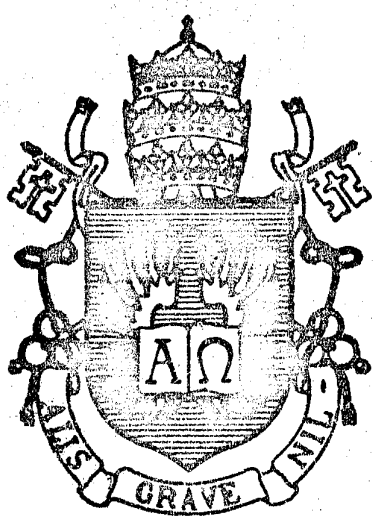



# PUC

Series: Monografias em Ciência da Computação
Nº10/88

# DEEP GENERATION IN A CRIME KNOWLEDGE-BASED SYSTEM

Maria das Graças Volpe Nunes

Departamento de Informática

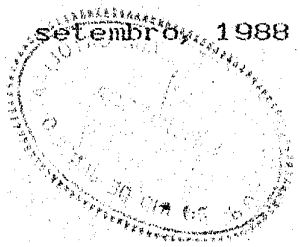
PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
MARQUÊS DE SÃO VICENTE, 225 – CEP 22453
RIO DE JANEIRO – BRASIL

PUC/RJ - DEPARTAMENTO DE INFORMÁTICA

Séries : Monografias em Ciência da Computação, no.10/88

Editor : Paulo A. S. Veloso

setembro, 1988



# Deep Generation in a Crime Knowledge-Based System *

*Maria das Graças Volpe Nunes*

* Presented by prof. Donia Scot, visiting professor from the University of Sussex, Sussex, England.

Responsável por publicações:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC/RJ-Depto. de Informática
Rua Marquês de São Vicente, 225 - Gávea
22453 - Rio de Janeiro, RJ
BRASIL

ABSTRACT : A deep generation technique is proposed for producing cooperative replies to Yes/No questions. It is based on the linguistic notion of focus and involves the use of common-sense knowledge about the domain of discourse. Information selected or derived from domain knowledge constitute the response and form the terminal elements of a discourse structure that is based on Rhetorical Structure Theory [Mann & Thompson-87].

KEY-WORDS : text generation, cooperativity, common-sense knowledge, relevance, semantic focus.

RESUMO : É proposta uma técnica de geração de respostas cooperativas a perguntas do tipo Sim/Não. Esta técnica é baseada na noção linguística de foco e involve o uso de conhecimento de senso comum sobre o domínio do discurso. A resposta consiste de um conjunto de informações selecionadas ou derivadas do conhecimento do domínio. As informações deste conjunto constituem os elementos terminais de uma estrutura de discurso baseada na teoria Rhetorical Structure [Mann & Thompson-87].

PALAVRAS-CHAVES : geração do texto, cooperação, conhecimento de senso comum, relevância, foco semântico.

CONTENTS

# 1. INTRODUCTION

## 1.1. TEXT GENERATION IN KNOWLEDGE-BASED SYSTEMS

For a variety of systems such as expert systems, database systems and problem-solving systems, Natural Language Interfaces are being introduced as a way of improving the communication with users. The attraction of natural language arises from the fact that the increasing levels of complexity of tasks and range of expertise of users put heavy demands on systems that they should be able to support "intelligent" and "natural" dialogue with users.

In knowledge-based systems, such as those to be discussed here, a natural language interface allows the user to retrieve information from several sources of knowledge - knowledge base - by asking questions and receiving answers in, for example, English or Portuguese.

As pointed out by McKeown [Mckeown-85], research in response generation has addressed the formulation of strategies that can be used to organize and determine content of the response and the problem of producing responses that cooperatively address the questioner's intentions.

We follow the maxims of Grice's cooperative principle [Grice-75] :
(1) Quantity : (1a) Make your contribution as informative as is required; (1b) Don't make it more informative than necessary;
(2) Quality : (2a) Try to make your contribution one that is true; (2b) Do not say what you believe to be false; (2c) Do not say that for which you lack adequate evidence;
(3) Relatedness : Be relevant;
(4) Modality : Be perspicuous; (4a) Avoid obscurity of expression; (4b) Avoid ambiguity; (4c) Be brief (avoid unnecessary prolixity); (4d) Be orderly.

Some other norms, ethic, aesthetic and social, are also considered. Of course, there are cases where it would not be suitable to obey all these aspects simultaneously.

If the text is not stored ahead of time for the system to retrieve it when needed - canned texts or templates - the text generator module is expected to have the following features [McDonald-85] :

- It must be able to decide **what to say** : filter out information in its knowledge bases that can be ignored and pinpoint information that should be included. One technique for determining what to say is to use different discourse strategies for different purposes. Information about the current user's beliefs ( User Model ) also influences what the system says in order to make communication succesful , in particular, to discover sources of user misconceptions and to tailor explanations to user's level of knowledge.
- It must be able to decide **when to say what** : the order of a text can be crucial to a reader's understanding of it. Discourse strucuture is currently used to help determine the order of presentation in several text generation systems.
- It must be able to determine what the surface text should

look like - **how to say it** : this involves making decisions about what vocabulary to use, when to use a pronoun or to use a full noun phrase to refer to an object or concept, whether to use a sequence of simple sentences or to combine several simple sentences into a single complex sentence, and how to arrange the words in each sentence. Almost all these decisions are influenced by syntactic constraints on language; thus we need a grammar. Information about user type can be used to select appropriate vocabulary. Knowledge about how a given sentence fits in with the rest of the text( its function ) can be used to choose the syntactic structure for a sentence and to decide whether to use pronouns.

Text generation can be divided in two distinct components :

a) A **Deep generator** [MacKeown-82] ( also referred to as a strategic component [Appelt-85] or text planner component [McDonald-85] ) :

This is devoted to planning communication. It involves to decide **what to say** and **when to say what**. Information must be organized for a best textual presentation. This entails adding rhetorical specifications, determining the sequence in which its information will be presented, and making some of the decisions about what words will be used. It produces an intermediary form of the output text.

b) A **Surface generator** [McKeown-82] ( or tactical component [Appelt-85], linguistic component [McDonald-85] ) :

This is devoted to grarmatical realization of the output of the deep generator. It is here that decisions about **how to say it** are made. The contributions of this component to the text involve the following categories of linguistic decisions and background knowledge, as outlined in [McDonald-85]:

- Choice of vocabularylary
- Prose style - context independent rules
- Syntactic and Morphological details
- Grammatically-forced restructuring of information
- Manifesting the speaker's intent
- Maintaining cohesion in a discourse

This distinction between deep and surface generation is atrractive because it provides at least two levels of abstraction; details that are relevant to the surface level can be ignored at the deep level. For example, the decision on which determiner to use in a clause is not an appropriate consideration at the time of devising a strategy to convince the user of some proposition. Moreover, if the interface between these two levels were carefully specified, it would appear possible to design a general tactical component that could be used for a variety of applications.

As we can see, decisions that must be made by a text generation system range over a variety of knowledge sources and are influenced by a variety of factors.

Here, we will be concern ourselves with discussing the deep generator of a natural language interface for a simple knowledge-based system about crimes.

A second step of this project is to build the corresponding surface generator.

## 1.2. THE PRESENT WORK

This work presents a deep generation component that has been developed for a question-answering system. The deep generator is intended to form part of a natural language interface which includes an input interpreter, dialogue, discourse and user modellers and a surface generator for producing textual responses.

The input to the deep generator is a semantic representation of user queries. Its output is structure set of information selected or derived from domain knowledge bases. The structure of the output indicates the relations that hold between the facts contained in the structure, and their predicted effect on the user's beliefs. In effect, the output of the deep generator is a language-independent semantic representation of a system response. Our eventual aim is that it should serve as input to a surface generator for produging portuguese text and to a user modeller for updating a user model when the textual response has been presented.

With few exceptions, questions are requests for information that is unknown to the asker, or of which he or she is unsure. From the point of view of response generation, the most significant difference between Wh- and Yes/No-questions is that where Wh-questions provide the hearer with explicit signals of what aspect(s) of the question the asker knows (or thinks) he doesn't know, Yes/No questions are rather more obscure. Another important distinction is that it is generally the case that Yes/No-questions expose more of the questioner's presuppositions than do Wh-question. For these reasons, we are primarily concerned with Yes/No-questions.

In cases where a Yes/No question prompts a negative reply, the semantic treatment of responses that conform to the Gricean maxims involves determining precisely which aspect of the propositional content of the query is, as far as the questioner is concerned, most unknown. This information forms the semantic focus of the question.

Semantic focus can be signalled directly by questioners (via syntactic choice, lexical choice or, in speech, accent placement) or can be derived from the structure of the discourse in which the question appears. In the work we present, neither source is avaiable to us for Yes/No questions. Instead, we determine their focus via the application of a set of domain-dependent default focus rules. Such rules would be required in any natural language system.

Our deep generator has been implemented in the context of a system which contains knowledge about crimes, and which responds to queries about known crimes and about a subset of the brazilian penal code. The system contains a USER MODEL and three domain-related knowledge bases :

- The LEGAL KNOWLEDGE BASE, comprised of the legal definitions of crimes, ther applicability conditions and

- The CRIME KNOWLEDGE BASE, containing facts about particular crimes.
- The COMMON SENSE KNOWLEDGE BASE, containing common-sense rules about the world of crimes. This we have found to be especially useful in the treatment of misconceptions.

The system generates the intermediary form of a textual response, which is a functional structure of the information selected or derived from the knowledge bases and the user model. The structure generated is a set of relations that link the facts chosen for the response. These relations are based on Rhetorical Structure Theory [Mann & Thompson-86a]. The structure is, in essence, the discourse structure of a response that is cohesive, coherent [Hobbs-79] and cooperative [Grice-75].
From these perspectives, our intention is to

a) investigate the role of the above mentioned knowledge sources in cooperative responses
b) implement a mechanism for detecting user misconceptions and for justifying information included in such responses that contradict the user's beliefs
c) test the applicability of Rhetorical Structure Theory as generation theory.

This present work addresses the first two issues. The third will only be achieved when we have completed our construction of the surface generator.

In section 2. we discuss related work on the treatment of user misconception and on Rhetorical Structure Theory. We present our system and examples in section 3. and conclusions in section 4. Several useful papers not mentioned in the text are also listed in References.

## 2. RELATED WORK

### 2.1. THE TREATMENT OF USER MISCONCEPTIONS

A cooperative answer is one which not only corrects the user's mistaken beliefs but also addresses the missing or mistaken user beliefs that led to it.

Misconceptions can fall into one of a number of categories, depending on the domain and the types of user questions the system handles.

