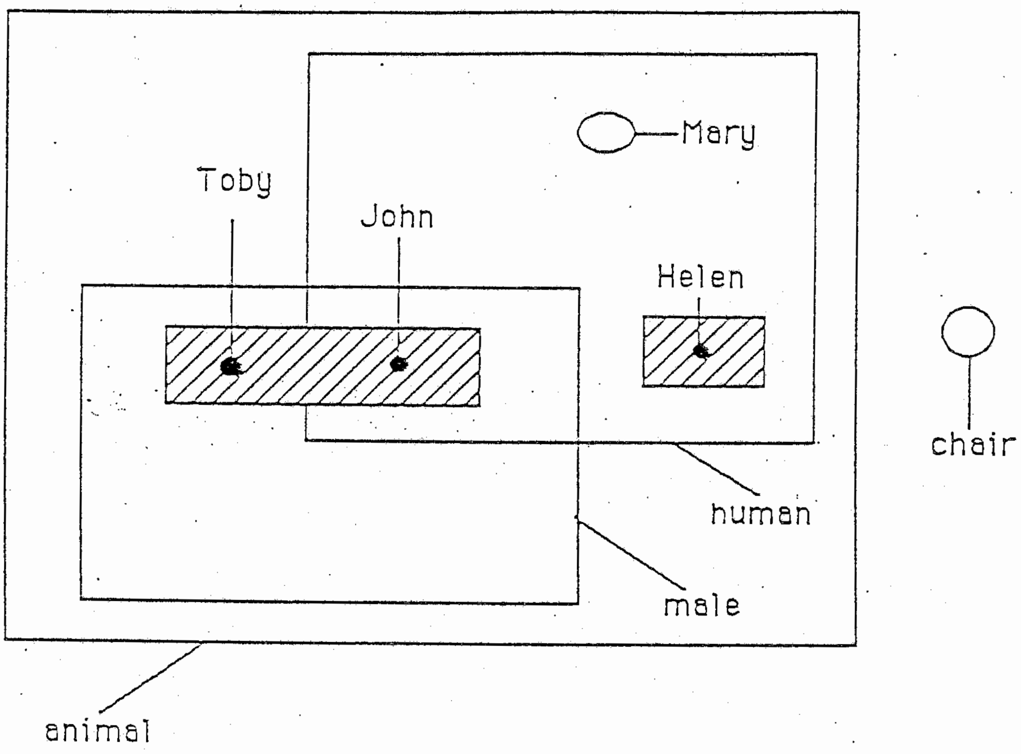
i.e. an element of that category is either Toby or a chair or any human being who is not Helen.

We will obtain as s-unifier of these general categories the general semantic category :

([{chair}, ({ John, Toby, Helen})], animal)

i.e. an element of that semantic category is either a chair or an animal which is not John nor Toby nor Helen.

We can notice that it is no longer necessary to add the constant "Mary" explicitly since it is implicitly included in the semantic category *animal*. On the other hand, although "Toby" is explicitly included in the second semantic category, since it is explicitly *excluded* from the first one no match involving "Toby" could possibly succeed and thus "Toby" must be *explicitly excluded* from the general semantic category resulting from the s-unification.



II.2 S-Unification of Predicates and Clauses

Hitherto, s-unification has only been defined at the level of the semantic network. In order to be able to answer queries, we have to extend this definition at the level of predicates and clauses.

II.2.1 S-Unification of Predicates

Here, we assume that a given predicate has only one name. Let P and Q be two predicate names.

1/ If P and Q are variables, they are s-unifiable as P=Q.

2/ If P=P_0 is a constant and Q is a variable, they are s-unifiable as Q=P_0.

3/ If P=P_0 and Q=Q_0 are both constants, then if $P_0 = Q_0$ they are s-unifiable as P_0, otherwise the s-unification is impossible.

We would like to stress here that in our semantic network a predicate is defined in the dictionnary by the number of arguments and the semantic constraints holding on each of its arguments. Hence, if we consider a binary predicate LIKES(x,y) where x is bound to the semantic category *male* and y to the semantic category *female*, and another binary predicate $LIKES^*$(x,y) where x is bound to the semantic category *human* and y to the semantic category *animal*, these two predicates – although they may capture the same concept – will be considered as distinct and will not be s-unifiable.

Therefore, predicates should be defined at the highest possible level in the network hierarchy – that is to say that semantic constraints on arguments of a predicate should be restricted as less as possible according to the concept represented by that predicate. For instance, in the example above, the predicate LIKES(x,y) should be restricted to x and y both belonging to the semantic category *animal*.

II.2.2 S-Unification of Clauses

As explained in the first chapter, we have chosen to represent assertions as in [Skim,Mink79], that is to say using the extended Π-clause representation. For instance, the notation

$$(\alpha, x, y) \{ [P]/\alpha, S/x, T/y \}$$

means that for every x belonging to the general semantic category S and for every y belonging to the general semantic category T, we have the property P(x,y).

Let's consider two extended Π-clause

$$(\alpha, x, y)\ \{\ [P]/\alpha,\ S_1/x,\ T_1/y\ \}$$

$$(\beta, z, t)\ \{\ [Q]/\beta,\ S_2/z,\ T_2/t\ \}$$

We would like to know under what condition these two clauses are s-unifiable.

1/ First of all, a necessary condition is that the two predicates P and Q should be s-unifiable.
Let R = S-UNIF (P,Q) be their s-unifier.

2/ If the first step is successful, then we can s-unify the general semantic categories as it has been defined previously :

$$S = \text{S-UNIF}\ (S_1, S_2)$$

$$T = \text{S-UNIF}\ (T_1, T_2)$$

Then, the two extended Π-clauses are s-unifiable as

$$(\gamma, u, v)\ \{\ [R]/\gamma,\ S/u,\ T/v\ \}$$

We present below a few examples illustrating the s-unification process of two clauses.

Example 1

$$(\alpha, x, y)\ \{\ [\text{LIVES}]/\alpha,\ \textit{animal1}/x,\ \textit{country}/y\ \}$$

$$(\beta, z, t)\ \{\ [\text{LIVES}]/\beta,\ \textit{animal}/z,\ \textit{country}/t\ \}$$

are s-unifiable as :

$$(\gamma, u, v)\ \{\ [\text{LIVES}]/\gamma,\ \textit{animal}/u,\ \textit{country}/v\ \}$$

Example 2

$$(\alpha, x, y)\ \{\ [\text{LIVES}]/\alpha,\ \textit{human}/x,\ \textit{country}/y\ \}$$

$(\beta, z, t)\ \{\ [P]/\beta,\ \textit{male}/z,\ \textit{country}/t\ \}$ (where P is a variable)

are also s-unifiable as :

$$(\gamma, u, v)\ \{\ [\text{LIVES}]/\gamma,\ \textit{animal}/u,\ \textit{country}/v\ \}$$

Notice here that the s-unification is possible only if the predicate *LIKES* is defined at the highest possible level in the network hierarchy, that is as a binary predicate whose attributes are taking values in the semantic category *animal.*
Thus, the two clauses considered here will only be restrictions of a *unique* predicate to the subcategories *human* and *male.*

Example 3

$$(\alpha, \mathrm{x}, \mathrm{y}) \ \{ \ [\mathrm{LIVES}]/\alpha, \ \mathit{animal1}/\mathrm{x}, \ \mathit{country}/\mathrm{y} \ \}$$

$$(\beta, \mathrm{z}, \mathrm{t}) \ \{ \ [\mathrm{WORKS}]/\beta, \ \mathit{human}/\mathrm{z}, \ \mathit{country}/\mathrm{t} \ \}$$

are not s-unifiable since *LIVES* and *WORKS* denote different predicates.

In the following chapter, we present an application of this generalized unification which allows a better word-to-word correspondence in different languages.

III. AN APPLICATION OF THE GENERALIZED UNIFICATION TO A BETTER CORRESPONDENCE AMONG WORDS IN DIFFERENT LANGUAGES

We will consider here two different examples of word-to-word correspondence in different languages. The first one, between French and English, deals with vocabulary related to transportation matters envisionned from the point of view of a researcher attending an international conference. The vocabulary has been extracted from samples of key-board conversations focusing on that specific situation. This word-to-word correspondence has been implemented in Prolog and the program can be found in Annex 3.

The second example has been developped between English and Japanese and is likely to be easier to understand for japanese people. It underlines the different kinds of ambiguities which may arise during the translation from English to Japanese or from Japanese to English. This part has not been implemented but obviously enough, the last Prolog program of Annex 3 could easily be adaptated to achieve word-to-word correspondence in that setting.

III.1 English-French and French-English word-to-word correspondence

We consider for each language (French and English) a specific semantic network related to the domain of transportation matters.
Since, if the semantic network in the two languages are identical, s-unification is reduced to a trivial identification, we will concentrate on the case where the two networks are different – the corresponding Prolog program is given in the Prolog Application No2 contained in Annex 3.

Figure 1 and Figure 2 describe respectively the English and the French semantic networks. As shown in these diagrams, each semantic network provides for three different views of the semantic category *TRANSPORTATION* (called *TRANSPORT* in the French network). The first one deals with transportation information parted into *General Information* (*Information Generale*), *Schedule* (*Horaire*) and *Country* (*Pays*). Moreover, in the French network, the semantic category *Information Generale* is divided into the two semantic categories *voyetape* – containing words related to the details of a trip such as *itineraire* (*route* in English) – and *voyglobal* – which refers only to a trip in its whole.

The second view expresses the purpose of the trip and distinguishes between the semantic categories *Business* (*Travail*) and *Leisure* (*Loisir*). Eventually, the last view deals with the means of transportation : *air* (*air*), *sea* (*mer*) or *ground* (*terre*) type of transportation, the semantic category *ground* (*terre*) being itself divided into *rail* (*rail*), *vehicle* (*vehicule*) and *walk* (*marche*).

In each language, every semantic category contains a set of words of that language. Each of these words represents a *general concept* and has been chosen as a representative for the set of synonyms of the language expressing the same concept.

In Figure 3, we give the list of English and French words (including synonyms) that will be considered for the translation process. Given a general English concept – for instance *cost* –, the English synonyms

{ cost, price, charge, charges, expense, expenses, outlay, expenditure }

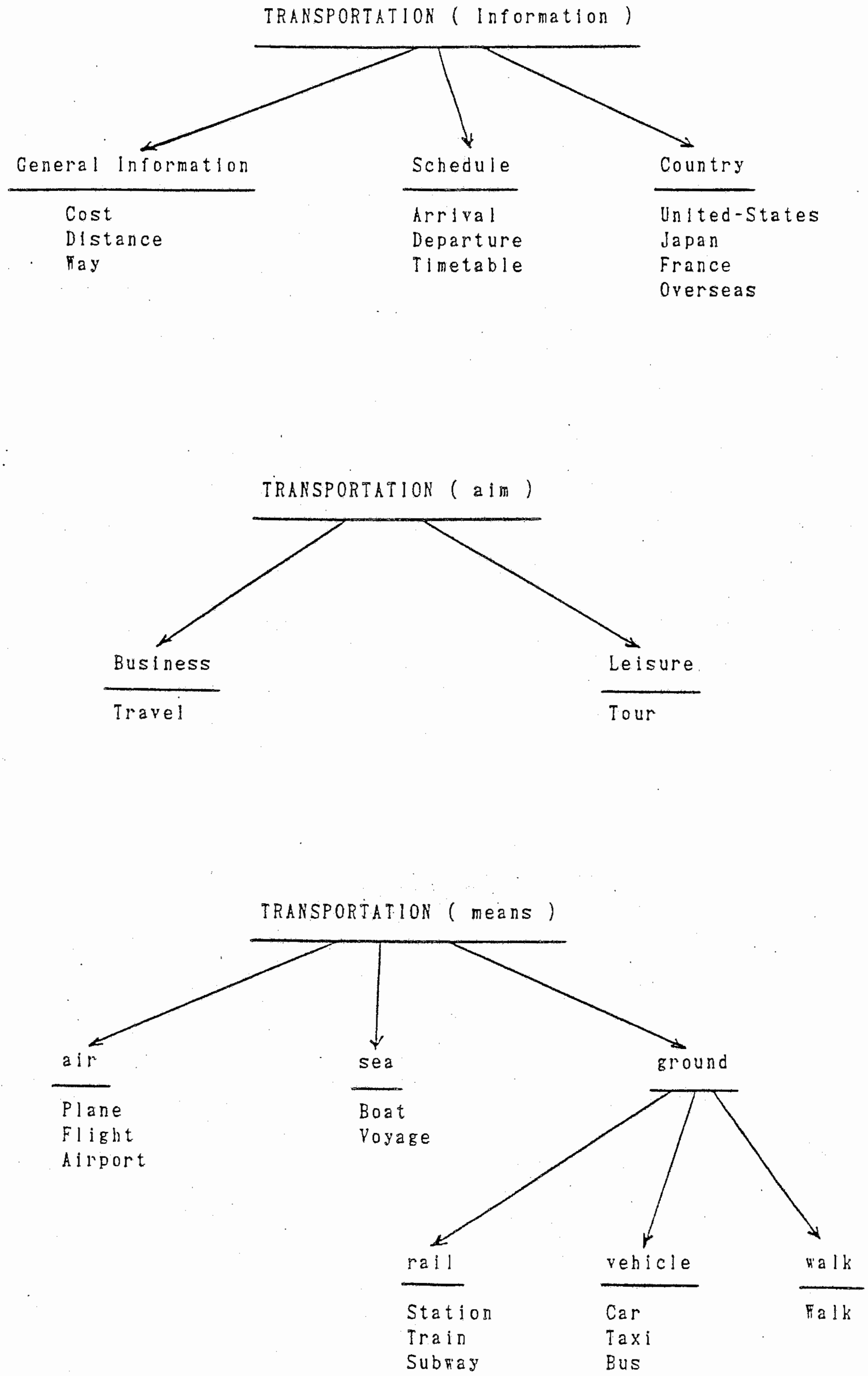
are regrouped and the English word “ cost ” is chosen as a representative of that general concept [in Figure 3, this word is underlined].
The French translation of “ cost ” being the French word “ coût ”, any of these English synonyms are considered to be equivalent as far as translation into French is concerned and they will be translated as “ coût ” in French.

Conversely the French words

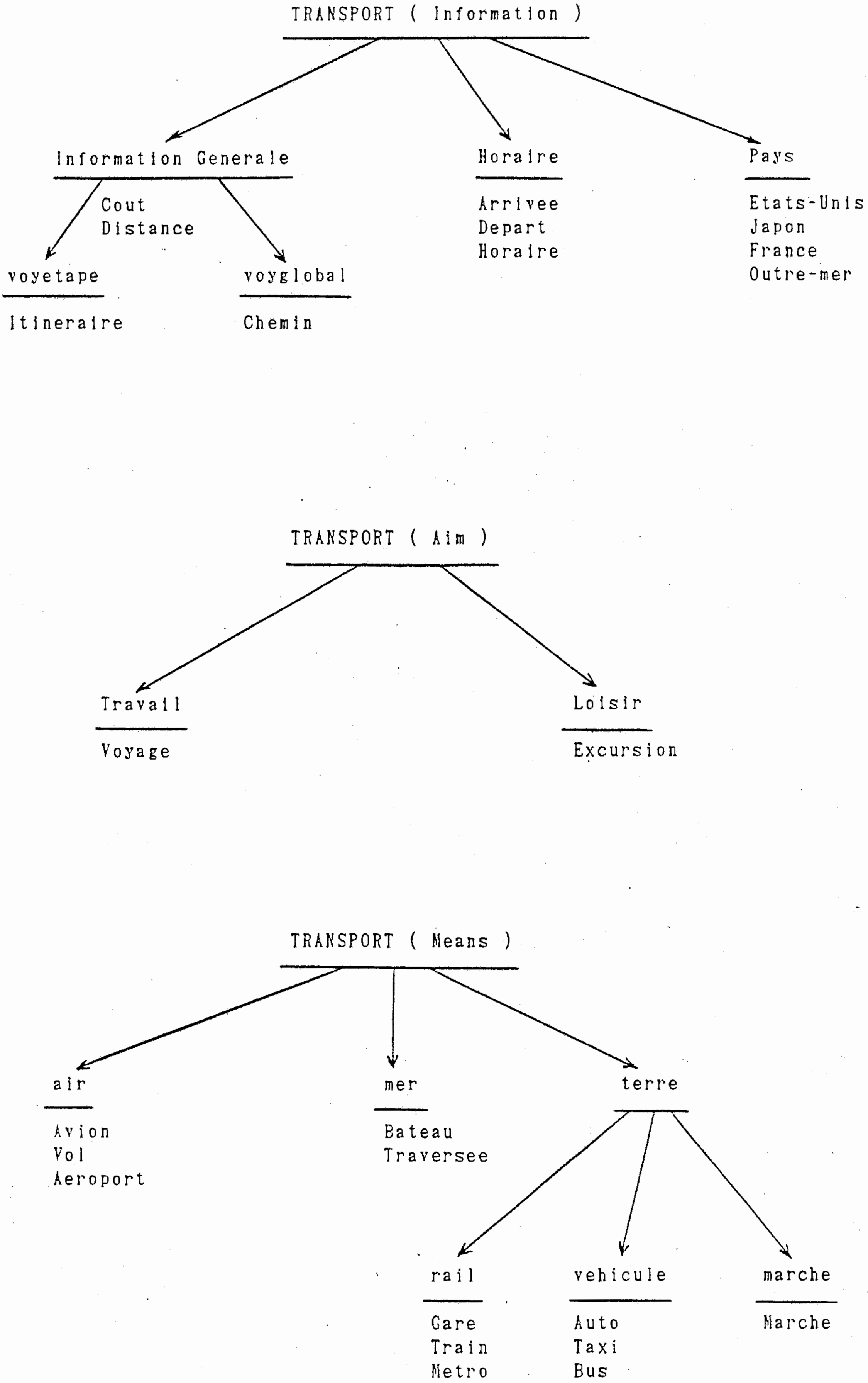
{ coût, prix, frais, dépense, débours}

are all representing the same French concept *coût*.
Thus, being identified to the French word “ coût ” during the translation process, they will all be tranlated into English by the word “ cost ”.