In [Quilici-87], the advice-seeking dialogues with novice UNIX users leads to the detection of plan-oriented misconceptions. Upon detection a misconception, the system shows a set of advisor beliefs that together contradict the user's belief, which corresponds to an explanation. The search for an explanation makes use of both domain-specific advisor beliefs and a taxonomy of

domain-independent explanations for potential user misconceptions. In essence, the model uses information about likely sources of different classes of user misconceptions to recognize user mistakes and infer their underlying causes.

McCoy's [McCoy-87] model for detecting and explaining user misconceptions [McCoy-87] is strictly related to a "highlighted" model of the user, which allows the identification of possible sources of the error. Her work concentrates on object-related misconceptions; thus the system's model of the world contains an object taxonomy with attribute/value pairs attached to the objects.

As McCoy points out, a human response to a misconception can be viewed as consisting of three parts : (1) a denial of the incorrect information, (2) a statement of the correct information, and (3) justification for the denial and correction given. Further, the justification often seems to refute support that might have led to the misconception.

If we could characterize all support/justification pairs in this kind of domain independent way, a computer system could appropriately respond to entire classes of misconceptions. But the support someone might have for a misconception is extremely varied. However, it is possible in an object taxonomy domain that for a given type of misconception only a few kinds of support are necessary for in the justification part of the response.

We see that both models handle classes of misconceptions which lead to justification procedures based on the type of domain. Another strong influence is the user model the system has; the more sophisticated and complete, the more correct the detection of user misconceptions.

In our system we do not classify misconceptions but we try to follow the human model of responses to misconceptions - denial, correction, justification. The user model we have is very simple but we rely on commom sense rules about the world of crime that we expect can contribute to a reasonable answer to questions involving misconceptions. Details of our treatment of misconceptions are given in section 3.5.

## 2.2. GENERATION TECHNIQUES

An organized text is one which is composed of discernible parts arranged in a particular way and conected to form a whole.

Rhetorical Structure Theory [Mann & Thompson-86a] is a descriptive theory of a major aspect of the organization of natural text. It is a linguistically useful method for describing natural texts, characterizing their structure primarily in terms of relations that hold between parts of the text and how they relate to the generation goal.

As a descriptive framework for text, this theory provides several useful features for discourse studies : it identifies hierarchic structure in text ; it describes the relations between text parts in functional terms, identifying both the transition

point of a relation and the extent of the items related; it provides a comprehensive analyses rather than selective commentary and it is insensitive to text size.

Since very litle has been written about RST as a generation framework for text, the authors idea is to generate a text from a goal to be reached by the text by means of successive selections of schemas to form a deep structure of the text.

In our case, having selected the information to be part of the answer, we choose a subgoal of the top level answer goal. So we are able to select an appropriate schema which has a particular role in a more external schema. Each piece of information will be a text unit. In RST this quantification is flexible.

According to RST, texts are composed of instances of schemas that take the following form :

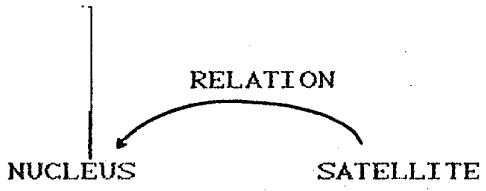



**Fig. 1. The strucutre of RST relations**

The vertical line points to one of the text spans which the schema covers, called the nucleus. The other spans are linked to the nucleus by relations ( usually one ), represented by labelled arcs; these spans are called satellites.The relations relate the conceptual span of a nucleus to the conceptual span of a satellite. The differences between nucleus and satellite will become clearer after some considerations about the members of the pair involved in a relation :

1. Often, the contribuition of the satellite to the text is incomprehensible independent of the nucleus, but not vice versa.
2. Often, the satellite is more suitable for substitution than nucleus.
3. Often, the nucleus is more essential to the writer's purpose than the satellite. People often strongly agree that a text with a particular satellite deleted would be more satisfatory (to the writer, as a substitute text) than a text with corresponding nucleus deleted.

A schema is composed of one or more relations which may be further decomposed into other relations. In text that is cohesive and coherent there is usually only one relation in the top-level schema.

An RST analysis of a text consists of a set of schema applications which collectively decompose the text into either terminal units or spans further decomposed in the analysis.

In a second level of decomposition, both the nucleus and satellites are text spans that can be instantiated as unitary

clauses or recursively as schemas.

The conventions for schema application are :

a) A schema is instantiated to describe the entire text.
b) Schemas are instantiated to describe the text spans produced in the instantiation of other schemas.
c) The schema does not constrain the order of nucleus and satellites in text span where it is instantiated.
d) All satellites are optional.
e) At least one satellite must occur.
f) A relation that is part of a schema may be instantiated indefinitely in the instantiation of that schema.
g) The nucleus and the satellites do not necessaraly correspond to a single uninterrupted text span.

There are, however, strong patterns in the use of schemas : relations tend to be used only one at a time, nucleus and satellites tend to occur in a certain order, and schemas tend to occur used in uninterrupted text spans (ie. they tend not to overlap.

The heart of RST is the relation definitions.

A relation definition consists of four fields :

1. Constraints on the Nucleus
2. Constraints on the Satellite
3. Constraints on the Combination of Nucleus and Satellite
4. The Effect

As an example, we present the values of the above fields in the definition of the Evidence relation. This relation refers to a situation where one piece of information provides evidence for another. Here the former (the evidence) is the satellite and the latter (the claim) is the nucleus. This is shown in figure 2.

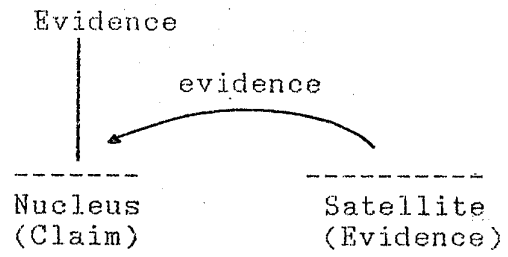



**Fig.2. The Evidence Relation**

The constraints on the fields of this relation are :

1. Constraints on the Nucleus ( the claim ) :
   The reader possibly does not already believe the claim.
2. Constraints on the Satellite ( the evidence )
   a. The satellite presents knowledge, (in contrast to, for example, advocating action or seeking approval. )

b. The reader believes the satellite or will find it credible.

3. Constraints on the combination of Nucleus and Satellite : Comprehending the evidence will increase the reader's belief in the claim.
4. The Effect : The reader's belief in the claim is increased.

Schemas define the structural constituency arrangements of text. They are abstract patterns consisting of a small number of constituent text spans, a specification of the relations between them, and a specification of how certain spans (nuclei) are related to the whole collection. They are thus loosely analogous to grammar rules.

The following example, extracted from [Mann & Thompson-87], shows another possible relations a text may involve :

1. Farmington police had to help control traffic recently
2. when hundreds of people lined up to be among the first applying for jobs at the yet-to-open Marriot Hotel.
3. The hotel's help-wanted announcement - for 300 openings - was a rare opportunity for many unemployed.
4. The people waiting in line carried a message, a refutation, of claims that the jobless could be employed if only they showed enough moxie.
5. Every rule has exceptions,
6. but the tragic and too-common tableaux of hundreds or even thousands of people snake-lining up for any task with a paycheck illustrates a lack of jobs,
7. not laziness.

Figure 3 gives the RST diagram for this excerpt.

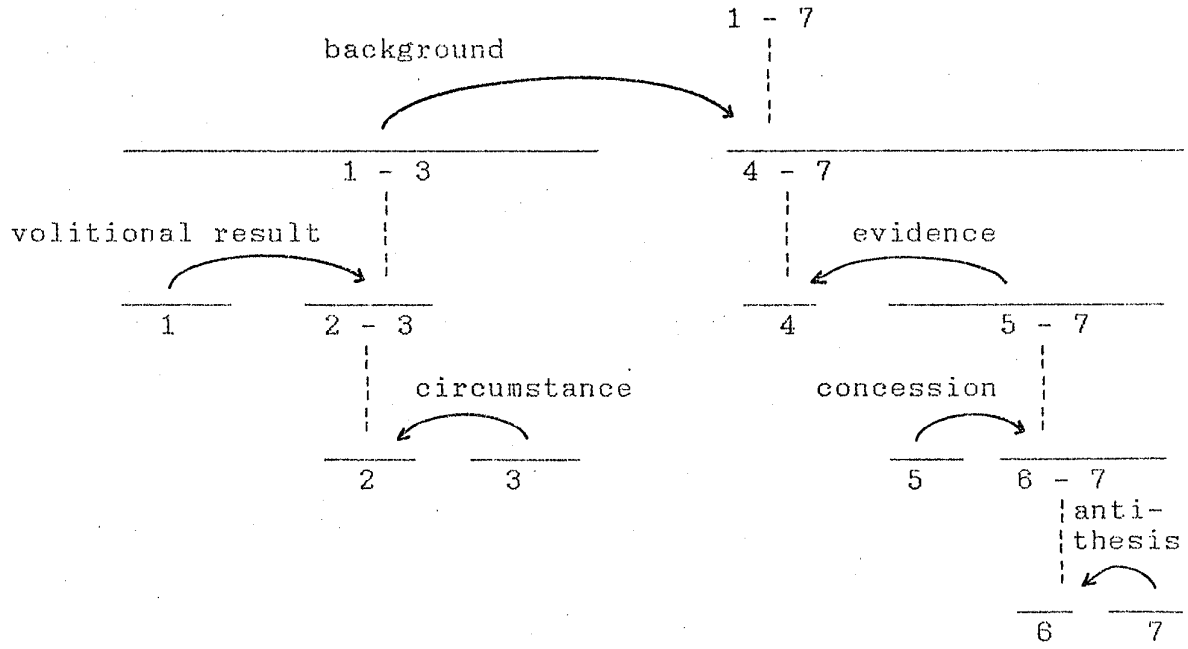



Fig.3. RST structure for the Marriot Hotel example

Another attempt to describe text structure in a way that is sufficiently detailed and general to serve as a basis for programming, is the TEXT system [McKeown-82], whose general task is to explain the structure and terminology, but not the content, of a particular data base.

In what follows, we describe some aspects of the TEXT system in order to justify our choice on RST as our generation technique. The comparative observations were extracted from [Mann-87].

TEXT has two kinds of defined objects : predicate semantics and schemas.

The predicate semantics definitions are essentially patterns which can be matched in the data base or knowledge base. They guide the system's search for particular knowledge. The schemas of TEXT are specifications of how expressions of predicates can be combined to form whole texts. The schemas are abstracted from previous texts which are judged to be doing the same task; the high frequency patterns are identified and represented in the schemas.

Three features of the TEXT schemas are particularly significant :

1. The scale of schemas is the whole text. Texts are specified as whole units, whose structural possibilities are prespecified rather than created for the individual instance.
2. Optionality of elements is built in at the whole-text level.
3. Order of elements is prespecified.

The most significant differences between the text structure accounts of TEXT and RST are :

- Size of the largest structural unit - whole text vs. RST schema.
- Dual use of TEXT schemas for determining what constitutes an adequate answer, and the text structure itself, versus separate uses in RST.
- Absence of relations in TEXT.
- Explicit nuclearity in RST.
- Fixed ordering in TEXT versus free order in RST.
- Recording of the predicted effects of text on the reader's beliefs in RST, but not in TEXT.

Since TEXT relies on predetermined text structures based on study of prior texts, it does not need to reason extensively about the need for particular text elements, nor about how to strucutre them (it does not handle relations). Its schemas encode both the task and the method with the major advantages of being implementable and effective in its designed domain (domain-dependent).

On the other hand, RST separates text structure building from other processes, which has the obvious advantages of flexibility and greater generality, but produces a corresponding need to implement and coordinate a diversity of processes.

Since the RST approach creates structure based on the immediate needs of the text under construction (depending on the relations), rather than past texts, it must reason much more about the reader's state (beliefs in User Model) and the way in which

the reader's state (beliefs in User Model) and the way in which relations affect that state. This is particularly useful for dialogue systems, where the need to maintain an accurate model of the user, and therefore to determine the effect of a response on the user's beliefs, is great.

Finally, RST provides a framework for investigating Relational Propositions, which are unstated but infered propositions that arise from the text structure in the process of interpreting texts (see [Mann & Thompson-86b]). Since the coherence of a text depends in part on these Relational Propositions, RST has been useful in the study of text coherence.

Details of the RST-based ideas used in our project are in section 3.7.

# 3. THE SYSTEM

## 3.1. GENERAL CONSIDERATIONS

The system has as input a Natural Language Yes/No question that is parsed ( in a phase not constructed but assumed by us ) into a system query to determine an event. In the domain we have chosen, these events relate to crimes.

A criminal event is determined by the following facts that we refer to as event information and that are most central to a criminal event :

*Autor* - the agent of the crime.
*Ação* - the action of the crime ( robbery, homicide, attempt homicide, kidnapping, rape, etc )
*Vítima* - the victim
*Local* - the locality of the crime
*Dia* - the day that it happened

The minimal sets of information that determine an event, in Yes/No questions, are : the agent or the victim or the locality and the day. Only the action does not determine an event.

Information other than that of the event may also be involved in the user's question. These are described in section 3.4.

The system tries to prove the input query ( see figure 4 ) using the Crime Knowledge Base and Legal Knowledge Base. Depending on it is success or failure, the deep strucutre of a Yes or a No answer is constructed. The format of this deep structure is a rhetorical structure : a functional definition of the text ( answer ). This structure is a relation between a nucleus and a satellite so that nucleus and satellite are recursively rhetorical structures.

We will not be controlling the *context*. Each question is analyzed independently.

The system is implemented in Arity Prolog, version 4.0.

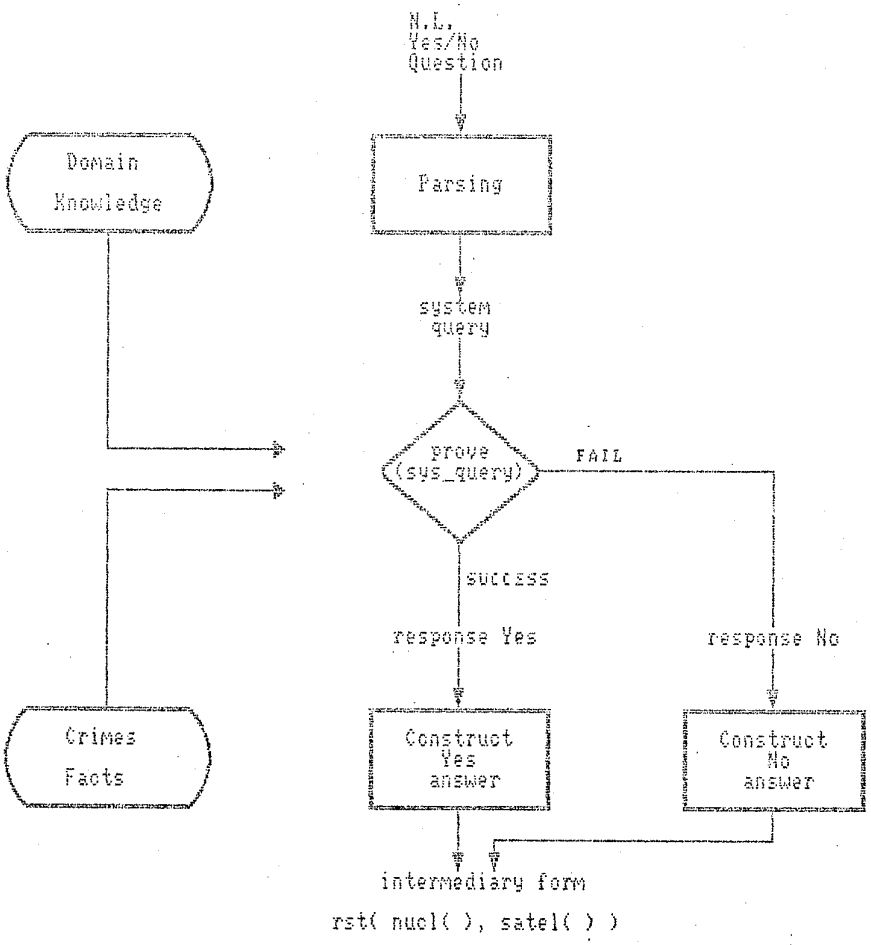



Fig.4. The Method of processing Yes/No Questions

## 3.2. KNOWLEDGE REPRESENTATION

We will be considering four sources of knowledge that affect the construction of the answer to the user :

**1. Crime Knowledge Base** : This module contains all the crime facts the system knows. Each crime event is represented by

*evento( ident, autor( ), ação( ), vítima( ), local( ), dia( ) )*

Extra-event information involved in questions are (and represented as Prolog facts ) :

```
Instrument - instrumento(Event, Inst)
Motive - motivo(Event, Mot)
Agent Situation - situação_autor(Event, preso),
                  situação_autor(Event, foragido)
Victim Situation - situação_vítima(Event, ferida),
                   situação_vítima(Event, morta)
Victim Type - tipo_vítima(Event, humano),
              tipo_vítima(Event, feto),
              tipo_vítima(Event, recem_nascido)
Victim-Agent Relation - rel_autor_vítima( Event, marido ),
                        rel_autor_vítima( Event, esposa ),
                        rel_autor_vítima( Event, amante ),..
Crime Type (Legal classification) - tipo_crime(Event, Tipo )
Penalty - pena(Tipo_crime,reclusão(Pena_mínima,Pena_máxima))
          pena(Tipo_crime,detenção(Pena_mínima,Pena_máxima))
Role - the role of a person in a crime.
       arrolado( Person, vítima, Event ),
       arrolado( Person, testemunha, Event ), etc.
```

**2. Legal Knowledge Base** : This module is a Knowledge Base of information extracted from the Brazilian Penal Code - definitions of crimes and their respective punishments. At present, these definitions permit us to classify a criminal event as a suicide, induced suicide, one of several kinds of homicide, attempt homicide, one of several kinds of abortion, or infanticide.

This kind of knowledge also allows us to establish differences and similarities between crimes from features of each kind of crime.

The definitions follow the model :

```
tipo_crime( Ev, suicidio ) :-
  tipo_vítima( Ev, ser_humano ),
  arrolado( Person, vítima, Ev ),
  arrolado( Person, autor, Ev ).
```

**3. Commom Sense Knowledge Base** : This module contains rules that enable the system to justify the user's misconceptions related to crime world. The semantics of these rules are :

If *potencial( X, Event, Y )* is deduced
then the information used for that deduction constitute a

plausible source for the fact that user thinks that Y is X.

The information used for the above deductions come from Crime Knowledge Base. Even if the user does not know some facts that make a misconception plausible, the respective rule is shown to him or her. The rules are applied to facts about the event, the persons involved and the world.

The rules used at present are :

```
%  Rules about potential suspect

% Someone who was present at the crime.

potencial( suspeito, Ev, Pessoa ) :-
     evento( Ev,_,_,_,local(Local), dia(Dia) ),
     estava( Pessoa, Local, Dia ).

% Someone with a motive for the crime.

potencial( suspeito, Ev, Pessoa ) :-
     tem_motivo( Pessoa, Ev ).

% Someone who has the instrument of the crime and who bears a
% relation to the victim.

potencial( suspeito, Ev,  Pessoa  ) :-
     vítima( Ev, Vítima ),
     instrumento( Ev, I ),
     possui( Pessoa, I ),
     mantem_relação( Pessoa, Vítima ).

%  Rules about potential agent

% Someone who is a known suspect.

potential( autor, Ev, Pessoa ) :-
     arrolado( Pessoa, suspeito, Ev ).

% Someone who is a potential suspect.

potencial( autor, Ev, Pessoa ) :-
     potencial( suspeito, Ev, Pessoa ).

% The person responsible of an animal that is the agent of a
% crime.

potencial( autor, Ev, responsavel_por(A) ) :-
     autor( Ev, A  ),
     tipo_autor( Ev, animal ).

%  Rules about potential instrument

% Any instrument that the agent is known to have had.

potencial( instrumento, Ev, I ) :-
     evento( Ev, autor(Autor),_,_,local(Local),dia(Day) ),
     possui( Autor, I, Local, Day ).
```

```
%  Rules about potential situation of the agent after the crime.

% When agent is known

potencial( situação_autor, Ev, preso ) :-
     autor( Ev, A ),
     not var( A ).

%  Rules about potential situation of the  victim  resulting  from
%   the crime.

% Death, if the instrument is a revolver

potencial( situação_vítima, Ev, morta ) :-
     instrumento( Ev, revolver ).

% Injury, if the action is a rape

potencial( situação_vítima, Ev, ferida ) :-
     ação( Ev, estupro ).

%  Rules about potential victim

% Someone who was present at the crime

potencial( vítima, Ev, Pessoa ) :-
     evento( Ev, _, _, _, local(Local), dia(Dia) ),
     estava( Pessoa, Local, Dia ).

%  Rules about potential action.

% Homicide if instrument is a revolver.

potencial( ação, Ev, homicídio ) :-
     instrumento( Ev, revolver ).

% Kidnapping if victim is very rich.

potencial( ação, Ev, sequestro ) :-
     vítima( Ev, V ),
     status( V, muito_rica ).

% Kidnapping if victim is very important.

potencial( ação, Ev, sequestro ) :-
     vítima( Ev, V ),
     status( V, muito_importante ).

% Robbery if victim is very rich.

potencial( ação, Ev, roubo ) :-
     vítima( Ev, V ),
     status( V, muito_rica ).

%  Rules about potential witness
```

```
% Someone who was at time and place of the crime.

potencial( testemunha, Ev, Pessoa ) :-
     evento( Ev, _, _, _, local(Local), dia(Dia) ),
     estava( Pessoa, Local, Dia ).

%  Rules about potential motive

% Jealousy if wife is victim.

potencial( motivo, Ev, ciume ) :-
     rel_autor_vítima( Ev, marido ).

% Jealousy if husband is victim.

potencial( motivo, Ev, ciume ) :-
     rel_autor_vítima( Ev, mulher ).

% Infidelity if wife is the victim.

potencial( motivo, Ev, infidelidade ) :-
     rel_autor_vítima( Ev, marido ).

% Infidelity if husband is the victim.

potencial( motivo, Ev, infidelidade ) :-
     rel_autor_vítima( Ev, mulher ).

% Revenge if the victim of that crime was  a  suspect  in  another
% crime.

potencial( motivo, Ev1, vinganca ) :-
     vítima( Ev1, V ),
     arrolado( V, suspeito, Ev2 ).

% Revenge if the victim of that crime was the  author  of  another
% crime.

potencial( motivo, Ev1, vinganca ) :-
     vítima( Ev1, V ),
     autor( Ev2, V ).

% To cover-up another crime  if  the  victim  of  that  crime  was
% witness of it.

potencial( motivo, Ev1, encobrir_crime ) :-
     vítima( Ev1, V ),
     arrolado( V, testemunha, Ev2 ).

%  Other Rules

% Someone has a motive of C1 if he is the author of a crime C2 and
% the victim of C1 was a witness of C2.

tem_motivo( Pessoa, Ev1 ) :-
```

```
tem_motivo( Pessoa, Ev1 ) :-
     vítima( Ev1, V ),
     arrolado( V, testemunha, Ev2 ),
     autor( Ev2, Pessoa ).

% Someone who is a competitor of the victim.

tem_motivo( Pessoa, Ev ) :-
     vitima( Ev, Vitima ),
     relação( Pessoa, Vitima, Rel ),
     pertence( Rel, [inimigo, rival, adversario,socio,concorrente] )

% If there is a known relation between them.

mantem_relacão( P1, P2 ) :-
     relação( P1, P2, Rel ),
     Rel \= desconhecido.
```