****
* Figure 1 : The English semantic network TRANSPORTATION *
****



****
* Figure 2 : The French semantic network TRANSPORT *
****



****
* Figure 3 : French-English word correspondence *
****

FRENCH	ENGLISH
******	*******
Arrivee	Arrival
Depart	Departure
Horaire	Timetable
*******	*********
Indicateur	Schedule
Emploi du temps	
Cout	Cost
****	****
Prix	Price
Frais	Charge(s)
Depense	Expense(s)
Debours	Outlay
	Expenditure
Distance	Distance
********	********
Eloignement	Interval
Chemin	Way
******	***
Route	Road
Trajet	Path
	Route
Itineraire	Route
**********	*****
Route	Itinerary
Parcours	
Voyage	Travel
******	******
Trajet	Trip
	Tour
	Journey
	Distance

Excursion	Tour
*********	****
Tour	Trip
Voyage	Excursion
Randonnee	Ramble
Promenade	

Etats-Unis (d' Amerique)	United States (of America)
**********	*************
U.S.A.	U.S.A.
U.S.	U.S.
Amerique	America
Japon	Japan
*****	*****
France	France
******	******
Outre-mer	Oversea(s)

Avion	Plane
*****	*****
Aeroplane	Aeroplane
	Aircraft
Vol	Flight
Aeroport	Airport

Bateau	Boat
******	****
Navire	Ship
Traversee	Voyage
*********	******
Voyage par mer	

Gare	Station
Train *****	Train ***** Railway-train
Metro *****	Subway ****** Underground Tube
Auto **** Automobile Voiture	Car *** Automobile
Taxi ****	Taxi (-cab) **** Cab
Bus *** Autobus Car Autocar	Bus *** Coach
Marche ****** Promenade a pied Tour	Walk **** Walking Stroll

Besides, if we consider for instance the semantic category *BUSINESS*, in that category the word " travel " has been chosen in order to represent the set of words :

{ Travel, Trip, Tour, Journey, Distance }

which are all different ways of expressing the same general concept.

In that semantic category, any of these words are viewed as equivalent as far as translation into French is concerned. That is to say that in the context of the semantic category *BUSINESS*, any of these words will be translated in French by the word " voyage ".

However, in the semantic category *LEISURE*, the general concept of sightseeing travel is expressed by the set of words :

{ Tour, Trip, Excursion, Ramble }

and is represented by the English word " tour " which will be translated by " excursion" in French.
Thus, the English word " tour " can be translated in French in two different ways : as " voyage " in the semantic category *BUSINESS* or as " excursion " in the semantic category *LEISURE*.

Hence, given an English (or a French) word, the Prolog program in Annex 3 will provide – in a first step – for the set of all possible translations of this word, depending on the context.

For instance, for the English word " tour ", we get two possible French translations :

(1) as a synonym of " travel " in the semantic category *BUSINESS*, we obtain the French translation " voyage ".

(2) as a synonym of " tour " in the semantic category *LEISURE*, we obtain the French translation " excursion ".

Thus, { Voyage, Excursion } is the set of possible French translations of the English word " tour ". Any additional information (semantic knowledge) about what semantic category should be considered will help to rule out some of these possibilities.

We define above under which conditions an English translation of a given French word will be *semantically correct.* Quite obviously, the same definition applies to define conversely the *semantically correct* French translations of a given English word.

Definition

An English word will be a *semantically correct* translation for a given French word if it fulfills the following conditions :

(1) This word is among the possible English translations of the French word.

(2) If we possess some extended knowledge about the semantic category to which the French word belongs, then :

    (a) if the semantic category to which the French word belongs can be matched with a semantic category of the English semantic network, then the English word must belong to that semantic category or to one of its subcategories.

    (b) otherwise, we go upwards in the French semantic network until we find a French semantic category which can be matched with an English semantic category. Then, the English word must belong to that semantic category or to one of its subcategories.

Example

Let's consider the English word " tour " in the Prolog program (Application No 2) in Annex 3. eqvfr (tour, Y) provides for all the French possible translations, i.e. we obtain :

{ Voyage, Excursion }

Moreover, if we have the information :

wdcat (tour, leisure)

we can discard some of the possible translations above and eftrad(tour, Y) provides for the unique *semantically correct* French translation " excursion ".

Other examples are given at the end of the program in Annex 3.

III.2 English-Japanese and Japanese-English word-to-word correspondence

This example has been elaborated for the presentation of these results at the conference in Hokkaido since the attendance of that conference was mainly japanese.

As far as translation from Japanese to English and from English to Japanese is concerned, we can distinguish two kinds of ambiguities.

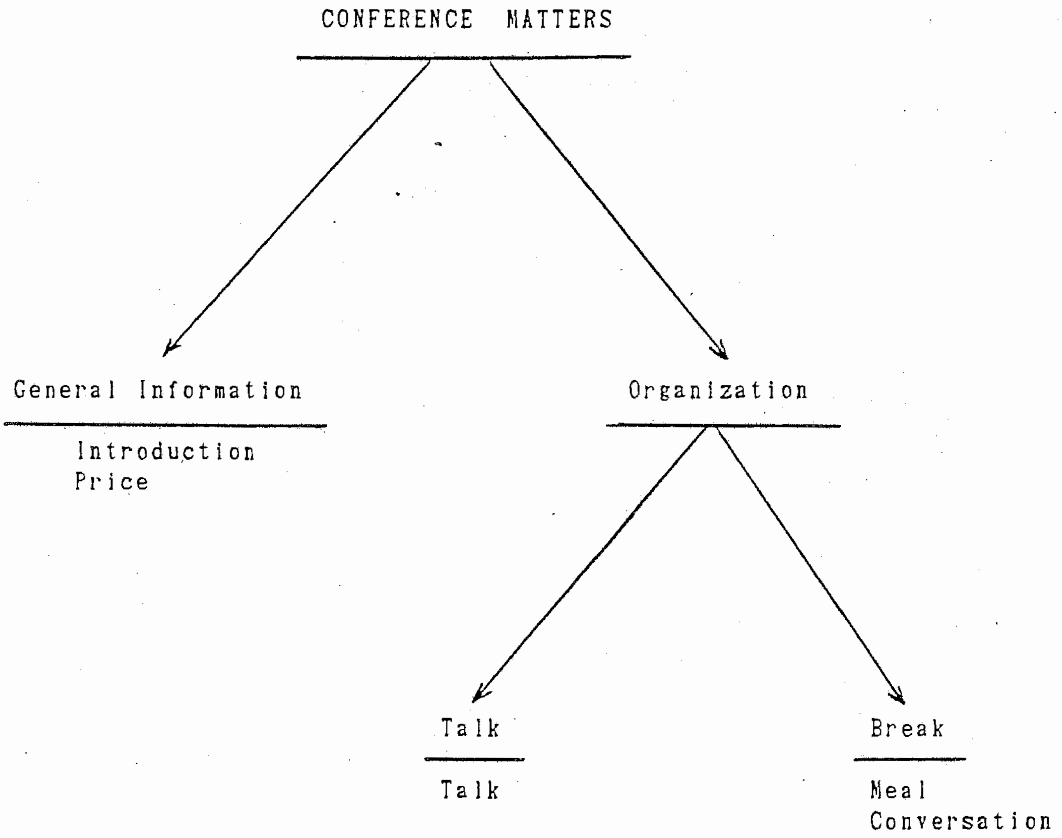
First of all, the Japanese language contains a lot of homonyms i.e. words whose pronunciation is identical but whose meaning is different. These homonyms can be easily recognized at the level of the written language since their kanjis are different. However, since we are interested in conversational speech, the meaning of a word will have to be extracted from the context in order to distinguish among the homonyms.

For instance, the English word " distance " can be translated in Japanese by " kankaku " which means " interval " but which can also mean " sensation ". Similarly, the word " form " (registration form) is translated in Japanese by " youshi " which can also mean " gist " (the main point of a speech).
This type of ambiguity has not been illustrated in the following example but it will clearly have to be considered in a future application.

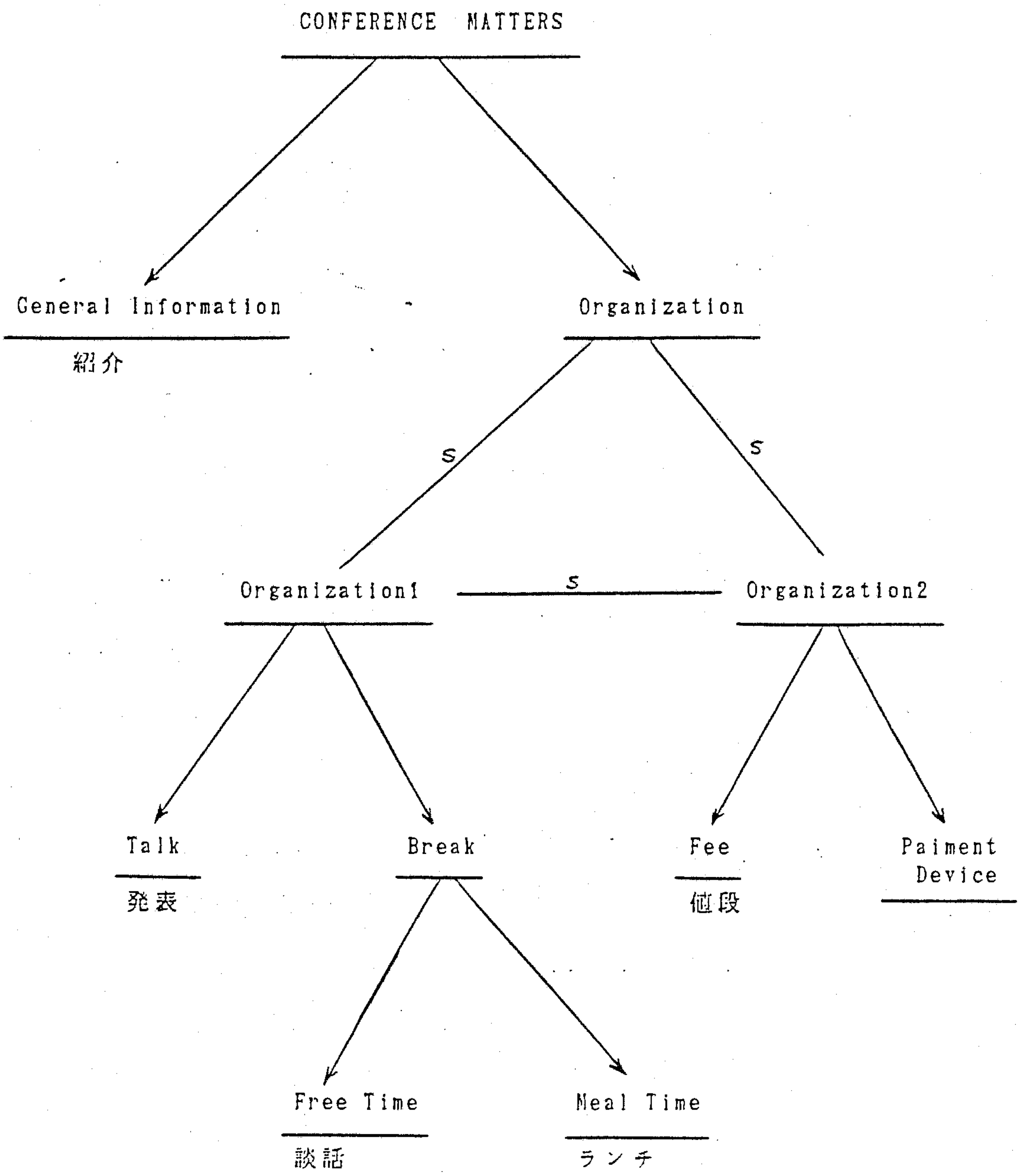
The second type of ambiguity – which has already been considered in the English-French and French-English application – concerns words who may have different meanings depending on the context.
Thus, for instance, if the English word " way " means " path " it will be translated in Japanese by " mitchi " whereas if it should be understood as " method " it will be translated by " houhou ".

In the following example, Figure 4 presents the English semantic network and Figure 5 the Japanese semantic network that have been selected to classify vocabulary related to the domain of conference matters.

****
* Figure 4 : The English semantic network *
****



****
* Figure 5 : The Japanese semantic network *
****



In that example, the English word " presentation " will be translated in Japanese by :

a/ " shoukai " if it is a synonym of " introduction ".

b/ " happyou " if it is a synonym of " talk ".

Therefore, if some semantic knowledge telling that " presentation " belongs to the semantic category *organization* is available, since in the Japanese semantic network

$$\text{S-UNIF (General-info, Organization)} \neq \text{Organization}$$

we discard " shoukai " as a possible Japanese translation.

Thus, by taking into account the additional semantic knowledge, we obtain the *unique* Japanese translation " happyou " for the English word " presentation ".

III.3 Conclusion

As underlined by the two examples considered above, the s-unification process helps disambiguating word-to-word correspondence in different languages, thus providing for a better translation process.

CONCLUSION AND
FURTHER RESEARCH PROSPECTS

Indeed, the generalized similarity unification presented here allows us to take advantage of semantical knowledge formalized by the semantic network presented in the first chapter. This formalization is very powerful and very convenient since it provides for different views of the same set of objects and allows to narrow the search space during the query answering process.

Moreover, we illustrated in the last chapter how the similarity unification could be applied towards a better word-to-word correspondence in different languages. Indeed, the semantic knowledge can then be used to discard some possible translations of a given word, thus providing for disambiguation of the word translation process.

Hitherto, that extra semantic knowledge has to be artificially introduced into the semantic network. Therefore, a further step for this research topic would be to consider full sentences or full part of speech on which some deductive process adding automatically that semantic knowledge could be elaborated.

Moreover, since semantic networks provide for a model of the organization of the human memory, I feel that their insertion into a neural network modeling human brains in order to achieve automatic translation should turn out to be profitable.

BIBLIOGRAPHIE

[Quil 68] Quillian M. R., 1968, Semantic memory. In *Semantic Information Processing.* M. Minsky (ed.). MIT Press, Cambridge, Massachusetts, pp. 227-270.

[Skim, Mink 79] McSkimin J. R., Minker J., 1979, A predicate calculus based semantic network for deductive searching. In *Associative Networks : Representation and Use of Knowledge by Computers.* N. V. Findler (ed.). Academic Press, New York, pp 205-238.

[Woo 75] Woods W. A., 1975, What's in a link ? Foundations for semantic networks. In *Representation and Understanding.* D. G. Bobrow and A. M. Collins (eds.). Academic Press, New York, pp 35-82.

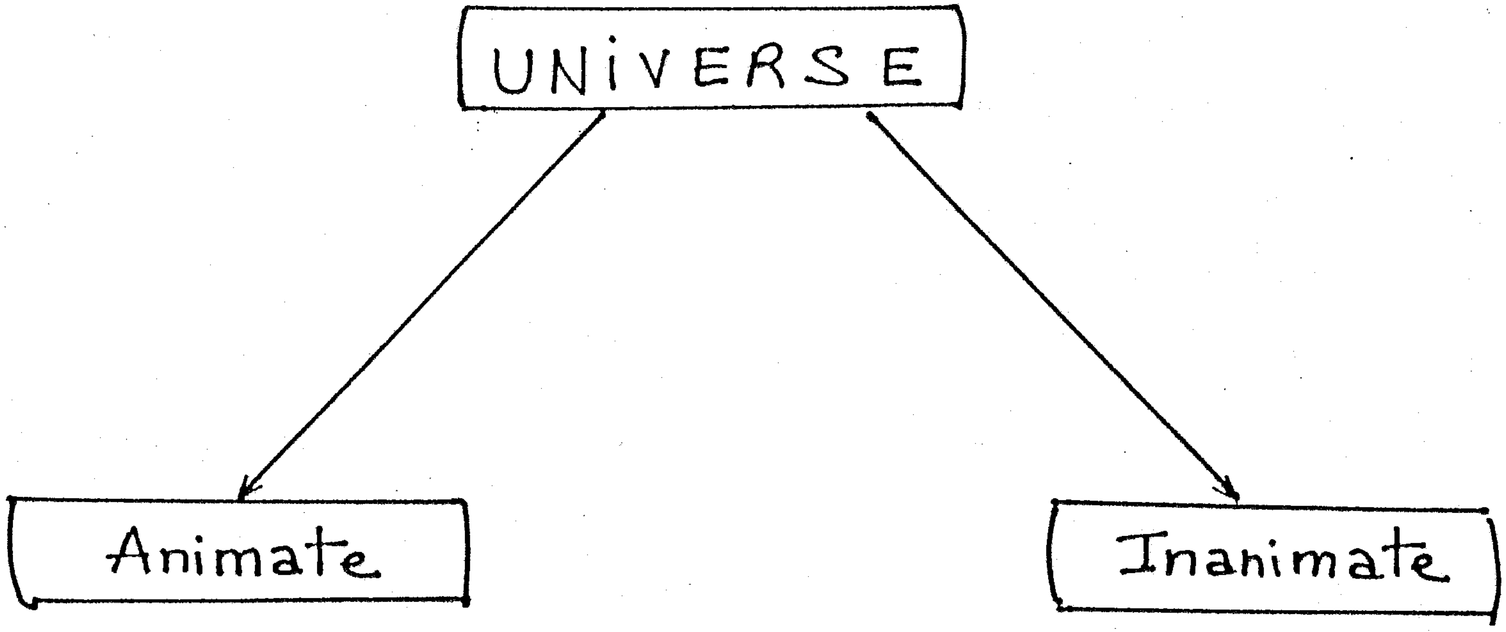
APPENDIX

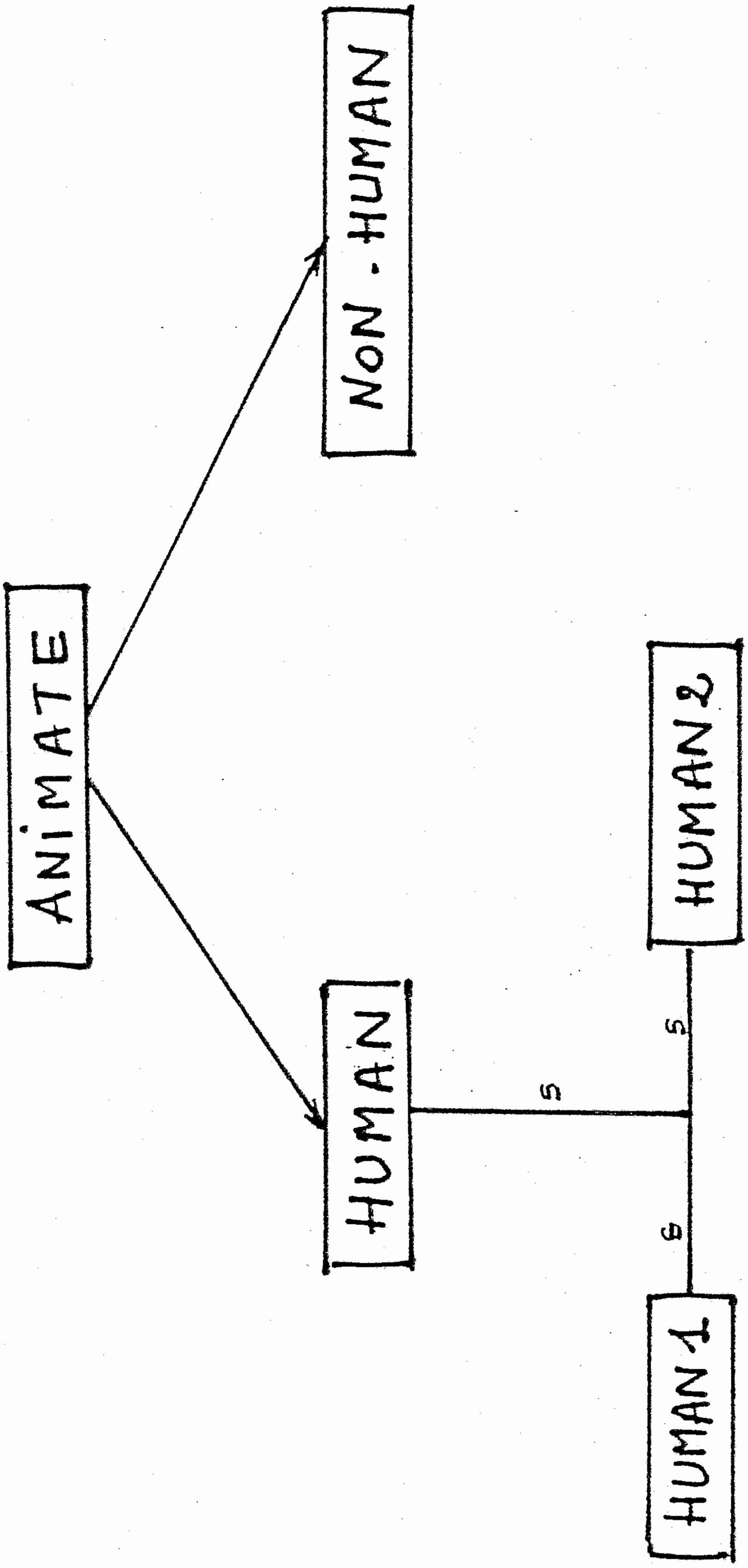
ANNEX No 1

In this section, we present the semantic graph which has been elaborated for dealing with topics concerning the conference domain.

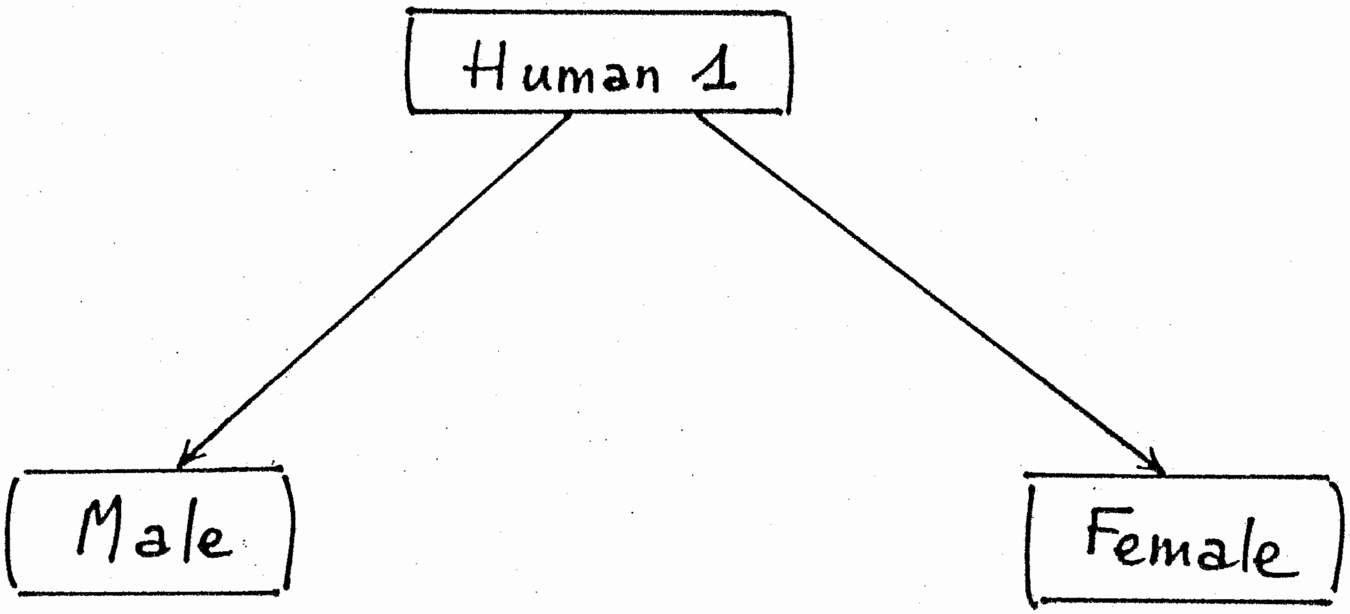
This graph has been elaborated after consultation of samples of English keyboard dialogue concerning that domain. The main occuring words have been listed and the structure of the semantic graph has been constructed according to that list and according to some features that should be stressed (for instance, the origin of people attending the conference represented through the hierachy of *Attendance2* is motivated by statistics purposes and also because usually the fee depends on the origin and the status of people).

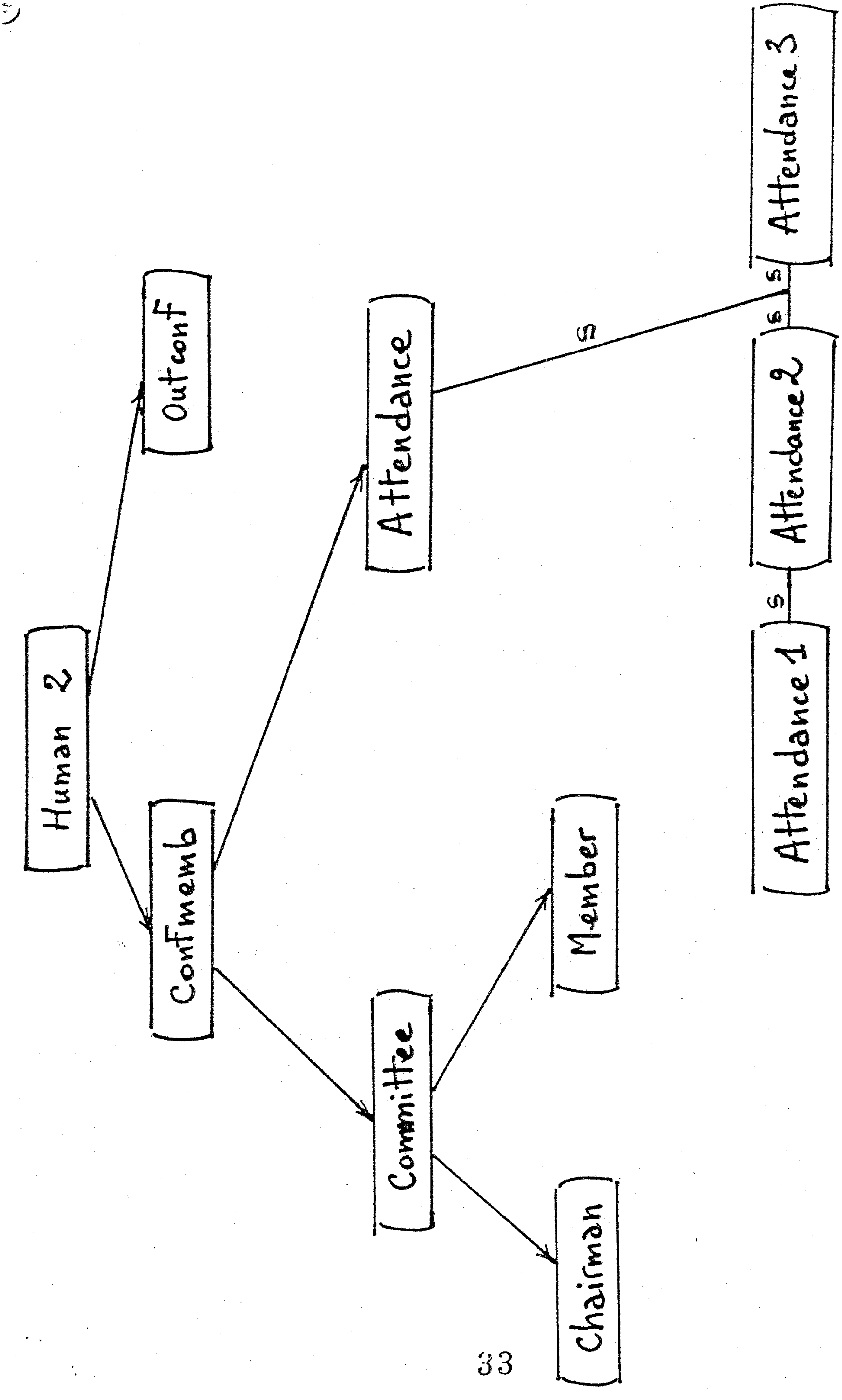
This representation does not pretend to be the best one, however I think it could be used with profit during the inference process as far as conference matters are concerned.