**4. User Model** : We are currently using a very simple user model. This module stores information related to what the user knows about each crime and about the crime domain. It is a file that is updated whenever the system generates a reply. Information contained in the User Model is important for the system to establish what to answer, e.g. it will not say what the user already knows. Of course if the user asks a question for which it already knows the response, the system is going to tell him something he already knows. What we want to make clear here is that the system provides additional (pertinent) information in its answers to both yes and no questions. If this pertinent information is already known to the user, then it will not tell him again (as additional information) unless in a context of an explanation.

The user knowledge of each fact is represented by :

```
sabe( user, autor( 1, joão ) ).
sabe( user, acão( 1, homicidio ) ).
etc...
```

Crime and Legal Knowledge Bases are consulted mainly during the proof of the system query but they are also used in the justification of the user misconceptions jointly with Commom Sense Knowledge Base. These justifications call for a trace of the applied rules.

### 3.3. FOCUS

We will be assuming a subsystem that indicates where is the focus of the user question. In a Yes/No question the focus indicates which information in the question is most doubtful to the speaker/writer. When the response is No, the first candidate for a wrong value is the focussed information. Knowledge of the focus is a crucial factor in providing the appropriate answer to a question whose response is No. For example, if the system knows

asks the question : "Did John kill Jane ?", the appropriateness of the answer is affected by focus in the following way :

If the focus is the agent ( in this case 'John' ), then the answer must indicate who is the correct agent of the crime - "No, Peter killed Jane."

If, however, the focus is the victim, then the answer must indicate who is the victim of the crime for which John was the agent - "No, John killed Mary."

The semantics of the system queries generated after a failure is therefore driven by the focus. The subsystem assumed will probably be controlling the context of discourse in order to determine the focus of each question. But the focus may be undetermined. For example, if the user's first interaction with the system is a question, there will not be enough contextual information avaiable for the discourse component to identify the focus (in fact, this is true for statements as well). In these cases, unless the focus is directly signalled by the user's choice of syntactic structure (e.g. "Is it John who killed Jane ?" or "Is it Jane that John killed?"), the deep generator will have to determine the focus by applying default, domain-dependent, focus rules. The information contained in the rules and their order of application reflect our views on the pragmatics of questions about crime :

- The less central information is to a crime the more likely it is to be a source of error.
- Information about the location and time of a crime tend to be likely sources of error.
- Information about the agent of a crime is almost always uncertain.
- The victim tends to be that entity in a crime of which people are usually most sure.

The mechanism for generation focus uses the following rules :

1. If event arguments and more than one extra_event argument are instantiated
   then default focus = all extra-event arguments
2. If event arguments and only one extra-event argument are instantiated
   then default focus = the extra event argument
3. If crime type and penalty are instantiated
   then default focus = penalty
4. If a location and/or time of a crime is instantiated
   then default focus = *local* and/or *data*
5. If an author of a crime is instantiated
   then default focus = *autor*
6. If a criminal action is instantiated
   then default focus = *ação*
7. If a victim is instantiated
   then default focus = *vítima*

These rules are applied in the order given.

## 3.4. USER QUESTION CLASSES

We now present the types of user questions with which we deal.

Remember that a user question must refer to an event. In addition to the event arguments - agent, action, victim, local, day - other information that can be requested are :

- the role of a person in a event.
- the situation of agent or victim.
- the motive of the action.
- the instrument used in the event.
- the legal classification of the commited crime.
- the punishment for the crime.

The input question is classified depending on what information it contains.

**Class 1.** : Questions that involve only *event information* :
autor, acção, vítima, local, dia

**Class 2.** : Questions that involve *extra-event information.* (see above)

**Subclass 2.1.** : When it involves the relation
arrolado( Pessoa, Papel, Evento )

**Subclass 2.2.** : When it involves the relations
motivo( Evento, Mot ) and instrumento( Evento, In)

**Subclass 2.3.** : When it involves the relations
situação_autor(Evento, SA) and situação_vítima(Evento,SV)

**Subclass 2.4.** : When it involves the relation
tipo_crime( Evento, Tipo )

**Subclass 2.5.** : When it involves the relation
pena( Tipo_crime, Pena )

## 3.5. INTERACTIONS BETWEEN FOCUS KNOWLEDGE AND SEMANTICS.

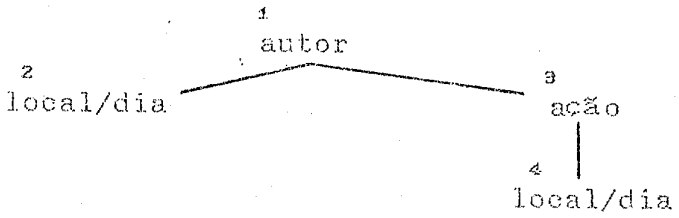
For each Yes/No question, which queries can potentially be asked to the system ?

The first question made is the system query produced by the parser. If this proves false the system then has to find the correct answer. If the query has many instantiated arguments there will be many possible queries. As shown in section 3.3 the best sequence is determined by the focus of the question.

In what follows we present the sequence of alternative system queries that are made after the establishment of a NO response. The choice will depend on the question class and on the focus information.

We will use a tree structure, constructed from the focus rules, to represent the sequence of queries. The goal sequence is equivalent to exploring the tree in a *depth-first* manner. The presence of an argument means that the query will have this argument uninstantiated. At the execution of a node, all the ancestor arguments are also uninstantiated.

For example, the tree




refers to the sequence of attempts for Class 1. question :

1. evento( AUTOR, acão, vítima, local, dia )
2. evento( AUTOR, acão, vítima, LOCAL, DIA )
3. evento( AUTOR, ACÃO, vítima, local, dia )
4. evento( AUTOR, ACÃO, vítima, LOCAL, DIA )

where uppercase letters refers to uninstantiated variables and lowercase letters to instantiated variables.

There will be cases where some arguments will not be given by the user.In these cases, not all the queries of the sequence will be tried. The limit is that not all given arguments be uninstantiated simultaneously. Moreover , a query is considered invalid when the action (acão) is the only instantiated argument.

The information *local* and *dia* are considered to belong to the same category of information : they are able to determine an event but are 'doubtful' when given. They seem to have the same probability to be wrong and appear to be complementary when only one of them is to be given by the system.

Let's see the influence of focus in each question class.
In the following examples we indicate the focussed item by underlining it.

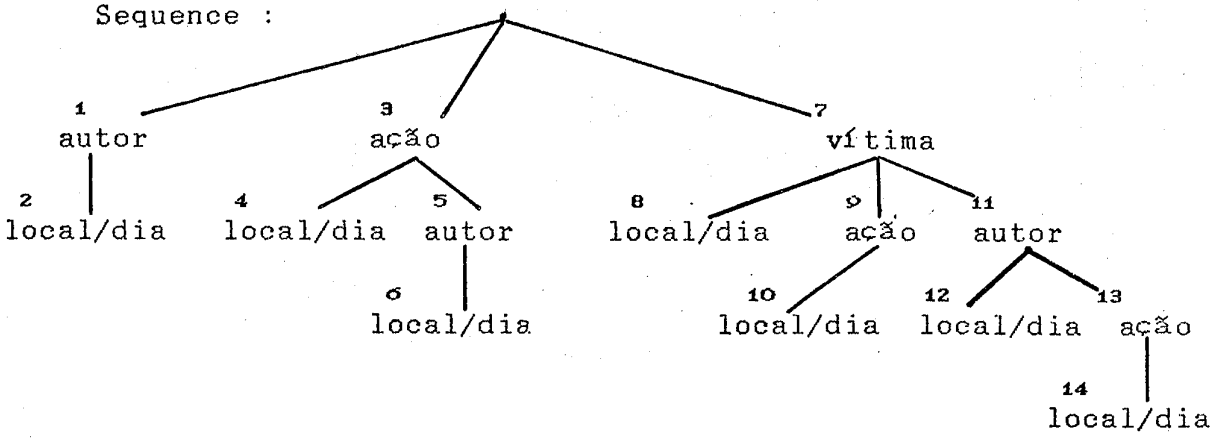
**Class 1.**

Questions that involve event information :

autor, acão, vítima, local, dia

**If focus = autor**

Sequence :




In the following examples we indicate the focussed item by underlining it.