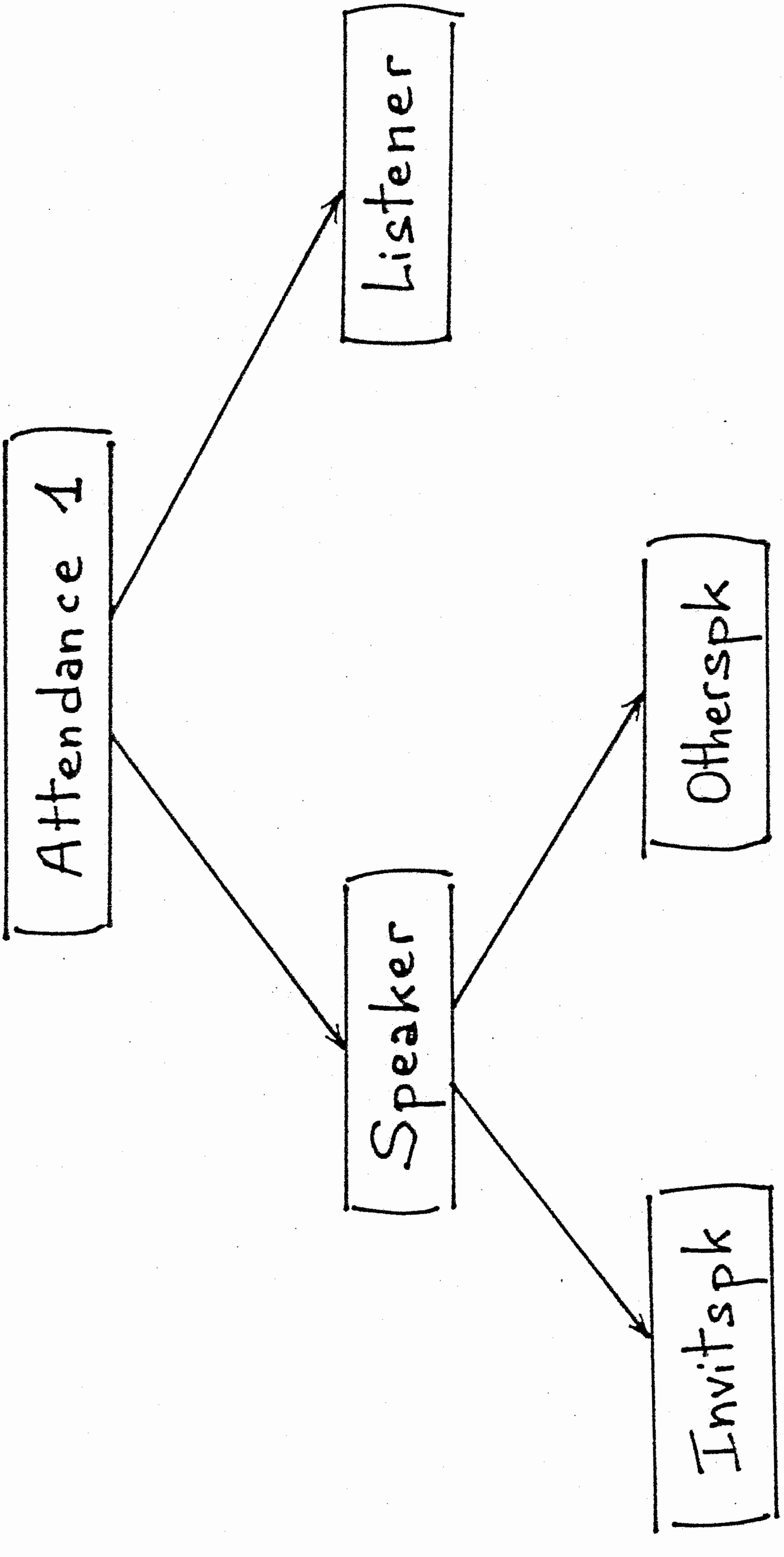

UNIVERSE
Animate
Inanimate

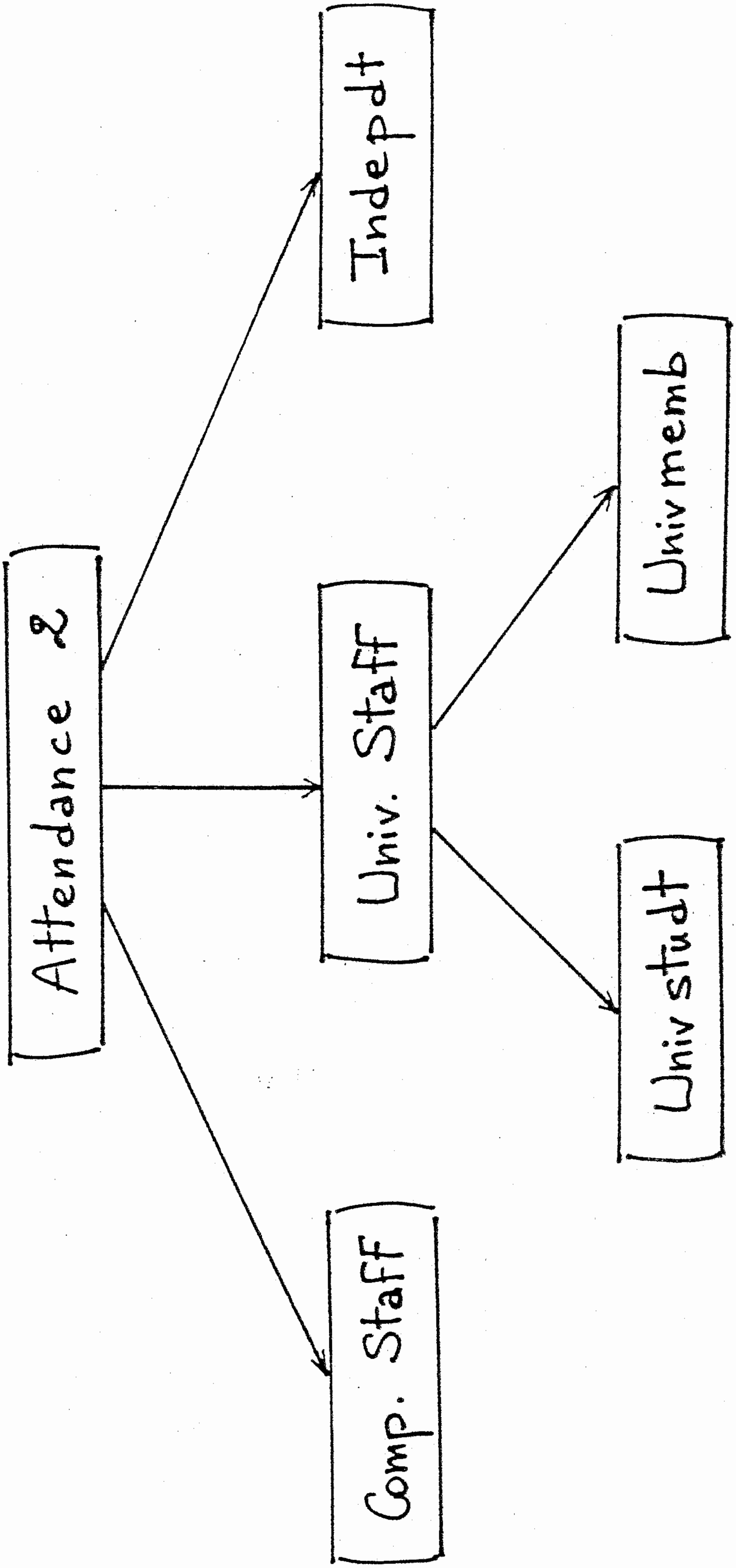

ANIMATE
NON-HUMAN
HUMAN
HUMAN 2
HUMAN 1
s
s
s

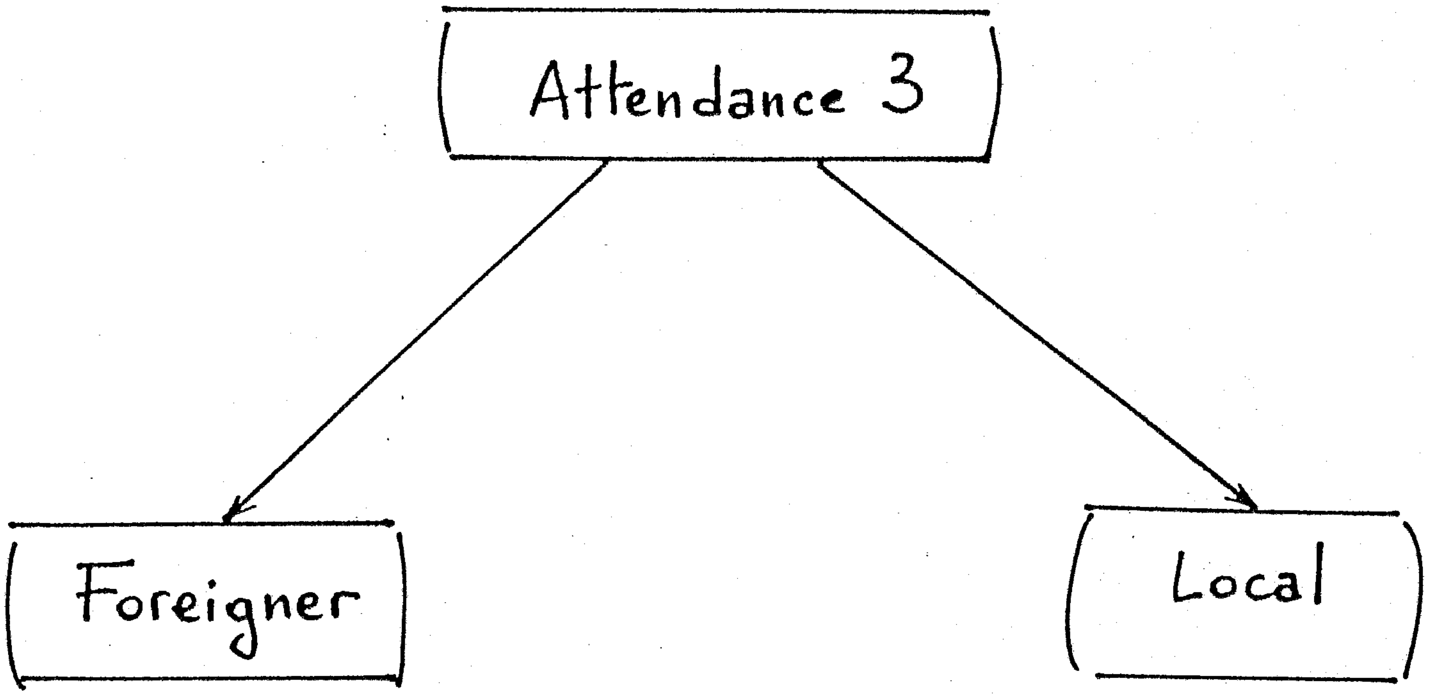
③

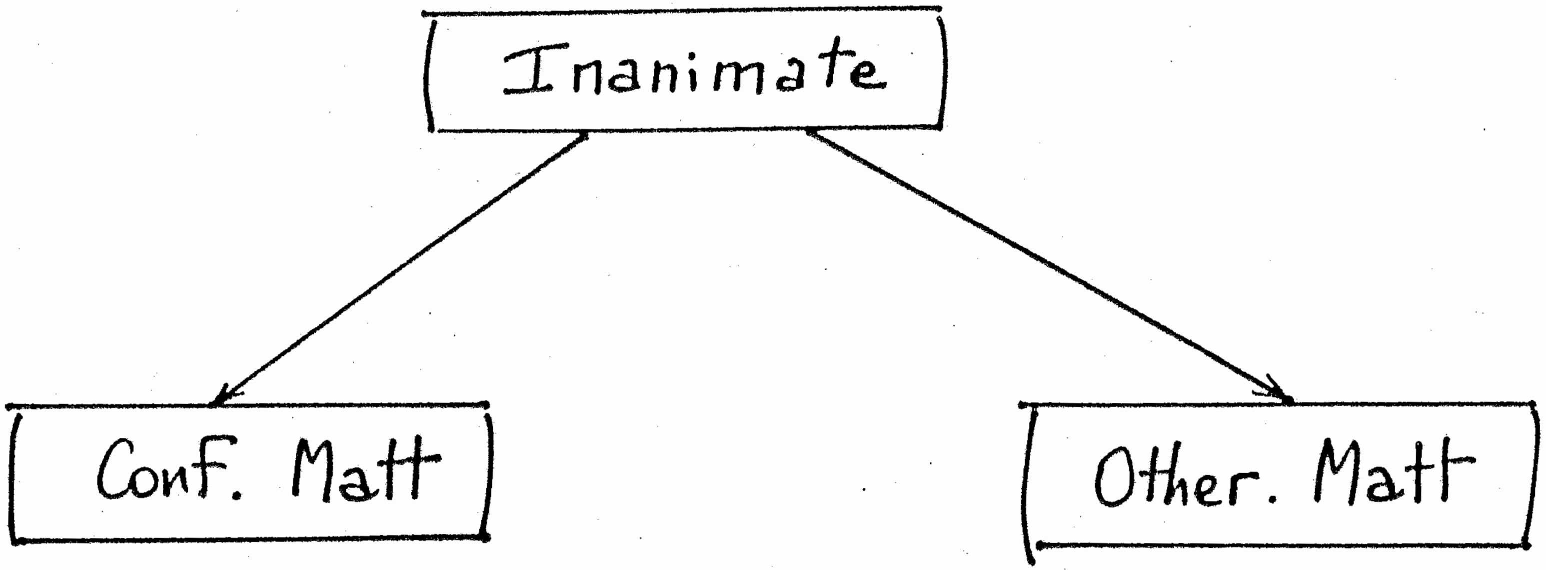


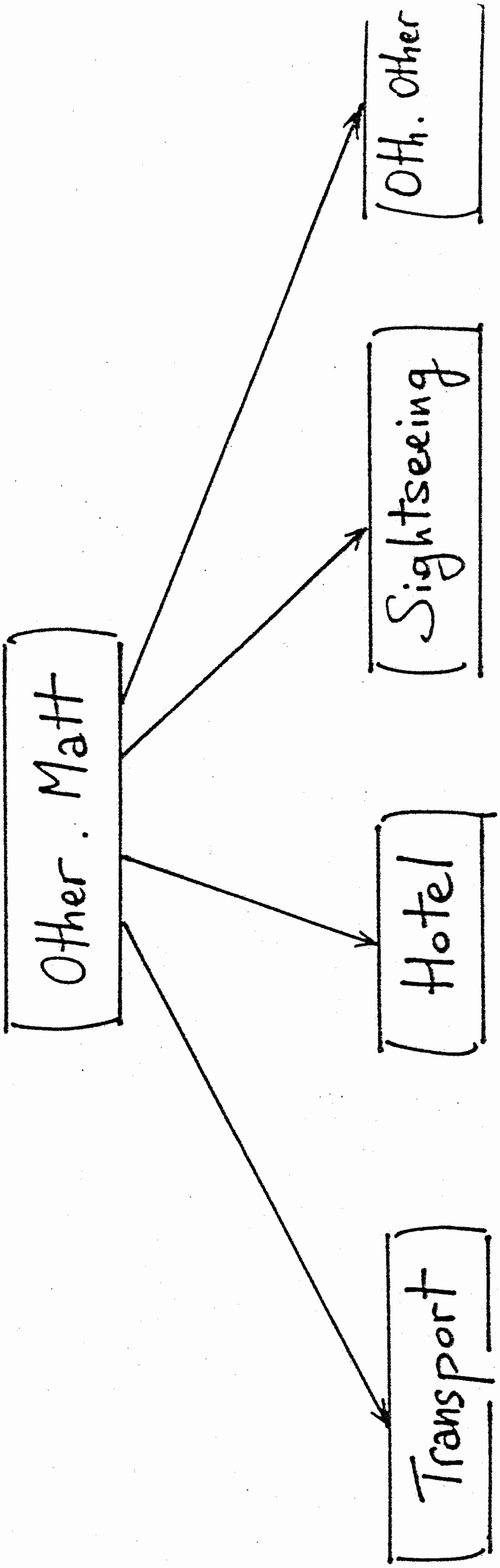

Human 2
Outconf
Confmemb
Attendance
Committee
Member
Chairman
Attendance 1
Attendance 2
Attendance 3
s
s
s
s

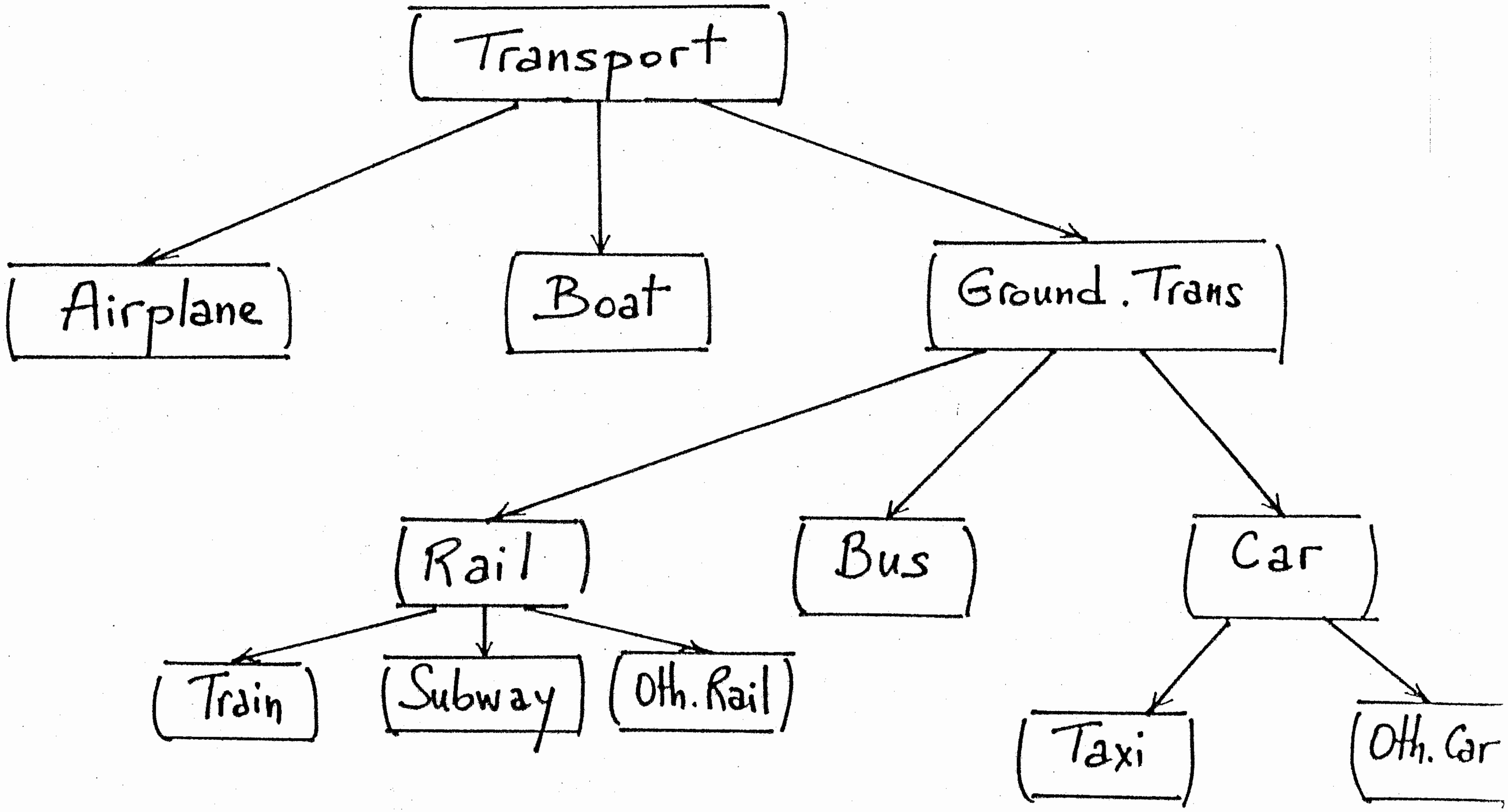

Attendance 1
Listener
Speaker
Otherspk
Invitspk

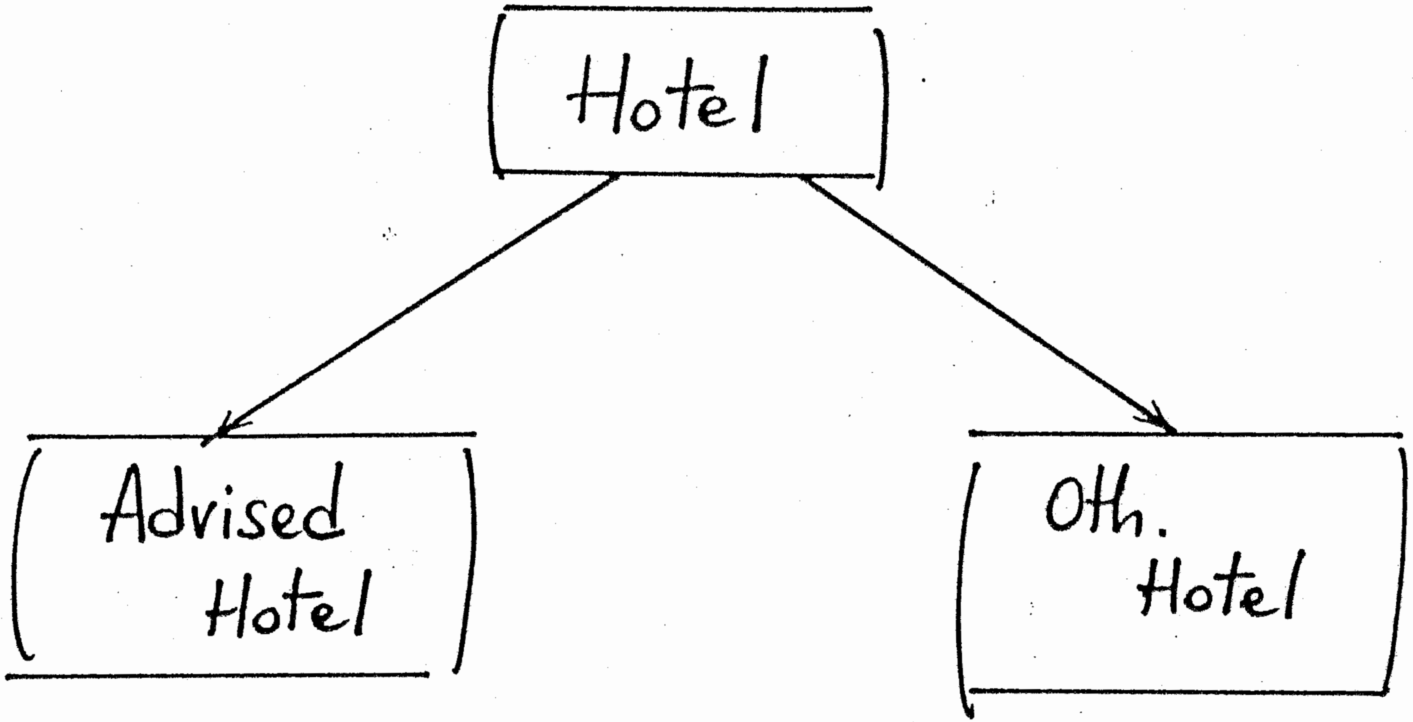

Attendance 2
Independt
Univ. Staff
Comp. Staff
Univmemb
Univstudt

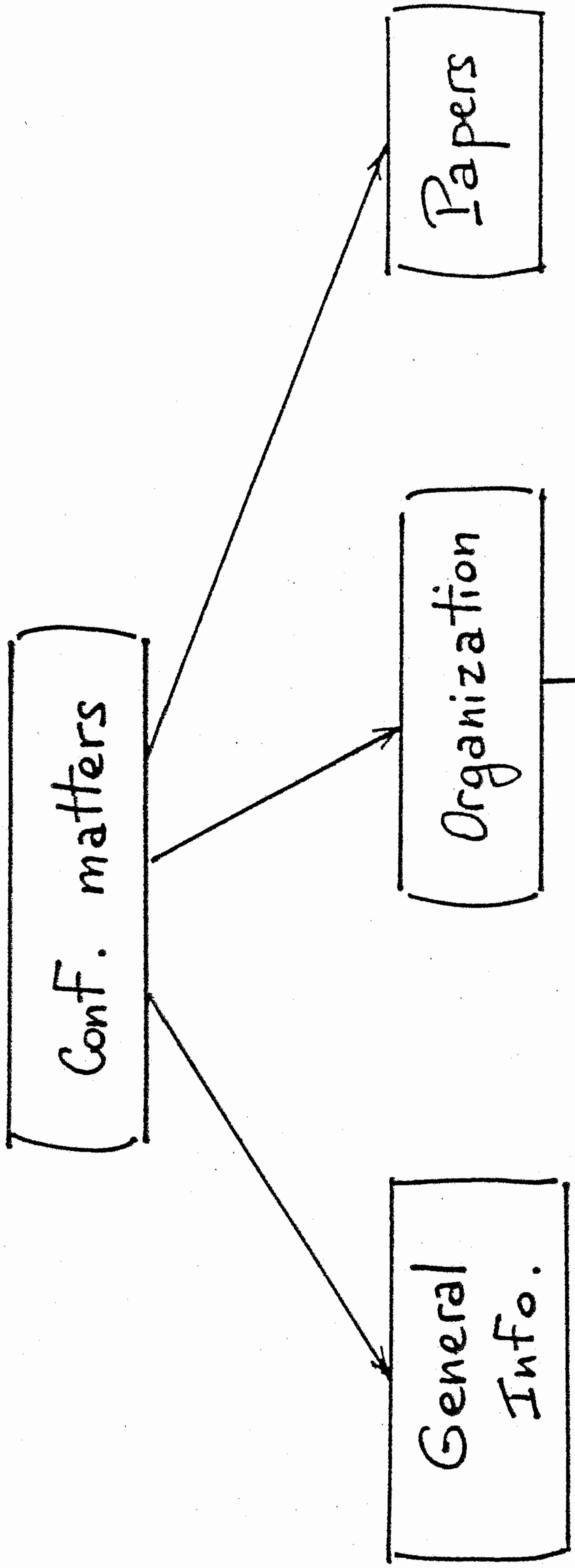

Attendance 3
Foreigner
Local

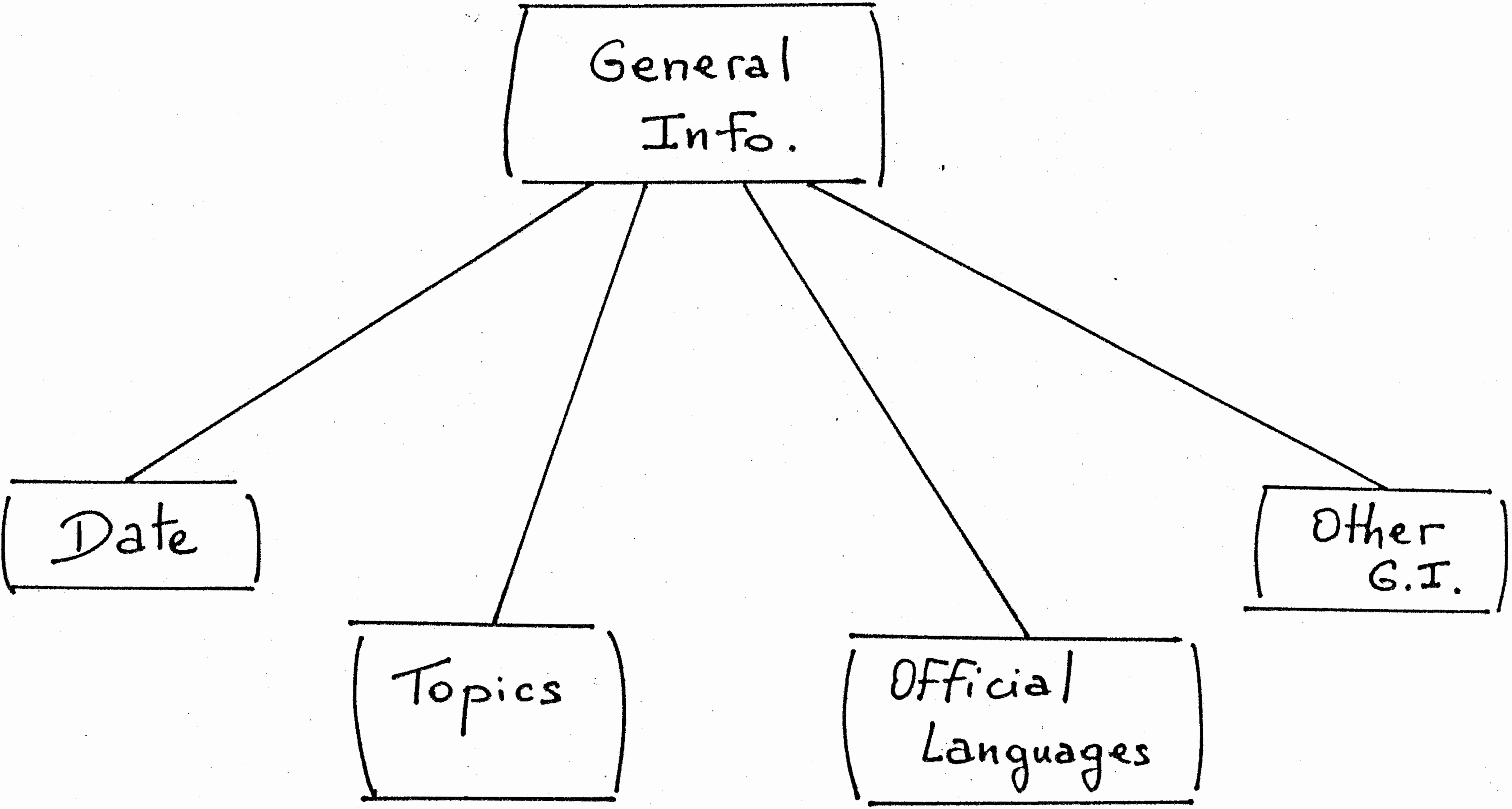

Inanimate
Conf. Matt
Other. Matt

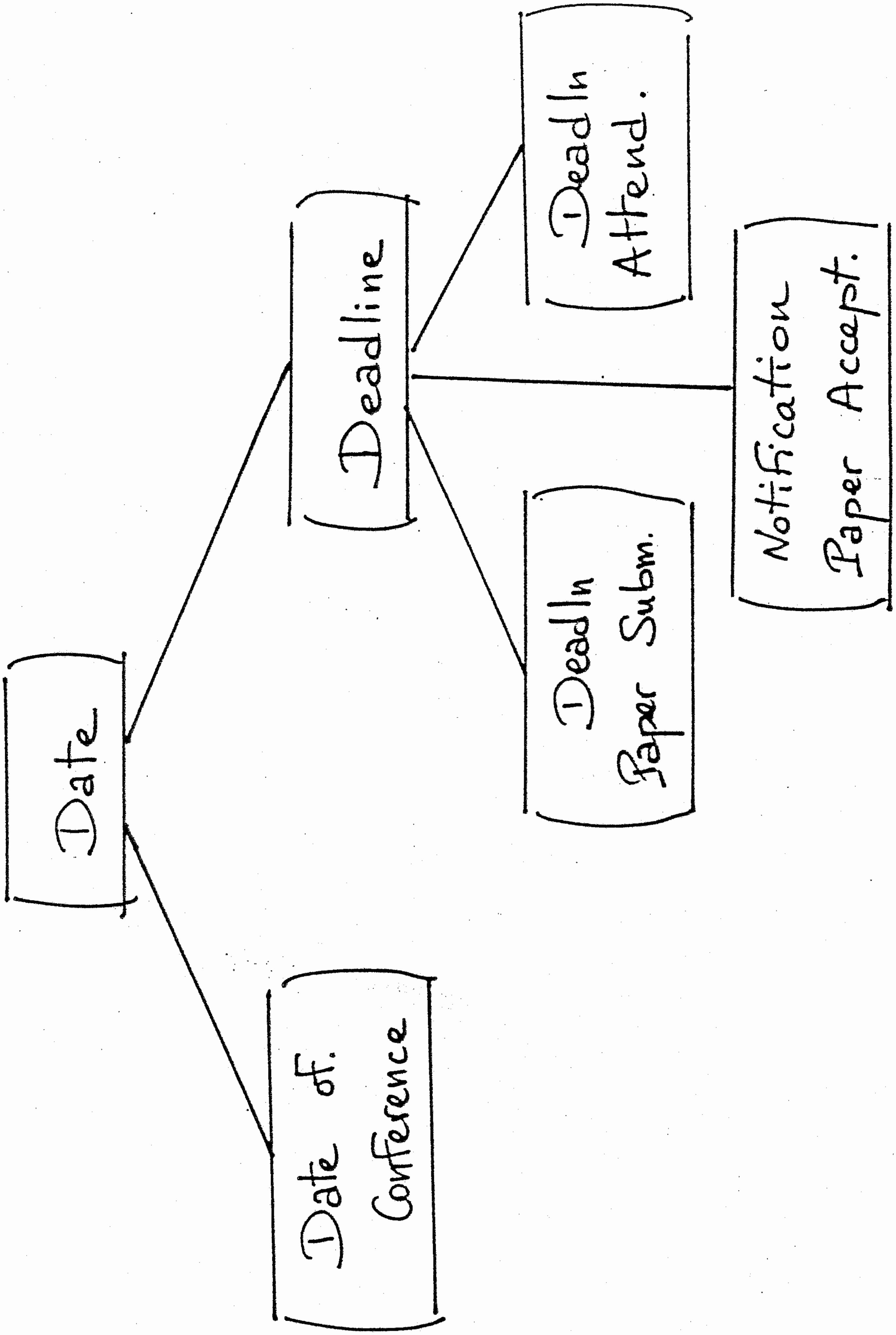

Other . Matt
Transport
Hotel
Sightseeing
Oth. Other

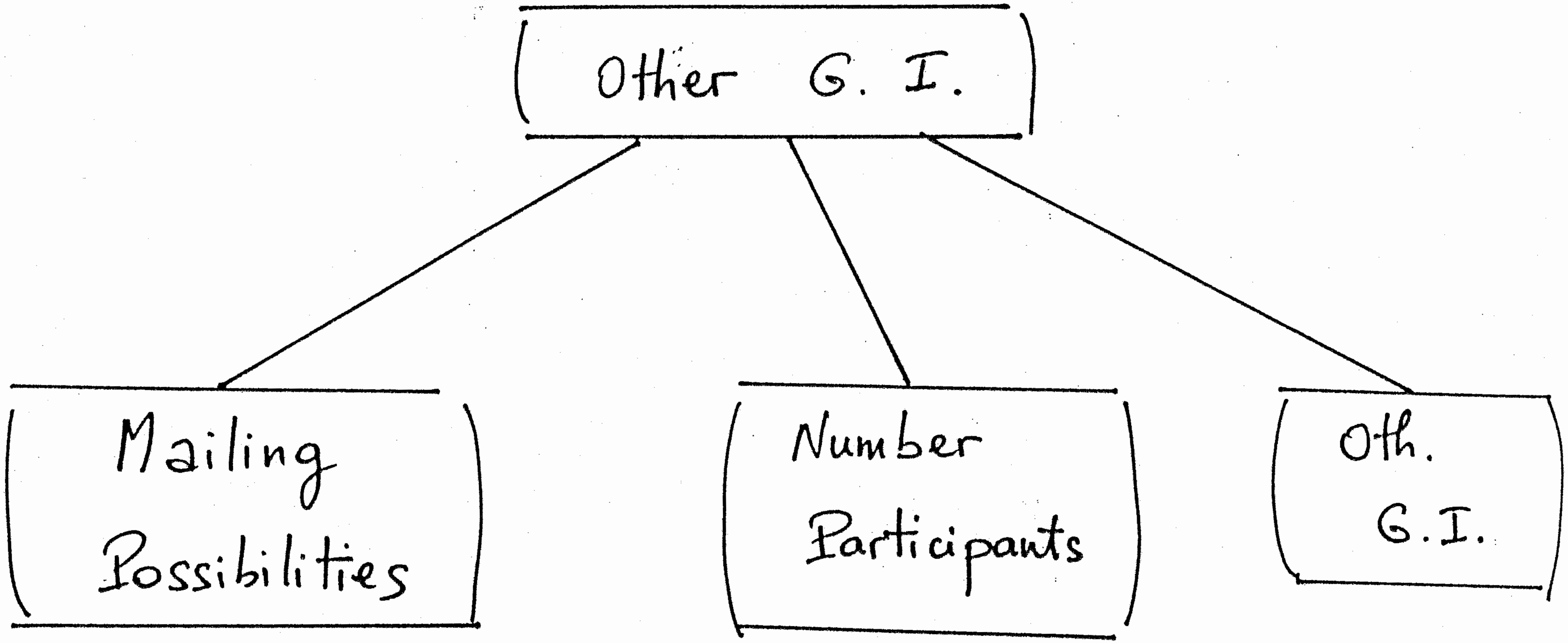

Transport
Airplane
Boat
Ground. Trans
Rail
Bus
Car
Train
Subway
Oth. Rail
Taxi
Oth. Car