**Ex.1.** João matou Maria no Leme ? *Did João kill Maria in Leme ?*

evento(autor(João), ação(homicídio), vítima(Maria), local(Leme), dia(_) ) ==> **False**

Response = **No.**

Queries :

1. Quem matou Maria no Leme ? *Who killed Maria in Leme ?*

evento(autor(AUTOR), ação(homicídio), vítima(Maria), local(Leme), dia(_) )

2. Quem matou Maria e onde ? *Who killed Maria and where ?*

evento(autor(AUTOR), ação(homicídio), vítima(Maria), local(LOCAL), dia(DIA) )

3. O que João fez a Maria no Leme ? *What did João do to Maria in Leme ?*

evento(autor(João), ação(AÇÃO), vítima(Maria), local(Leme), dia(_) )

4. O que e onde João fez a Maria ? *What did João do to Maria and where did he do it ?*

evento(autor(João), ação(AÇÃO), vítima(Maria), local(LOCAL), dia(DIA) )

5. Quem fez o quê a Maria no Leme ? *Who did what to Maria in Leme ?*

evento(autor(AUTOR), ação(AÇÃO), vítima(Maria), local(Leme), dia(_) )

6. Quem fez o quê a Maria em algum lugar ? *Who did what to Maria and where ?*

evento(autor(AUTOR), ação(AÇÃO), vítima(MARIA), local(LOCAL), dia(DIA) )

7. João matou quem no Leme ? *Who did João kill in Leme ?*

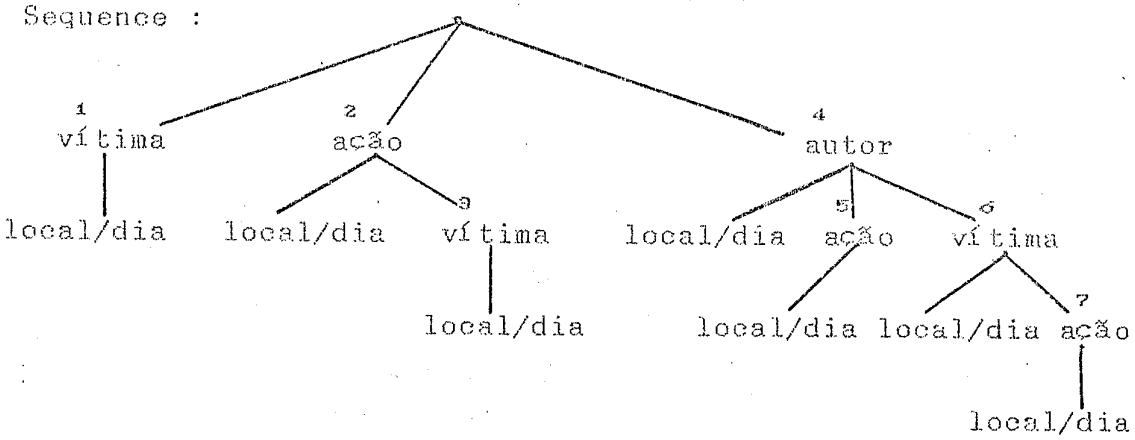
evento(autor(João), ação(homicídio), vítima(VÍTIMA), local(Leme), dia(_) )

8.João matou quem e onde ? *Who did João kill and where did he do it ?*
evento(autor(João), ação(homicídio), vítima(VÍTIMA), local(LOCAL), dia(DIA) )
etc.....

The questions corresponding to nodes 11 through 14 will not be put to the system because they imply in a very distant event.

If focus = vitima

Sequence :




Ex.2. João matou Maria ? *Did João kill Maria ?*
evento(autor(João), ação(homicídio), vítima(Maria), local(_), dia(_) ) ==> **False**
Response = **No.**

Queries :
1. Quem João matou ? *Who did João kill ?*
evento(autor(João), ação(homicídio), vítima(VÍTIMA), local(_), dia(_) )

2. João fez o quê a Maria ? *What did João do to Maria ?*
evento(autor(João), ação(AÇÃO), vítima(Maria), local(_), dia(_) )

3. João fez o quê a quem ? *What did João do to whom ?*
evento(autor(João), ação(AÇÃO), vítima(VÍTIMA), local(_), dia(_) )

4. Quem matou Maria ? *Who killed Maria ?*
evento(autor(AUTOR), ação(homicídio), vítima(Maria), local(_), dia(_) )

5. Quem fez o quê a Maria ? *Who did what to Maria ?*
evento(autor(AUTOR), ação(AÇÃO), vítima(Maria), local(_), dia(_) )

6. Quem matou quem ? *Who killed who ?*
evento(autor(AUTOR), ação(homicídio), vítima(VÍTIMA), local(_),

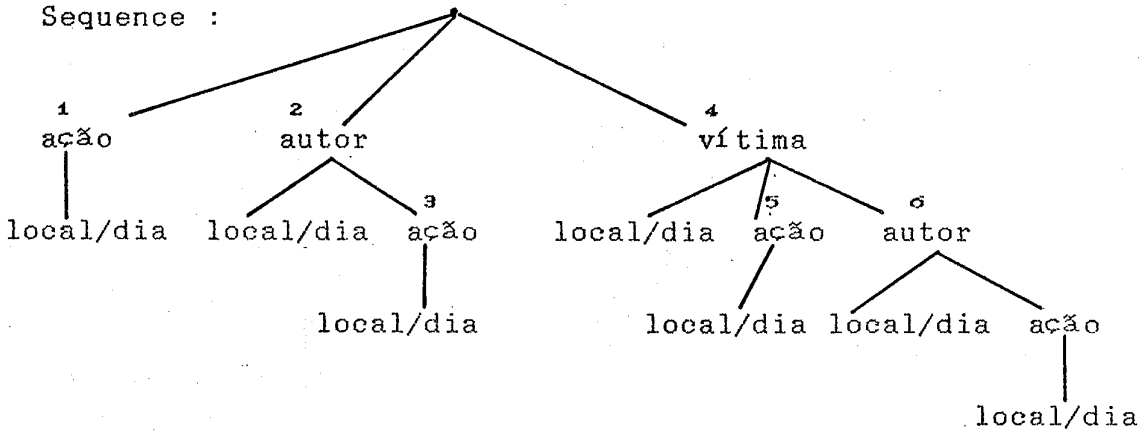
dia(_) )

7. Quem fez o quê a quem ? *Who did what to who ?*
evento(autor(AUTOR), ação(AÇÃO), vítima(VÍTIMA), local(_), dia(_))

These queries are all the possible ones. The queries starting in node 6 are discharged for the reasons given above.

**If focus = ação**

Sequence :




Ex.3. João matou Maria ? *Did João kill Maria ?*
evneto(autor(João), ação(homicídio), vítima(Maria), local(_), dia(_) ) ==> False
Response = No.

Queries :
1. O que João fez a Maria ? *What did João do to Maria ?*
evento(autor(João), ação(AÇÃO), vítima(Maria), local(_), dia(_) )

2. Quem matou Maria ? *Who killed Maria ?*
evento(autor(AUTOR), ação(homicídio), vítima(Maria), local(_), dia(_) )

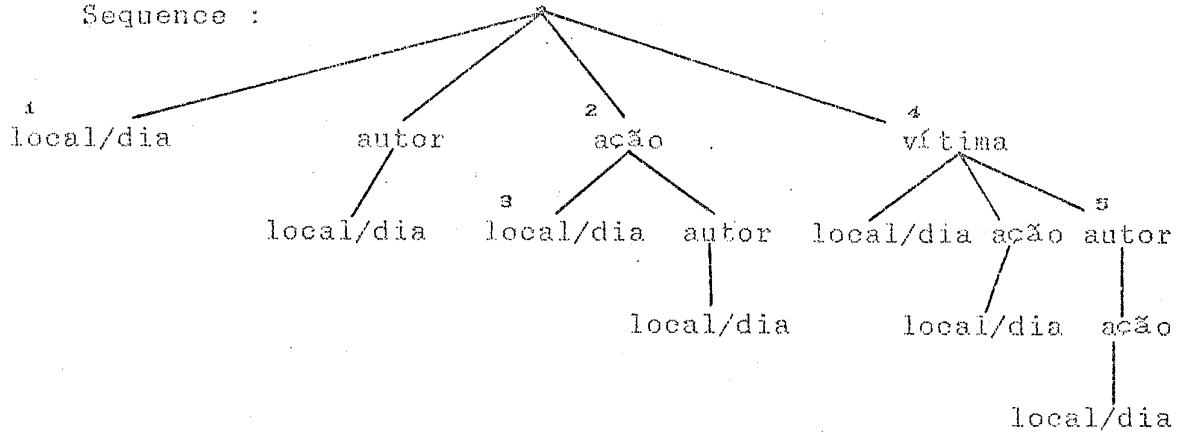
3. Quem fez o quê a Maria ? *Who did what to Maria ?*
evento(autor(AUTOR), ação(AÇÃO), vítima(Maria), local(_), dia(_) )

4. Quem João matou ? *Who did João kill ?*
evento(autor(João), ação(homicídio), vítima(VÍTIMA), local(_), dia(_) )

5. João fez o quê a quem ? *What did João do to whom ?*
evento(autor(João), ação(AÇÃO), vítima(VÍTIMA), local(_), dia(_))

The sequence starting in node 6 is discharged for the same reasons above.

If focus = local/dia




Ex.4. Maria foi morta no Leme ? *Was Maria killed in Leme ?*
evento(autor(_), ação(homicídio), vítima(Maria), local(Leme),
dia(_) ) ==> False
Response = No.

Queries :
1. Onde Maria foi morta ? *Where was Maria killed ?*
evento(autor(_), ação(homicídio), vítima(Maria), local(LOCAL),
dia(DIA) )

2. O que fizeram a Maria no Leme ? *What happened to Maria in Leme?*
evento(autor(_), ação(AÇÃO), vítima(Maria), local(Leme), dia(_))

3. Onde e o quê fizeram a Maria ? *What happened to Maria and where ?*
evento(autor(_), ação(AÇÃO), vítima(Maria), local(LOCAL),
dia(DIA))

4. Quem foi morto no Leme ? *Who was killed in Leme ?*
evento(autor(_), ação(homicídio), vítima(VÍTIMA), local(Leme),
dia(_) )

The sequence starting in node 5 is discharged for the reasons given above.

**Class 2.**

Questions that involves extra-event information

In these cases the focus may be on these extra-event arguments or on event arguments that appear in the question. In the former case, when a distinct event instance is necessary -the procedure to be performed is that described in **Class 1.**

It is important to note that when the *correct instance* involves a different event the new event information will be shown to the user only if the information about agent(autor) or

victim(vítima) or local and day(local/dia) coincide with that from the user question. Otherwise this *correct instance* will be very distant from what would be cooperative to the user (i.e. not very relevant to the user's question).

If the initial focus is on the value argument and a correct event must be found, the new value for the focus will be, when given, the agent or local/day or victim, in that order. That test order is based on the usually lack of confidence in crime information. For example, if the user question is

João matou Maria com uma faca ? *Did João kill Mary with a knife ?*

the sequence to be tried will be :

1. O que João usou para matar Maria ? *What did João use to kill Maria ?*
2. Quem matou Maria com uma faca ? *Who killed Mary with a knife?*
3. Quem João matou com uma faca ? *Who did João kill with a knife?*

etc...

**Subclass 2.1**

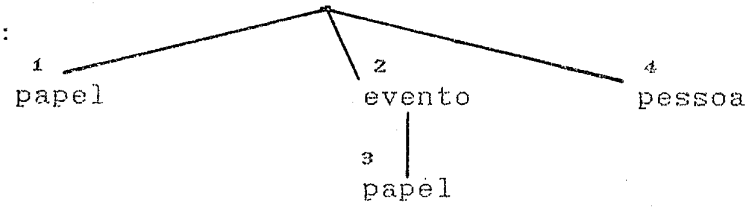
The question is about the relation

arrolado( pessoa, papel, evento )

The focus may be on
a) the role of the person (papel)
b) the person who has that role (pessoa)
c) some argument of the event (evento).

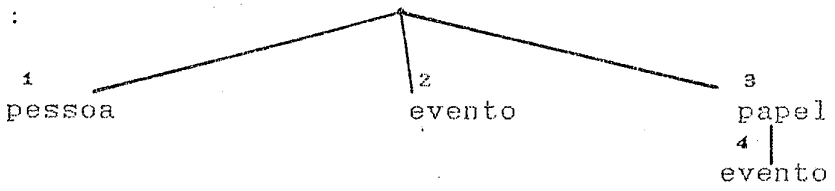
If focus = papel

Sequence :



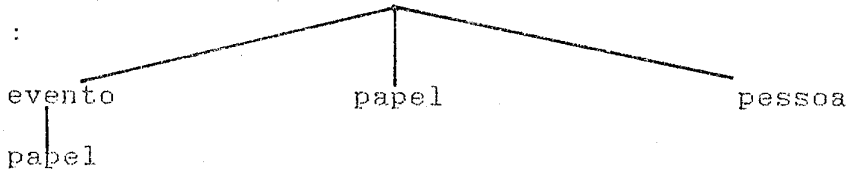

If focus = pessoa

Sequence :




If focus on evento

Sequence :



Ex.5. João é testemunha da morte de Maria ? *Was João a witness of Maria's murder ?*

```
arrolado(João,  testemunha,  evento(autor(_),  acão(homicídio),
        vítima(Maria), local(_), dia(_) ) ) ==> False
```

Response =No.

Queries :

1. Qual o papel de João na morte de Maria ? *What is João's role in Maria murder ?*

```
arrolado(João, PAPEL, evento(autor(_),....) )
```

2. Em qual crime João foi testemunha ? *In what crime João was witness ?*

```
arrolado(João, testemunha, EVENTO1)
```

3. Qual é o crime no qual João tem participacão e qual foi o seu papel ? *What is João's role in what event ?*

```
arrolado(João, PAPEL, EVENTO1)
```

4. Quem foi testemunha da morte de Maria ? *Who was witness in Maria murder ?*

```
arrolado(PESSOA, testemunha, evento(autor(_),....) )
```

Ex.6. Ana é suspeita de matar José ? *Is Ana suspected of killing José ?*

```
arrolado(Ana,  suspeito,  evento(autor(_),  acão(homicídio),
        vítima(José), local(_), dia(_) ) ) ==> False
```

Response = No.

Queries :

1. Quem é suspeito de matar José ? *Who is suspected of killing José ?*

```
arrolado(PESSOA, suspeito, evento( autor(_),....) )
```

2. Ana é suspeita de que crime ? *What is Ana suspected of ?*

```
arrolado(Ana, suspeita, EVENTO1)
```

3. Qual a participação de Ana na morte de José ? *What is Ana's role in José murder ?*

```
arrolado(Ana, PAPEL, evento(autor(_),....) )
```

4. De qual crime Ana participou e qual foi o seu papel ? *In which crime does Ana have a role and what is that role ?*

```
arrolado(Ana, PAPEL, EVENTO1)
```

Subclass 2.2.

Questions involving the relations :

motivo( evento, motivo ) and instrumento( evento, inst )

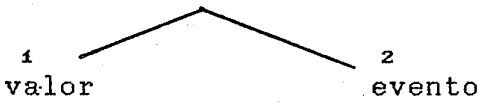
We will denote these relation by :

relação( evento, valor )

Again the focus may be on or out of these extra-event relations:
a) on valor, or
b) on some event argument

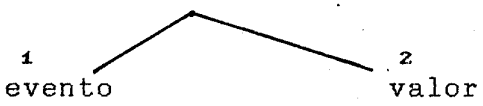
**If focus = valor**

Sequence :




**If focus on evento**

Sequence :




**Ex.7.** João matou Maria com uma faca ? *Did João kill Maria with a knife ?*
instrumento( evento(autor(João), ação(homicídio), vítima(Maria), local(_), dia(_) ) , faca) ==> False

Response = No.

Queries :
1. Com o que João matou Maria ? *What did João use to kill Maria ?*
instrumento( evento(autor(João),....), VALOR )

2. Detect the mistakes related to this event and the instrument used. The correct answer could be : Antonio killed Maria with a gun.
instrumento( EVENTO1, Valor )

Ex.8. Ana matou José porque ele era infiel ? *Did Ana kill José because he was unfaithful ?*
motivo( evento(autor(Ana), ação(homicídio), vítima(José), local(_), dia(_) ), infiel ) ==> False

Response = No.

Queries :
1. Quem matou José por infidelidade ? *Who killed José because of infidelity ?*
motivo( EVENTO1, infiel )

2. Porque Ana matou José ? *Why did Ana kill José ?*
motivo( evento(autor(Ana),....), VALOR )

Subclass 2.3.

Questions involving the relations :

situação_vítima(evento, sitv) and situação_autor(evento,sita)

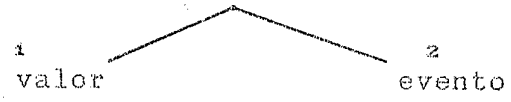
that will be denoted by

situação( evento, valor )

Again the focus may be on
a) the value (valor), or
b) some event argument

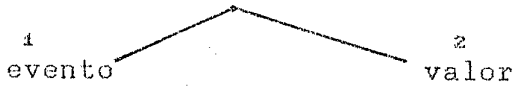
If focus = valor

Sequence :




If focus on evento

Sequence :




Ex.9. João fugiu após matar Ana ? *Did João escape after killing Ana ?*
situação_autor(evento(autor(João), ação(homicídio), vítima(Ana), local(_), dia(_)), foragido ) ==> False

Response = No.

Queries :
1. Qual é a situação de João após matar Ana ? *What happened to João after he killed Ana ?*
situação_autor(evento(autor(João),....), VALOR)

2. Detect the mistakes of the event and the value of agent situation.
situação_autor( EVENTO1, Situação )

Ex.10. Ana está em coma ? *Is Ana in coma ?*
situação_vítima( evento(autor(_), ação(_), vítima(Ana), local(_), dia(_) ), coma ) ==> False

Response = No.

Queries :
1. No mesmo evento,quem está em coma ? *Who is in coma ?*
situação_vítima( EVENTO1, coma )

2. Qual a situação de Ana ? *What is Ana health situation ?*

situação_vítima( evento(autor(_), ação(_), vítima(Ana),...), VALOR )

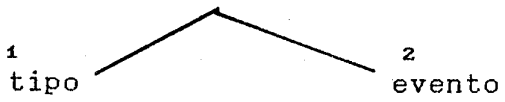
**Subclass 2.4.**

Questions that involve the relation

tipo_crime( evento, tipo )

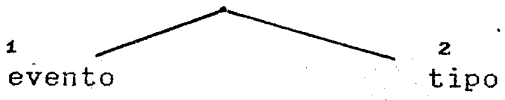
**If focus = tipo**

Sequence :




**If focus on evento**

Sequence :




**Ex.11.** João cometeu homicidio culposo ? *Did João commit culpable homicide ?*

tipo_crime( evento(autor(João), ação(homicídio), vítima(_), local(_), dia(_)), homicídio_culposo ) ==> **False**

Response = **No.**

Queries :

1. Que tipo de crime João cometeu ? *What type of crime did João commit ?*

tipo_crime( evento(autor(João),....), TIPO )

2. Detect the mistakes of this event and the type of crime commited.

tipo_crime( EVENTO1, Tipo )

**Ex.12.** Ana fez aborto_necessário ? *Did Ana commit an necessary abortion ?*

tipo_crime( evento(autor(Ana), ação(aborto), vítima(_), local(_), dia(_)), aborto_necessario ) ==> **False**

Response = **No.**

Queries :

1. Quem fez aborto_necessario ? *Who commited necessary abortion ?*

tipo_crime( EVENTO1, aborto_necessario )

2. Que tipo de crime Ana cometeu ? *What did Ana commit ?*

tipo_crime( evento(autor(Ana),...), TIPO )

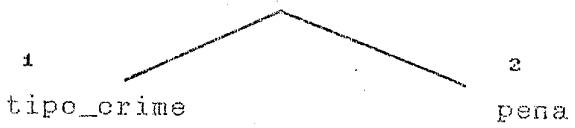
Subclass 2.5.

Questions involving the relation

pena( tipo_crime, pena )

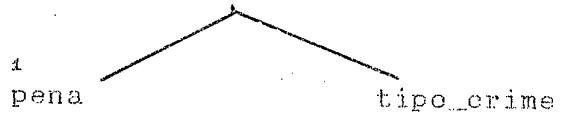
If focus = tipo_crime

Sequence :




If focus = pena

Sequence :




Ex.13. A pena de aborto provocado é de 2 anos ? *Is penalty for provoked abortion 2 years ?*
pena(aborto_provocado, 2) ==>False

Response = No.

Queries :
1. Qual é o tipo de crime cuja pena é de 2 anos ? *What is the crime which penalty is 2 years ?*
pena( TIPO, 2 )

2. Qual é a pena para aborto_provocado ? *What is the penalty for provoked abortion ?*
pena( aborto_provocado, PENA )

**Ex.14.** A pena para homicídio culposo é de 10 anos ? *Is the penalty for culpable homicide 10 years ?*
pena( homicídio_culposo, 10 ) ==> **False**

Response = **No.**

Queries :
1. Qual é a pena para homicídio culposo ? *What is the penalty for culpable homicide ?*
pena( homicídio_culposo, PENA )

2. Qual é o crime cuja pena é de 10 anos ? *What is the crime which penalty is 10 years ?*
pena( TIPO_CRIME, 10 )

## 3.6. PRAGMATICS

We will be discussing here what information should be added to the answer after the system finds a Yes or No response . The decision on *what to say* is based on the focus and results in a list of system queries.

Two schemes will be proposed : one for Yes response, other one for No response.

### 3.6.1. YES ANSWER

Figure 5 describes the treatment of yes answers.

The Yes answer is comprised of four parts :

**1. Affirmation** : The response YES.

**2. Focused Argument Evidence** : Show evidences to reinforce the focussed argument validity, if any.

**3. Why not ?** : At times the user has some reason to believe in the negative response to his question. In this case the counter-evidences for the negative response - if known by the system - should be shown. They are constructed from the search for another value for potential "focus-argument".

**4. Complement** : Complete the answer with relevant information about the crime.

We will now show how to get each of these.