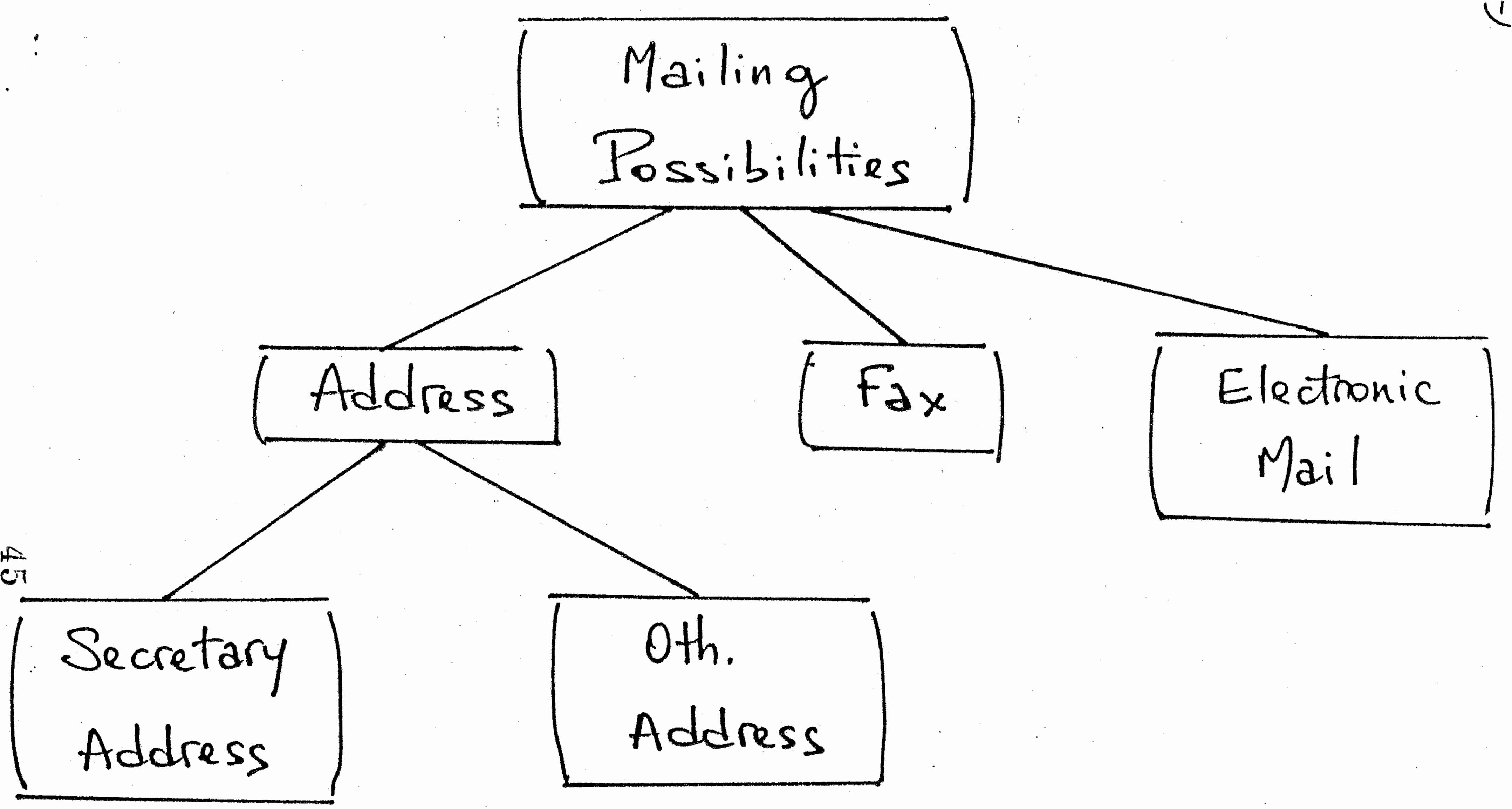

Hotel
Advised Hotel
Oth. Hotel


Conf. matters
Papers
Organization
General Info.
Organization 2
Organization 1
s
s
s

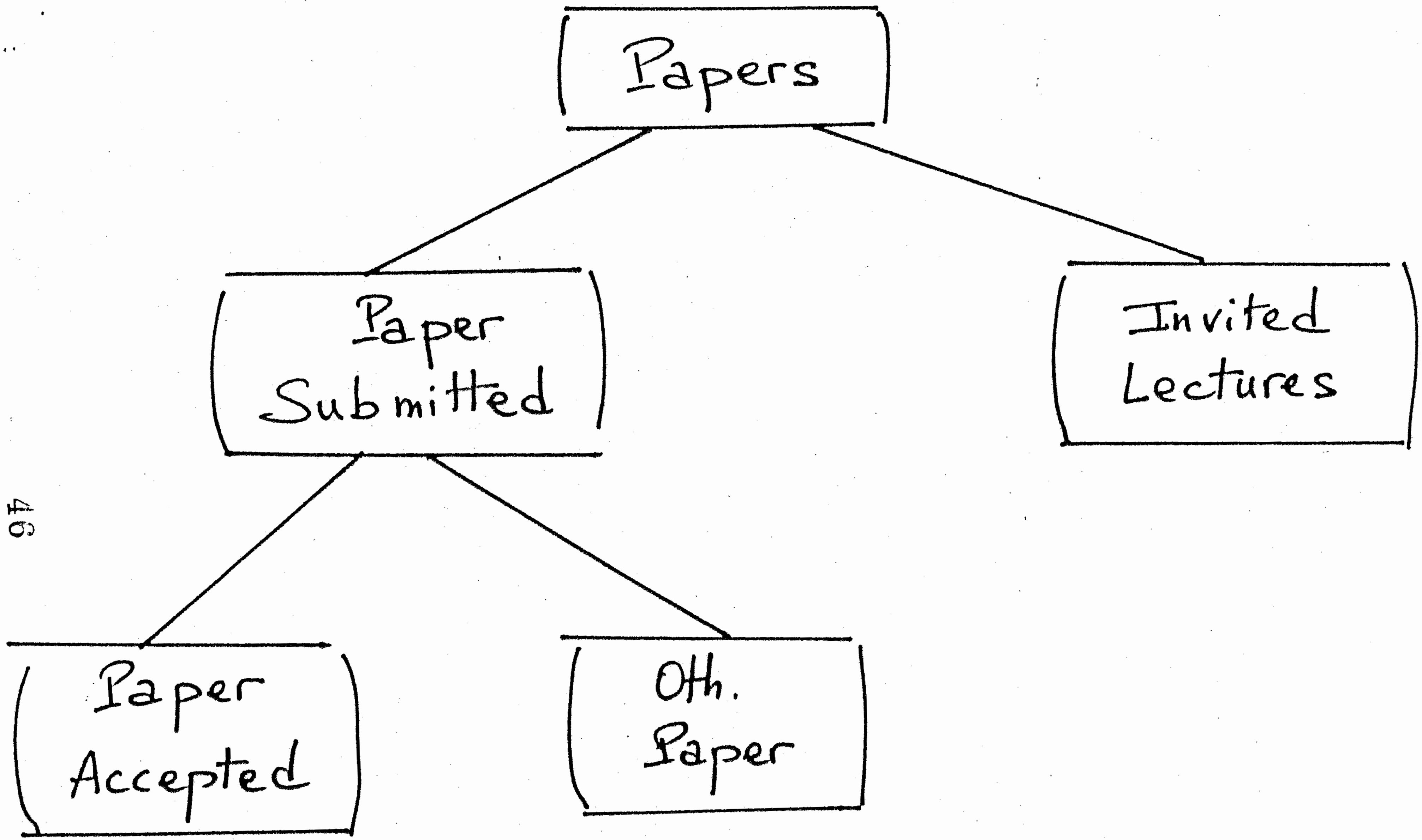

General Info.
Date
Topics
Official Languages
Other G.I.

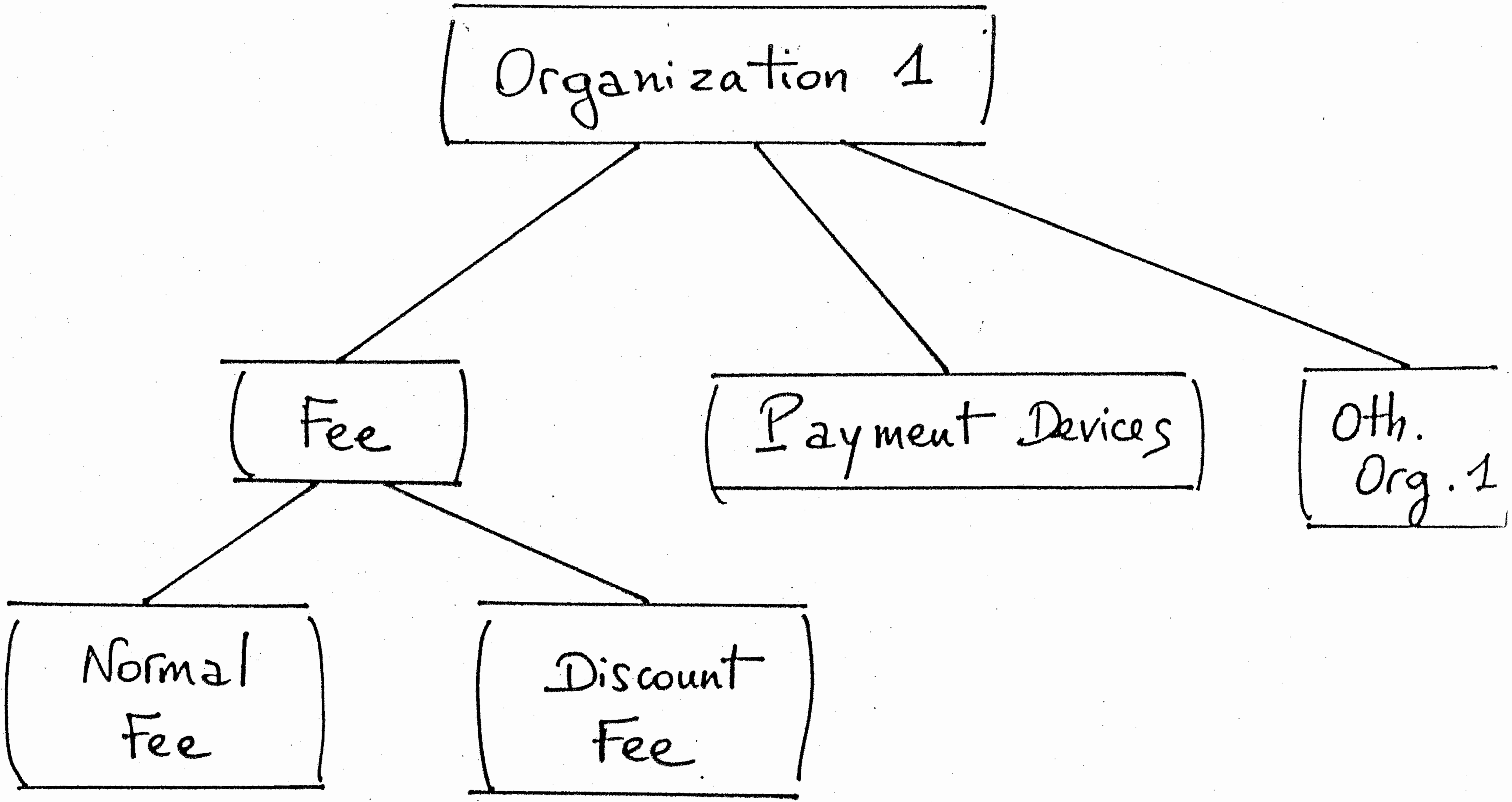

Date
Date of. Conference
Deadline
Deadln Paper Subm.
Deadln Attend.
Notification Paper Accept.

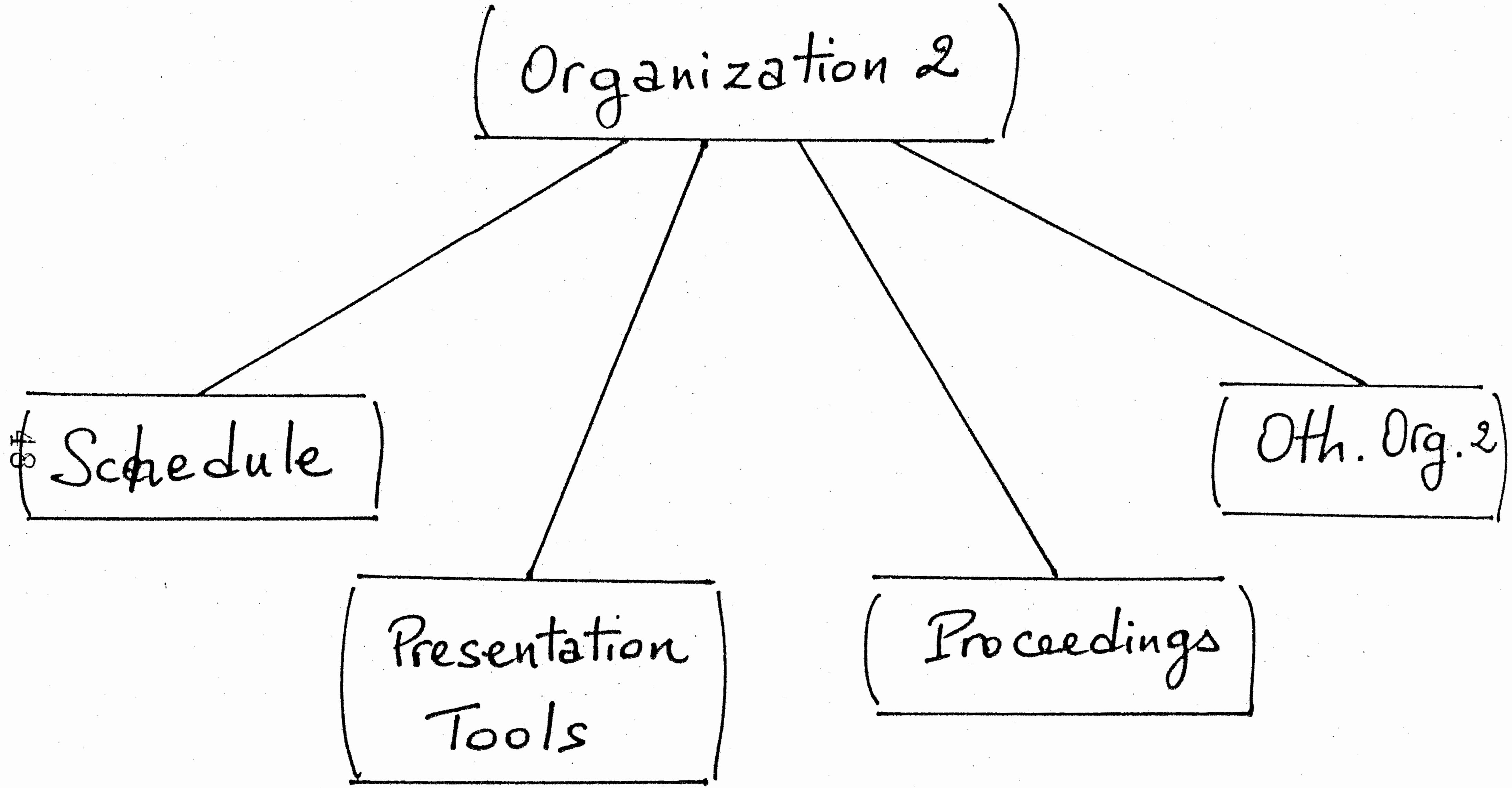

Other G. I.
Mailing Possibilities
Number Participants
Oth. G.I.

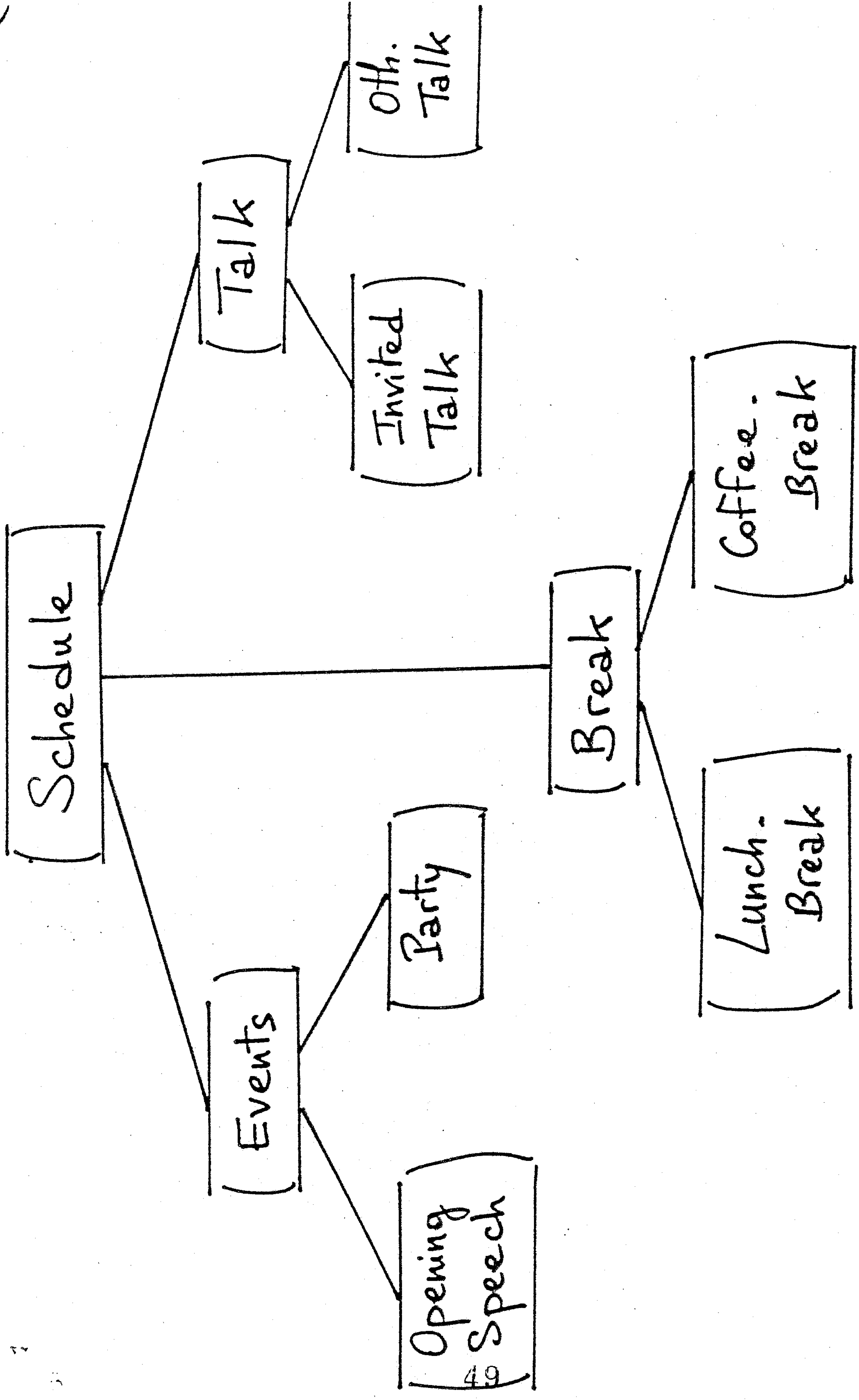

Mailing Possibilities
Address
Fax
Electronic Mail
Secretary Address
Oth. Address

i)


Papers
Paper Submitted
Invited Lectures
Paper Accepted
Oth. Paper


Organization 1
Fee
Payment Devices
Oth. Org. 1
Normal Fee
Discount Fee


Organization 2
Schedule
Presentation Tools
Proceedings
Oth. Org. 2


Schedule
Talk
Oth. Talk
Invited Talk
Break
Coffee. Break
Lunch. Break
Events
Party
Opening Speech

ANNEX No 2

This section contains a sample of the article that I presented at the conference of the "Information Processing Society of Japan" held in Hokkaido on the 28-30th September 1987.