#### 3.6.1.1. Affirmation

This part comes from the system's proof that the proposition questioned is true.

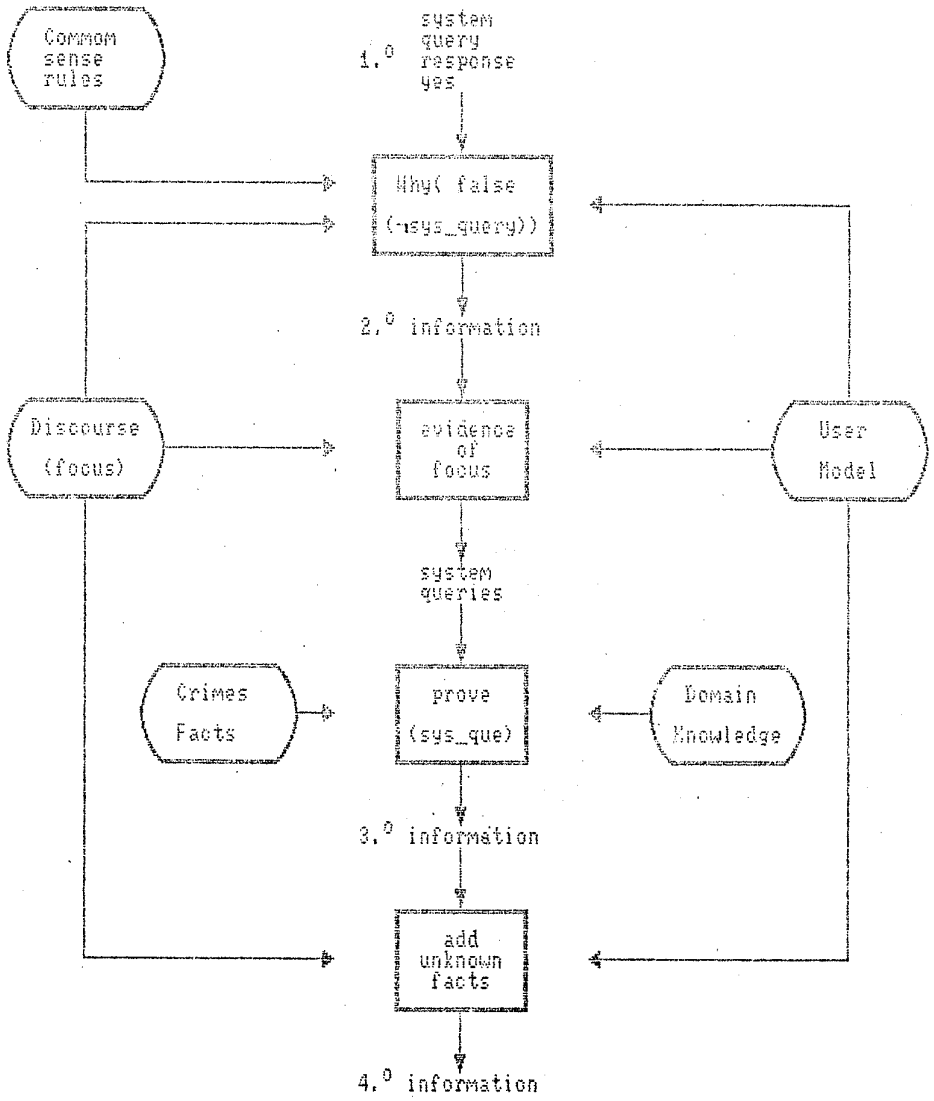
#### 3.6.1.2. Focused Argument Evidence

Remember that the focussed argument is information in the question about which the user is most doubtful.
The queries to the system depends on the focussed argument.
The answer should show information that reinforce the argument validity.

Class 1.

Focus = autor : show the crime motive (motivo) the action details (modo) and the agent situation.

Focus = vítima : show the crime motive (motivo) and the action details (modo) and the victim situation.




$$1.^0 + 2.^0 + 3.^0 + 4.^0 \Longrightarrow \text{rst(nucl( ), satel( ))}$$

**Fig.5. Constructing a Yes Answer**

Focus = ação : show the action details (modo) and the motive (motivo).

Focus = local/dia : show anything. We do not handle evidence for local and day.

Class 2.

Subclass 2.1.

Focus = pessoa or papel : show why. The system must prove "papel(X)" and give the trace information.

Focus on event : see Class 1.

Subclass 2.2.

Focus = valor : show action details (modo), if relation = instrumento( evento, valor ).
show agent/victim relation (rel_autor_vítima), if relation = motivo( evento, valor ).

Focus on event : see Class 1.

Subclass 2.3.

Focus = valor : show action details (modo), if relation = situação_vítima(evento, valor)
show capture local and day (captura) and escape mode (modo_fuga), if relation = situação_autor(evento, valor)

Focus on event : see Class 1.

Subclass 2.4.

Focus = tipo : show why. Get this information from the proof trace
Focus on event : see Class 1.

Subclass 2.5.

Focus = tipo_crime or pena : show the complete punishment year interval for that type of crime.

3.6.1.3. Why not ?

We have to find a new value for the focussed argument such that it is *potential* for the user (see section 3.2.).

Ex. João matou Paulo ? *Did João kill Paulo ?*

Sim. Apesar de Antonio ser inimigo de João, não foi ele a vítima.

*Yes. In spite of the fact that Antonio is his known enemy, he was not the victim.*

This kind of knowledge is extracted from the Crime Knowledge Base and Commom Sense *potential* rules.

For example, if the information about the victim is focussed, the system tries to prove some rule about potential victim with another victim value. If it is possible, the proof trace is shown as user justification for the possible misconception.

3.6.1.4. Complement

A list of relevant crime information may be included in the response. The main requirement is that the user does not already know it. It is possible that the system does not have all of these suggested information. Again, what to say depends on the focussed argument.

In what follows, we present the suggested information for each class of Yes/No question according to focus, in decreasing order of relevance.

Class 1.

Focus = autor : vítima, rel_autor_vítima, motivo, local/dia, situação_autor, situação_vítima.

Focus = vítima : autor, rel_autor_vítima, motivo, local.dia, situação_autor, situação_vítima.

Focus = ação : vítima, autor, rel_autor_vítima, motivo, local/dia, situação_autor, situação_vítima.

Focus = local/dia : autor, vítima, ação, rel_autor_vítima, motivo, modo.

Class 2.

Subclass 2.1.

Focus = pessoa or papel : autor, modo.
otherwise, see Class 1.

Subclass 2.2.

Focus = valor : autor, vítima, motivo, local/dia, situação_autor, situação_vítima, if relation = instrumento;
autor, vítima, modo, local/dia,

situação_autor, situação_vítima, if relation = motivo.

otherwise, see Class 1.

Subclass 2.3.

Focus = valor : autor, vítima, modo, local/dia, situação_vítima, if relation = situação_autor;
autor, vítima, motivo, modo, local/dia, situação_autor, if relation = situação_vítima.
otherwise, see Class 1.

Subclass 2.4.

Focus = tipo : autor, vítima, situação_autor, situação_vítima

otherwise, see Class 1.

Subclass 2.5.

Focus = tipo_crime or pena : add anything.

otherwise, see Class 1.

### 3.6.2. NO ANSWER

Figure 6 shows the treatment of a no answers.

Negative replies to Yes/No questions are divided into four parts :

1. Negation : The Response NO.

2. Correct Answer : This part of the reply presents the correct information about the event mentioned by the user.

3. Misconceptions Sources : The system should try to justify the user mistakes showing the potential source for each one.

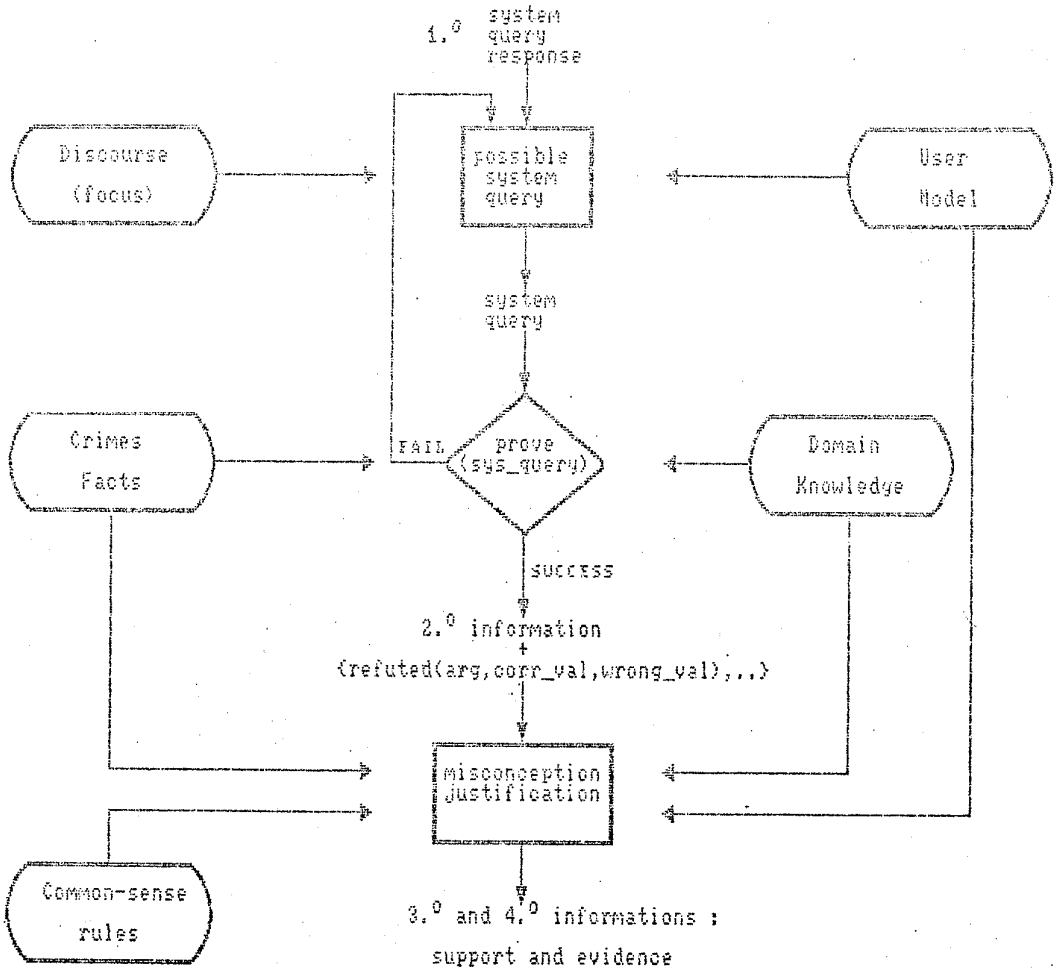
4. Complement : Same as Yes Response.