1P-9

Redefining Unification in Semantic Networks towards Natural Language Understanding

Nadine Lerat[†] Teruaki Aizawa
ATR Interpreting Telephony Research Laboratories

1. Introduction

Semantic networks serve as a guide for information retrieval and thus turn out to be suitable for natural language understanding. In that prospect, the semantic network proposed in [McSk,Mk] is particularly attractive since it supports the full power of expression of predicate calculus and contributes to narrowing the search space during the deductive process. We define a less restrictive semantic network that may handle some cases of information incompleteness.

However, the usual unification based on "exact" matching cannot be applied to such a semantic network. Hence, we define a *similarity unification* (*s-unification* in short) based on "similar" matching, where a similarity relation in the semantic network links two nodes representing different views (i.e. different partitions) of the same concept. This s-unification is later used for defining a generalized unification at the level of clauses.

2. The Semantic Network (S)

A semantic network S is a finite tree whose nodes are intensions of concepts (i.e. their meaning as opposed to their extension - set of objects belonging to them, [Jan,Sw]) linked by labeled arcs specifying two types of relations :

(1) A node c is a "generalization" of a node d if the underlying extension of c contains the underlying extension of d. Then, the arc is labeled by "g".

(2) A node c is "similar" to a node d if their underlying extensions are equal. In that case, the arc is labeled by "s".

The similarity relation allows different nodes to represent the same concept, which corresponds to different views of that concept.

For instance, in the following semantic network "attend1" and "attend2" both represent the attendance of a conference but "attend1" distinguish between "speaker" and "listener" whereas "attend2" discriminate people w.r.t. their origin "company" or "university".

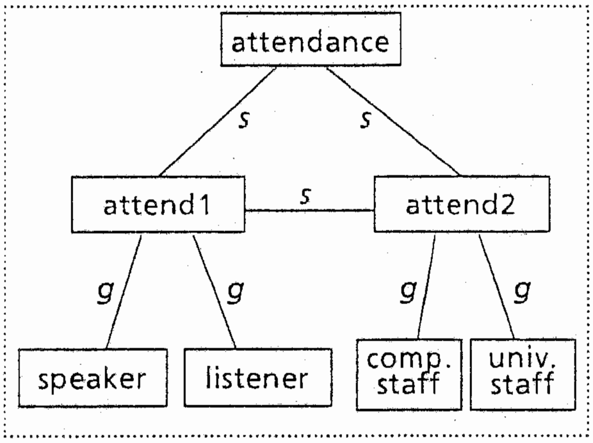


Fig. 1 Example of a Semantic Network

Notice that here, unlike in [McSk, Mk], an element of the extension of "attendance" has not to be classified either as company or university staff, thus being more flexible for extensions of concepts.

3. Similarity Unification

First of all, we define an equivalence relation on the set of semantic categories

---

[†] Intern Student of Ecole Nationale Supérieure des Télécommunications, Paris France

自然言語理解のための意味ネットワークを用いたユニフィケーションの一般化
留良那伝(ルラ ナディン)、相沢輝昭
ATR自動翻訳電話研究所

(nodes of the semantic network) such that two nodes linked by a "*s*" arc belong to the same equivalence class.
In each equivalence class C, we select a unique node of the semantic network, *rep*(C), that will represent that class.

Henceforth, we define the *s-unification* at the level of the semantic network S as follows:

(1) Two variables x and y are *unifiable* as $x' = y'$ where $x' = rep(Cl(x))$ (resp. $y' = rep(Cl(y))$) denotes the unique representative of the equivalence class containing x (resp. y).

(2) A constant "male" and a variable x are *unifiable* as $x' = rep(Cl(\text{male}))$ where x' is defined as before and where $rep(Cl(\text{male}))$ denotes the unique representative of the equivalence class containing male.

(3) For two constant c and d, nodes of S:
 a) if c and d are equivalent, then they are *unified* as $rep(Cl(c))$;
 b) otherwise, there exists c' (resp. d') such that there is a path in S from c (resp. d) to c' (resp. d') involving either a similarity link or one or more generalization links and c' and d' are equivalent. Then, $rep(Cl(c'))$ *unifies* c and d.

This s-unification can easily be extended to more complex semantic categories - introduced in [McSk,Mk] - as

([{Tom}, ({Brown})], speaker),

which denotes "either Tom or a speaker, Brown excepted".

4. General Unification of Clauses

A clause (a, x, y) / $\{[P]/a, S/x, T/y\}$ means that for any x in S and y in T, (S, T may be complex categories), $P(x,y)$ is true.

Therefore, in order to unify two clauses, we unify the predicates in the usual way (i.e. assuming uniqueness of predicate names, they are unifiable unless they denote different predicate names) and then replace the semantic categories by their s-unifier defined above.

For instance, in the network described in *Fig.1*, the two clauses:

(a, x, y) / $\{ATTEND/a$, speaker/x, conf/$y\}$
(b, z, t) / $\{ATTEND/b$, univ.staff/z, conf/$t\}$

are unifiable as:

$(c, u, v$ / $\{ATTEND/c$, attendance/u, conf/$v\}$

since "speaker" and "univ.staff" are s-unifiable as "attendance" (whereas they could not be unified by the usual concept).

5. Implementation and Research Prospect

The s-unification, discarding unmatchable literals w.r.t. the semantic network, narrows the search space in the resolution process.

Moreover, since the same concept may be viewed in many different ways, it is of crucial importance to include such a similarity in the semantic network.

Presently, we are investigating an application in Prolog of this unification process for word to word correspondence in different languages. For instance, the English verb *know* may be conveyed by *savoir* or *connaître* in French and the s-unification should help manipulating such a correspondence.

Indeed, the similarity relation that we introduced is likely to turn out to be very helpful in a machine translation system.

Acknowledgements The authors wish to thank Dr.Akira Kurematsu, President of ATR Interpreting Telephony Research Laboratories, for supporting this research.

6. References

[Jan, Sw] Janas & Schwind: "Extensional Semantic Networks: their represen-tation, application and generation", *Associative Networks*, Academic Press, 1979.

[McSk, Mk] McSkimin & Minker: "A predicate calculus based semantic network for deductive searching", *Associative Networks*, Academic Press, 1979.

ANNEX No 3

In this section, we present a few Prolog programs which achieve the generalized s-unification and word-to-word correspondence from English to French and from French to English in a specific domain (namely transportation matters).

This small prolog program achieves the "similarity unification" of semantic categories. The semantic network provides for a semantic description of human beings and more specifically of the attendance of a conference.

Since this has been elaborated in order to be used within the conference domain, several views of the attendance are considered.

```
% member(X,Y).
% X is a member of Y
member(X,[X|_]).
member(X,[_|Y]) :- member(X,Y).

% rep(X,Y).
% X represent the equivalence class Y
rep(universe,[universe]).
rep(animate,[animate]).
rep(inanimate,[inanimate]).
rep(human,[human,human1,human2]).
```

Two views of the semantic category "human" are envisionned.

```
rep(non-human,[non-human]).
rep(confmemb,[confmemb]).
rep(outconf,[outconf]).
rep(committee,[committee]).
rep(attendance,[attendance,attendance1,attendance2,attendance3]).
```

We consider here three different partitionning of the attendance of a conference.

```
rep(chairman,[chairman]).
rep(member,[member]).
rep(male,[male]).
rep(female,[female]).
rep(speaker,[speaker]).
rep(listener,[listener]).
rep(invitspk,[invitspk]).
rep(otherspk,[otherspk]).
rep(compstaff,[compstaff]).
rep(univstaff,[univstaff]).
rep(indepdt,[indepdt]).
rep(univstudt,[univstudt]).
rep(univmemb,[univmemb]).
rep(foreigner,[foreigner]).
rep(local,[local]).

% subset(X,Y).
% X is a subset of Y
```

This predicate defines the hierarchy among semantic categories of the semantic network.

```
subset(animate,universe).
subset(inanimate,universe).
subset(human,animate).
subset(non-human,animate).
subset(male,human1).
subset(female,human1).
```

The first "view" of human distinguishes between "male" and "female", whereas the second view distinguishes between "confmemb" (i.e. people attending the conference) and "outconf" (people who do not belong to the conference).

```
subset(confmemb,human2).
subset(outconf,human2).
subset(committee,confmemb).
subset(attendance,confmemb).
subset(chairman,committee).
subset(member,committee).
subset(speaker,attendance1).
subset(listener,attendance1).
subset(invitspk,speaker).
subset(otherspk,speaker).
```

The first view of the attendance of the conference focuses on the role played by the person attending the conference : speaker (split into invited speakers (invitspk) and others) or ordinary attendance (listener).

```
subset(compstaff,attendance2).
subset(univstaff,attendance2).
subset(indepdt,attendance2).
subset(univstudt,univstaff).
subset(univmemb,univmemb).
```

The second view of "attendance" discriminates people according to their working origin : company (compstaff), university (univstaff) or independent individuals (indepdt). Moreover, university staff is split in students and others (for fee distinction, for instance).

```
subset(foreigner,attendance3).
subset(local,attendance3).
```

The third partition of attendance discriminates people according to where they come from. Obviously, we have considered here three different partitions of the same set of people which is represented by the semantic category "attendance".

```
% eqsub(X,Y).
% sublink from X to Y
eqsub(X,Y) :- rep(Z,T), member(X,T), subset(Z,Y).
eqsub(X,Y) :- subset(X,Z), rep(U,T), member(Y,T), member(Z,T).
eqsub(X,Y) :- rep(U,T), member(X,T), subset(U,Z), eqsub(Z,Y).

% unify(X,Y,Z).
% Z unifies X and Y
unify(X,Y,Z) :- rep(Z,T), member(X,T), member(Y,T).
unify(X,Y,Z) :- eqsub(X,Y), rep(Z,T), member(Y,T).
unify(X,Y,Z) :- eqsub(Y,X), rep(Z,T), member(X,T).
unify(X,Y,Z) :- eqsub(X,U), eqsub(Y,V), unify(U,V,Z).
```

Thus, there exists a sublink from the semantic category X to the semantic category Y if, in the semantic network, we can link - using s-links and g-links but at least one g-link - the semantic category Y to X.

Then, given to semantic categories X and Y, "unify" provides for their "most general unifier", that is to say for the first semantic category in the network regrouping the semantic knowledge contained in X and Y.
For instance, if we want to unify the semantic categories "speaker" and "univstaff" we have to ``go upwards'' in the semantic network until we find a semantic category on which both of these semantic knowledges can be defined

(here we get "attendance" as a unifier).

```
% not(P)
% not(P) is false if P is true
not(P) :- P, !, fail.
not(_).
```

PROLOG APPLICATION No 1

This prolog program illustrates the use of similarity unification towards a better word-to-word correspondence between French and English.

The French and the English semantic networks considered here are *identical* and correspond to the semantic network described in Figure 1 in the third part of the report.

This prolog program illustrates the process of similarity unification when the English and French semantic networks are identical.

```
% member(X,Y).
% X is a member of Y
member(X,[X|_]).
member(X,[_|Y]) :- member(X,Y).


% rep(X,Y).
% the semantic category X represents the equivalence class of sem cat Y

rep(transpttion,[transpttion,transpttion1,transpttion2,transpttion3]).
rep(generalinfo,[generalinfo]).
rep(schedule,[schedule]).
rep(country,[country]).
rep(business,[business]).
rep(leisure,[leisure]).
rep(aireng,[aireng]).
rep(sea,[sea]).
rep(ground,[ground]).
rep(raileng,[raileng]).
rep(vehicle,[vehicle]).
rep(walkcat,[walkcat]).

rep(transport,[transport,transport1,transport2,transport3]).
rep(infgene,[infgene]).
rep(horairecat,[horairecat]).
rep(pays,[pays]).
rep(travail,[travail]).
rep(loisir,[loisir]).
rep(airfr,[airfr]).
rep(mer,[mer]).
rep(terre,[terre]).
rep(railfr,[railfr]).
rep(vehicule,[vehicule]).
rep(marcheapied,[marcheapied]).
```

The English and the French semantic networks provide fo several views of the semantic category "transportation" (called "transport" in the French network).
In both networks, the first view of "transportation" represents general information, the second view is related to the purpose of the trip and the third one distinguishes the different means of transportation.

```
% subset(X,Y).
% The semantic category X is a subset of the semantic category Y
subset(generalinfo,transpttion1).
subset(schedule,transpttion1).
subset(country,transpttion1).
subset(business,transpttion2).
subset(leisure,transpttion2).
subset(aireng,transpttion3).
subset(sea,transpttion3).
subset(ground,transpttion3).
subset(raileng,ground).
subset(vehicle,ground).
subset(walkcat,ground).
```

```
subset(infgene,transport1).
subset(horairecat,transport1).
subset(pays,transport1).
subset(travail,transport2).
subset(loisir,transport2).
subset(airfr,transport3).
subset(mer,transport3).
subset(terre,transport3).
subset(railfr,terre).
subset(vehicule,terre).
subset(marcheapied,terre).
```

```
% eqsub(X,Y).
% sublink from the semantic category X to the semantic category Y
eqsub(X,Y) :- rep(Z,T), member(X,T), subset(Z,Y).
eqsub(X,Y) :- subset(X,Z), rep(U,T), member(Y,T), member(Z,T).
eqsub(X,Y) :- rep(U,T), member(X,T), subset(U,Z), eqsub(Z,Y).
```

There exists a sublink from X to Y if, in the semantic network, we can find a path from the semantic category Y to the semantic category X involving at least one g-link and possibly some s-links.

```
% unify(X,Y,Z).
% the sem cat  Z unifies the sem cat  X and Y
unify(X,Y,Z) :- rep(Z,T), member(X,T), member(Y,T).
unify(X,Y,Z) :- eqsub(X,Y), rep(Z,T), member(Y,T).
unify(X,Y,Z) :- eqsub(Y,X), rep(Z,T), member(X,T).
unify(X,Y,Z) :- eqsub(X,U), eqsub(Y,V), unify(U,V,Z).
```

Given two semantic categories X and Y, "unify" provides for their "most general unifier", that is to say for the first semantic category in the network which regroups the semantic knowledge contained in X and Y.

The following predicate "id" identifies the semantic category X of the French network with the corresponding semantic category Y in the English network.

```
% id(X,Y).
% Y is the sem cat of the English network corresponding to the sem cat X in
the French network
id(transport,transpttion).
id(infgene,generalinfo).
id(horairecat,schedule).
id(pays,country).
id(travail,business).
id(loisir,leisure).
id(airfr,aireng).
id(mer,sea).
id(terre,ground).
id(railfr,raileng).
id(vehicule,vehicle).
id(marcheapied,walkcat).

id(X,Y) :- rep(U,T), member(X,T), rep(V,S), member(Y,S), id(U,V).
```

```
% freng(X,Y).
% Y is the English equivalent of the French word X
freng(arrivee,arrival).
freng(depart,departure).
freng(horaire,timetable).
freng(cout,cost).
freng(distance,distance).
freng(chemin,way).
freng(itineraire,route).
freng(voyage,travel).
freng(excursion,tour).
freng(etats-unis,united-states).
freng(japon,japan).
freng(france,france).
freng(outremer,overseas).
freng(avion,plane).
freng(vol,flight).
freng(aeroport,airport).
freng(bateau,boat).
freng(traversee,voyage).
freng(gare,station).
freng(train,train).
freng(metro,subway).
freng(auto,car).
freng(taxi,taxi).
freng(bus,bus).
freng(marche,walk).
```

```
% repfr(X,Y).
% X is an English word representing the set of English words Y each of them
corresponding to the same french general concept
repfr(timetable,[timetable,schedule]).
repfr(cost,[cost,price,charge,charges,expense,expenses,outlay,expenditure]).
repfr(distance,[distance,interval]).
repfr(way,[way,road,path,route]).
repfr(route-e,[route-e,route,itinerary]).
repfr(travel,[travel,trip,tour,journey,distance]).
repfr(tour-l,[tour-l,tour,trip,excursion,ramble]).
repfr(united-states,[united-states,united-states-of-america,usa,us,america]).
repfr(plane,[plane,aeroplane,aircraft]).
repfr(boat,[boat,ship]).
repfr(train,[train,railway-train]).
repfr(subway,[subway,underground,tube]).
repfr(car,[car,automobile]).
repfr(taxi,[taxi,taxi-cab,cab]).
repfr(bus,[bus,coach]).
repfr(walk,[walk,walking,stroll]).
```

The set of words Y above regroups English synonyms corresponding to the same general concept in French and the English word X is chosen to represent this concept. Hence, an English word, "route" for instance, may be interpreted in several different ways, i.e. either as a synonym of "way" or as a synonym of "route-e" (this notation was necessary since "route" can also be a French word), and will be translated differently depending on the context.

Conversely, as shown above, in the French semantic network, French words are regrouped according to the same English general concept and similarly a French representative for that concept is selected.

```
% repeng(X,Y).
% X is a French word representing the set of French words Y each of them
corresponding to the same english general concept
```

```
repeng(horaire,[horaire,indicateur,emploi-du-temps]).
repeng(cout,[cout,prix,frais,depense,debours]).
repeng(distance,[distance,eloignement]).
repeng(chemin,[chemin,route,trajet]).
repeng(itineraire,[itineraire,route,parcours]).
repeng(voyage,[voyage,trajet]).
repeng(excursion,[excursion,tour,voyage,randonnee,promenade]).
repeng(etats-unis,[etats-unis,etats-unis-d-amerique,usa,us,amerique]).
repeng(avion,[avion,aeroplane]).
repeng(bateau,[bateau,navire]).
repeng(traversee,[traversee,voyage-par-mer]).
repeng(auto,[auto,automobile,voiture]).
repeng(bus,[bus,autobus,car,autocar]).
repeng(marche,[marche,promenade-a-pied,tour]).

% wdcat(X,Y).
% X is a word of the semantic category Y

wdcat(cost,generalinfo).
wdcat(distance,generalinfo).
wdcat(way,generalinfo).
wdcat(route,generalinfo).

wdcat(arrival,schedule).
wdcat(departure,schedule).
wdcat(timetable,schedule).

wdcat(united-states,country).
wdcat(japan,country).
wdcat(france,country).
wdcat(overseas,country).

wdcat(travel,business).
wdcat(tour,leisure).

wdcat(plane,aireng).
wdcat(flight,aireng).
wdcat(airport,aireng).

wdcat(boat,sea).
wdcat(voyage,sea).

wdcat(station,raileng).
wdcat(train,raileng).
wdcat(subway,raileng).

wdcat(car,vehicle).
wdcat(taxi,vehicle).
wdcat(bus,vehicle).

wdcat(walk,walkcat).


wdcat(cout,infgene).
wdcat(distance,infgene).
wdcat(chemin,infgene).
wdcat(itineraire,infgene).

wdcat(arrivee,horairecat).
wdcat(depart,horairecat).
wdcat(horaire,horairecat).

wdcat(etats-unis,pays).
wdcat(japon,pays).
wdcat(france,pays).
wdcat(outremer,pays).
```

```
wdcat(voyage,travail).
wdcat(excursion,loisir).

wdcat(avion,airfr).
wdcat(vol,airfr).
wdcat(aeroport,airfr).

wdcat(bateau,mer).
wdcat(traversee,mer).

wdcat(gare,railfr).
wdcat(train,railfr).
wdcat(metro,railfr).

wdcat(auto,vehicule).
wdcat(taxi,vehicule).
wdcat(bus,vehicule).

wdcat(marche,marcheapied).
```

Here, every French and Enlish word previously selected as a representative of a general concept is related to a semantic category of the French and English networks respectively.

```
% eqveng(X,Y).
% Y is among the possible English equivalents for the French word X

eqveng(X,Y) :- repeng(Z,T), member(X,T), freng(Z,Y).

% fetrad(X,Y).
% Y is the English translation of the French word X
fetrad(X,Y) :- wdcat(X,S), eqveng(X,Y), freng(Z,Y), wdcat(Z,T), unify(S,T,S).
```

Whereas "eqveng" provides for all the possible translations of the French word X, "fetrad" gives only the English translations which are consistent with the semantic knowledge we have on X.
Conversely, "eqvfr" provides for all the possible French translations of the English word X and "eftrad" reduces that set to the English translations consistent with our semantic knowledge of X.

```
% eqvfr(X,Y).
% Y is among the possible French equivalents for the English word X

eqvfr(X,Y) :- repfr(Z,T), member(X,T), freng(Y,Z).

% eftrad(X,Y).
% X is the French translation of the English word Y
eftrad(X,Y) :- wdcat(X,S), eqvfr(X,Y), freng(Y,Z), wdcat(Z,U), unify(S,U,S).



% not(P)
% not(P) is false if P is true

not(P) :- P,!,fail.
not(_).
```

PROLOG APPLICATION No 2

This prolog program illustrates the use of similarity unification towards a better word-to-word correspondence between French and English.

The French and the English semantic networks considered here are *different* as it is described in Figure 1 and Figure 2 in the third part of the report.

This prolog programm illustrates the process of similarity unification when the English and French semantic networks are different.

```
% member(X,Y).
% X is a member of Y
member(X,[X|_]).
member(X,[_|Y]) :- member(X,Y).
```

```
% rep(X,Y).
% the semantic category X represents the equivalence class of sem cat Y

rep(transpttion,[transpttion,transpttion1,transpttion2,transpttion3]).
rep(generalinfo,[generalinfo]).
rep(schedule,[schedule]).
rep(country,[country]).
rep(business,[business]).
rep(leisure,[leisure]).
rep(aireng,[aireng]).
rep(sea,[sea]).
rep(ground,[ground]).
rep(raileng,[raileng]).
rep(vehicle,[vehicle]).
rep(walkcat,[walkcat]).

rep(transport,[transport,transport1,transport2,transport3]).
rep(infgene,[infgene]).
rep(voyetape,[voyetape]).
rep(voyglobal,[voyglobal]).
rep(horairecat,[horairecat]).
rep(pays,[pays]).
rep(travail,[travail]).
rep(loisir,[loisir]).
rep(airfr,[airfr]).
rep(mer,[mer]).
rep(terre,[terre]).
rep(railfr,[railfr]).
rep(vehicule,[vehicule]).
rep(marcheapied,[marcheapied]).
```

The English and the French semantic networks provide for several views of the semantic category "transportation" (called "transport" in the French network).
In both networks, the first view represents general information, the second view is related to the purpose and the third one distinguishes the different means of transportation.

```
% subset(X,Y).
% The semantic category X is a subset of the semantic category Y
subset(generalinfo,transpttion1).
subset(schedule,transpttion1).
subset(country,transpttion1).
subset(business,transpttion2).
subset(leisure,transpttion2).
subset(aireng,transpttion3).
subset(sea,transpttion3).
subset(ground,transpttion3).
```

```
subset(raileng,ground).
subset(vehicle,ground).
subset(walkcat,ground).

subset(infgene,transport1).
subset(voyetape,infgene).
subset(voyglobal,infgene).
subset(horairecat,transport1).
subset(pays,transport1).
subset(travail,transport2).
subset(loisir,transport2).
subset(airfr,transport3).
subset(mer,transport3).
subset(terre,transport3).
subset(railfr,terre).
subset(vehicule,terre).
subset(marcheapied,terre).
```

In that case, the two semantic networks are slightly different since in the French network the semantic category "infgene" - corresponding to the semantic category "generalinfo" in the English network - is subdivised into two semantic categories "voyetape" and "voyglobal" distinguishing whether we consider a trip as a whole or in details (considering the route and transfer schedule for instance).

```
% eqsub(X,Y).
% sublink from the semantic category X to the semantic category Y
eqsub(X,Y) :- rep(Z,T), member(X,T), subset(Z,Y).
eqsub(X,Y) :- subset(X,Z), rep(U,T), member(Y,T), member(Z,T).
eqsub(X,Y) :- rep(U,T), member(X,T), subset(U,Z), eqsub(Z,Y).
```

There exists a sublink from X to Y if, in the semantic network, we can find a path from the semantic category Y to the semantic category X involving at least one g-link and possibly some s-links.

```
% unify(X,Y,Z).
% the sem cat  Z unifies the sem cat  X and Y
unify(X,Y,Z) :- rep(Z,T), member(X,T), member(Y,T).
unify(X,Y,Z) :- eqsub(X,Y), rep(Z,T), member(Y,T).
unify(X,Y,Z) :- eqsub(Y,X), rep(Z,T), member(X,T).
unify(X,Y,Z) :- eqsub(X,U), eqsub(Y,V), unify(U,V,Z).
```

Given two semantic categories X and Y, "unify" provides for their "most general unifier", that is to say for the first semantic category in the network which regroups the semantic knowledge contained in X and Y.

The following predicate "id" identifies the semantic category X of the French network with the corresponding semantic category Y in the English network, whenever it is possible.

```
% id(X,Y).
% Y is the sem cat of the English network corresponding to the sem cat X in
the French network
id(transport,transpttion).
id(infgene,generalinfo).
```

```
id(horairecat,schedule).
id(pays,country).
id(travail,business).
id(loisir,leisure).
id(airfr,aireng).
id(mer,sea).
id(terre,ground).
id(railfr,raileng).
id(vehicule,vehicle).
id(marcheapied,walkcat).



% freng(X,Y).
% Y is the English equivalent of the French word X
freng(arrivee,arrival).
freng(depart,departure).
freng(horaire,timetable).
freng(cout,cost).
freng(distance,distance).
freng(chemin,way).
freng(itineraire,route-e).
freng(voyage,travel).
freng(excursion,tour-l).
freng(etats-unis,united-states).
freng(japon,japan).
freng(france,france).
freng(outremer,overseas).
freng(avion,plane).
freng(vol,flight).
freng(aeroport,airport).
freng(bateau,boat).
freng(traversee,voyage).
freng(gare,station).
freng(train,train).
freng(metro,subway).
freng(auto,car).
freng(taxi,taxi).
freng(bus,bus).
freng(marche,walk).


% repfr(X,Y).
% X is an English word representing the set of English words Y each of them
corresponding to the same french general concept
repfr(timetable,[timetable,schedule]).
repfr(cost,[cost,price,charge,charges,expense,expenses,outlay,expenditure]).
repfr(distance,[distance,interval]).
repfr(way,[way,road,path,route]).
repfr(route-e,[route-e,route,itinerary]).
repfr(travel,[travel,trip,tour,journey,distance]).
repfr(tour-l,[tour-l,tour,trip,excursion,ramble]).
repfr(united-states,[united-states,united-states-of-america,usa,us,america]).
repfr(plane,[plane,aeroplane,aircraft]).
repfr(boat,[boat,ship]).
repfr(train,[train,railway-train]).
repfr(subway,[subway,underground,tube]).
repfr(car,[car,automobile]).
repfr(taxi,[taxi,taxi-cab,cab]).
repfr(bus,[bus,coach]).
repfr(walk,[walk,walking,stroll]).
```

The set of words Y above regroups English synonyms corresponding to the same general concept in French and the English word X is chosen to represent this concept. Hence, an English word, "route" for instance, may be interpreted in several different ways, i.e. either as a synonym of "way" or as a synonym of "route-e" (this notation was necessary since "route" can also be a French word), and will be translated differently depending on the context.

Conversely, as shown above, in the French semantic network, French words are regrouped according to the same English general concept and similarly a French representative for that concept is selected.

```
% repeng(X,Y).
% X is a French word representing the set of French words Y each of them
corresponding to the same english general concept
repeng(horaire,[horaire,indicateur,emploi-du-temps]).
repeng(cout,[cout,prix,frais,depense,debours]).
repeng(distance,[distance,eloignement]).
repeng(chemin,[chemin,route-f,trajet]).
repeng(itineraire,[itineraire,route-f,parcours]).
repeng(voyage,[voyage,trajet]).
repeng(excursion,[excursion,tour-f,voyage,randonnee,promenade]).
repeng(etats-unis,[etats-unis,etats-unis-d-amerique,usa,us,amerique]).
repeng(avion,[avion,aeroplane]).
repeng(bateau,[bateau,navire]).
repeng(traversee,[traversee,voyage-par-mer]).
repeng(auto,[auto,automobile,voiture]).
repeng(bus,[bus,autobus,car,autocar]).
repeng(marche,[marche,promenade-a-pied,tour-f]).

% wdcat(X,Y).
% X is a word of the semantic category Y

wdcat(cost,generalinfo).
wdcat(distance,generalinfo).
wdcat(way,generalinfo).
wdcat(route-e,generalinfo).
wdcat(route,generalinfo).

wdcat(arrival,schedule).
wdcat(departure,schedule).
wdcat(timetable,schedule).

wdcat(united-states,country).
wdcat(japan,country).
wdcat(france,country).
wdcat(overseas,country).

wdcat(travel,business).
wdcat(tour-l,leisure).
wdcat(tour,leisure).

wdcat(plane,aireng).
wdcat(flight,aireng).
wdcat(airport,aireng).

wdcat(boat,sea).
wdcat(voyage,sea).

wdcat(station,raileng).
wdcat(train,raileng).
wdcat(subway,raileng).

wdcat(car,vehicle).
wdcat(taxi,vehicle).
wdcat(bus,vehicle).

wdcat(walk,walkcat).
```

```
wdcat(cout,infgene).
wdcat(distance,infgene).
wdcat(trajet,infgene).

wdcat(itineraire,voyetape).
wdcat(route-f,voyetape).

wdcat(chemin,voyglobal).

wdcat(arrivee,horairecat).
wdcat(depart,horairecat).
wdcat(horaire,horairecat).

wdcat(etats-unis,pays).
wdcat(japon,pays).
wdcat(france,pays).
wdcat(outremer,pays).

wdcat(voyage,travail).
wdcat(excursion,loisir).

wdcat(avion,airfr).
wdcat(vol,airfr).
wdcat(aeroport,airfr).

wdcat(bateau,mer).
wdcat(traversee,mer).

wdcat(gare,railfr).
wdcat(train,railfr).
wdcat(metro,railfr).

wdcat(auto,vehicule).
wdcat(taxi,vehicule).
wdcat(bus,vehicule).

wdcat(marche,marcheapied).
```

Here, every French and English word previously selected as a representative of a general concept is related to a semantic category of the French and English networks respectively.

```
% eqveng(X,Y).
% Y is among the possible English equivalents for the French word X

eqveng(X,Y) :- repeng(Z,T), member(X,T), freng(Z,Y).

% fetrad(X,Y).
% Y is the English translation of the French word X
fetrad(X,Y) :- wdcat(X,S), eqveng(X,Y), freng(Z,Y), wdcat(Z,T), unify(S,T,S).
```

Whereas "eqveng" provides for all the possible translations of the French word X, "fetrad" gives only the English translations which are consistent with the semantic knowledge we have on X.

Conversely, "eqvfr" provides for all the possible French translations of the English word X and "eftrad" reduces that set to the English translations consistent with our semantic knowledge of X.

```
% eqvfr(X,Y).
% Y is among the possible French equivalents for the English word X

eqvfr(X,Y) :- repfr(Z,T), member(X,T), freng(Y,Z).

% eftrad(X,Y).
% X is the French translation of the English word Y
eftrad(X,Y) :- wdcat(X,S), eqvfr(X,Y), freng(Y,Z), wdcat(Z,U), unify(S,U,S).


% not(P)
% not(P) is false if P is true

not(P) :- P,!,fail.
not(_).
```

In that program, in order to test fetrad(X,Y) [resp. eftrad(X,Y)] which provides for the semantically correct English (resp. French) translation of the French (resp. English) word X, we have introduced the extra semantic knowledge :

```
wdcat ( route, generalinfo )

wdcat ( tour, leisure )

wdcat ( trajet, infgene )

wdcat ( route-f, voyetape )
```

N.B. : Here, "route" is an English word whereas "route-f" denotes a French word.

With that extra semantic knowledge, we obtain the following results :

```
eqvfr ( route, Y )        ------->    chemin ;

                                      itineraire.

eftrad ( route, Y )       ------->    chemin ;

                                      itineraire.

eqvfr ( tour, Y )         ------->    voyage ;

                                      excursion.

eftrad ( tour, Y )        ------->    excursion.


eqveng ( trajet, Y )      ------->    way ;

                                      travel.

fetrad ( trajet, Y )      ------->    way.

eqveng ( route-f, Y )     ------->    way ;

                                      route-e.

fetrad ( route-f, Y )     ------->    route-e.
```

Thus, for instance, since "route-f" belongs to the semantic category "voyetape", it is viewed as a synonym of "itineraire" and we obtain the unique English equivalent "route-e".

Hence, in that case, the semantic knowledge is helpful for disambiguating the translation process.

On the other hand, since no partition in the English network correspond to the partition (voyetape, voyglobal) in the French network, for the English word "route" none of its synonyms "way" and "route-e" can be discarded and the semantic knowledge doesn't help to discard any of the possible French translations "chemin" and "itineraire".