#### 3.6.2.1. Negation

The No response comes from the system failure answer for the initial question.

#### 3.6.2.2. Correct Answer

Whenever the system fails to prove the user's query, it attempts to provide the user with information that should increase




$$1.^0 + 2.^0 + 3.^0 + 4.^0 \Longrightarrow \text{rst(nucl( ), satel( ))}$$

Fig.6. Constructing a No Answer

his understanding of the circumstance presented in the query. The best situation is where the system is able to find a replacement for the incorrect focussed information, and this is what it attempts to do first. When it cannot, it searches for other information relevant to the circumstance. This search is directed by focus, as discussed in section 3.5.

The attempt to prove the input query produces a list of the refuted arguments, represented as

refutado( argument, correct_value, wrong_value )

Ex. U.Q. : João matou Maria ?
Subprodut : [ refutado(autor, José, João), refutado(ação, estupro, homicídio) ]
S.A. : Não. José estuprou Maria.

### 3.6.2.3. Misconceptions Sources

There is no direct influence of focus on the process of detecting misconception source : every user misconception must be corrected and justified, where possible. The user misconceptions are obtained as described in 3.6.2.2.

Even when the system justifies the misconception it will give evidence for the correct argument value. This information is equivalent to that from section 3.6.1.2, where the focussed argument is substituted by the refuted argument. So the justification part will have two subparts :

a) evidence for the correct argument value and
b) support for the user misconception,

for each user mistake. Note that in this case the justification for the focussed argument is the least important one because of the meaning of focus : the user is waiting for a negation or confirmation of this argument.

The refutation list must be inspected before the justification generation because there are some cases when the system should not justify. One case is when both agent (*autor*) and victim (*vítima*) were corrected by the "correct answer" generator ( see section 3.6.2.2. ). In such cases the new instance is very far from the event presented in the user question and it makes no sense to try to justify the mistakes. In general, wrong agent and victim refer in fact to another, remote, event.

The support and the evidence depend on the refuted argument :

1.
If refutado( autor, corr_val, wro_val )
Then support :
if potencial( autor, evento, wro_val )
then justify showing this trace.
else check if wro_val is related to the crime

```
                then present arrolado(wro_val, PAPEL, evento)
evidence :
     show motivo and modo

2.
If refutado( vitima, corr_val, wro_val )
Then support :
     if potencial( vítima, evento, wro_val )
     then justify showing this trace.
     else       idem autor above
evidence :     idem autor above

3.
If refutado( ação, corr_val, wro_val )
Then support :
     if potencial( ação, evento, wro_val )
     then show the trace.
     else show similarities and differences between
               corr_val and wro_val.
evidence :
     show modo and motivo

4.
If refutado(pessoa, corr_val, wro_val)  in  arrolado(pessoa,pap,e)
Then support :
     if potencial( pap, e, wro_val )
     then show trace.
     else check if wro_val is related to the crime
                   arrolado( wro_val, PAPEL, e )
               then show similarities and differences
                         between pap and PAPEL.
evidence :
     show why with the trace of "pap( corr_val )".

5.
If refutado(papel, corr_val, wro_val)  in  arrolado(pes,papel,ev)
Then support :
     if potencial( wro_val, ev, pes )
     then show trace.
     else show similarities and differences between
               corr_val and wro_val.
evidence :
     show why with the trace of "corr_val( pes )".

6.
If refutado( motivo, corr_val, wro_val )  in  motivo(ev, motivo)
Then support :
     if potencial( motivo, ev, wro_val )
     then show trace.
evidence :
     show rel_autor_vitima.

7.
If refutado(inst_val,corr_val,wro_val) in instrumento(ev,inst_val)
Then support :
     if potencial( instrumento, ev, wro_val ).
```

```
     then show trace.
     else show similarities and differences between
                corr_val and wro_val.
evidence :
     show modo

8.
If refutado( sit_val, corr_val, wro_val )      in
                                        situação_vítima(ev,sit_val)
Then support :
     if potencial( situação_vítima, ev, wro_val)
     then show trace
evidence :
     show modo

9.
If refutado( sit_val, corr_val, wro_val )    in
                                      situação_autor(ev, sit_val)

Then evidence :
     captura or modo_fuga

10.
If refutado( tipo, corr_val, wro_val )   in  tipo_crime(ev, tipo)
                                         and  pena( tipo, pena )
Then support :
     show the similarities and differences between
     corr_val and wro_val
11.
If refutado( pena, corr_val, wro_val )  in pena( tipo, pena )
Then evidence :
     complete the  punishment  information  with  the  whole  year
     interval.
```

There is no support in case 9 because both possible sit_val values are strongly likely to be assumed by the user.

There is no necessity to provide evidence in 10 because the support information will be always accessible.

There is no support in case 11. because the punishment is a fact established by law.

## 3.7. THE DEEP STRUCTURE

### 3.7.1. THE GENERAL STRUCTURES FOR YES AND NO ANSWER SCHEMAS

In the previous section, we showed how the system generates decisions about what information to include in the answer. The output of this component is a collection of facts. What is left is to provide a structure in which to fit these pieces of information such that the pragmatic relations between them are evident to the reader. This structure, and the evidence of pragmatic relations

are crucial for the ease of comprehension of the information and for ensuring that the final text produces the desired effect on the reader's beliefs.

The schemas presented for Yes and No response in section 3.6. shows us a track of the function of the pieces of information in the resulting text ( the answer ). These schemas drive the search for what to say. So, for each query that the system poses to itself, we know what motivated that query. Given this, for each fact in the answer, the system knows what role that fact plays in the answer, and therefore how those facts are related to each other. This is as follows :

The Yes Answer Schema :

1. Affirmation
2. Focused Argument Evidence
3. Why not ?
4. Complement

Let's identify some relations between parts of this schema. The general structure for Yes answer is shown in figure 7.

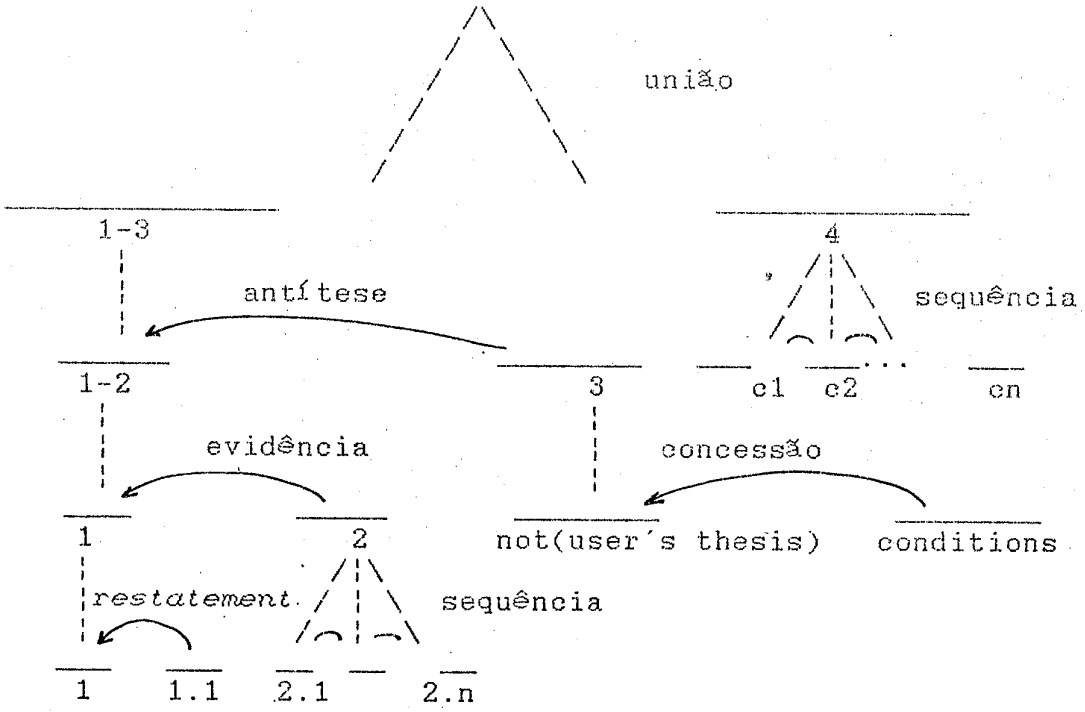



Fig.7. Schema for Yes Answer

The affirmation (Part 1) may contain an optional **restatement** of the affirmative ( for example, Yes (nucleus), he killed him (satellite) ).

Given that the role of the focused argument evidence (Part 2) is to reinforce (ie. increase) the reader's belief in the affirmation, it is a satellite component of an **evidence** (evidência) relation whose nucleus is in part 1. The corresponding

satellite may be a **sequence** of information as the crime motive, the crime consequences, etc.

Part 3 shows a wrong, but plausible, alternative for the focused argument that may have been considered by the user. This is the user's thesis for expecting the negative answer, thus it is a satellite of an **antithesis** (antítese) relation whose nucleus is the affirmation and its evidence. Moreover, the user's belief in the negation constitutes a **concession** (concessão) for those plausible reasons.

Part 4 is quite independent from the rest of text in the sense that the information it contains is not foreseen but depends on the user model. For this reason it is only joined to the rest of the text through the multi-nuclear relation - **union** (união). On the other hand, parts 2, 3 and 4 may contain several pieces of information that are related to each other.

For example, sometimes evidence constitutes a trace of a proof; in this case we have a **conditional** (condição) relation between what was proved (nucleus) and its conditions (satellite). Remember that part 3 involves the attempt to find a new value for a potential focused argument, so there will be a concession relation between the conditions of the trace (nucleus) and the negation of potential argument (satellite).

In the Complement part, non-related pieces of information are often together because of the goal to complete the user information about the crime. When this is the case, a non-nuclear relation - **sequence** (sequência) - relate them.

Note that not all these parts will eventually be instantiated in the answer, which does not change the above observations.

The No Answer Schema :

1. Negation
2. Correct Answer
3. Evidences for refuted arguments
4. Sources for misconceptions detected
5. Complement

The above considerations about restatement relation and complement part are still valid. Note that there is an antithesis relation between the negation and the correct answer and its evidences. Although the fact presented in Correct Answer part may not exclude that fact presented by the user, from the point of view of the system, everything it doesn't know it is not true.

The correct answer part can also have related constituents since it is going to show the relation between the correct argument and the event. There is an obvious evidence relation between part 3 ( satellite ) and part 2 ( nucleus ). The parts 1, 2 and 3 form the nucleus of an antithesis relation with part 4 since each possible element from 4 is a negation of a refuted user's argument and its justification. Both parts 3 and 4 may be sequences of information about each refuted argument.

Again each evidence or justification is an information unity or an instantiation of another relation. For example, the misconception justification may be a concession relation if the wrong argument is plausible, or an evidence relation (See relation definitions in the next section ).

Again , some parts may be uninstantiated in this general structure for No answer presented in figure 8.

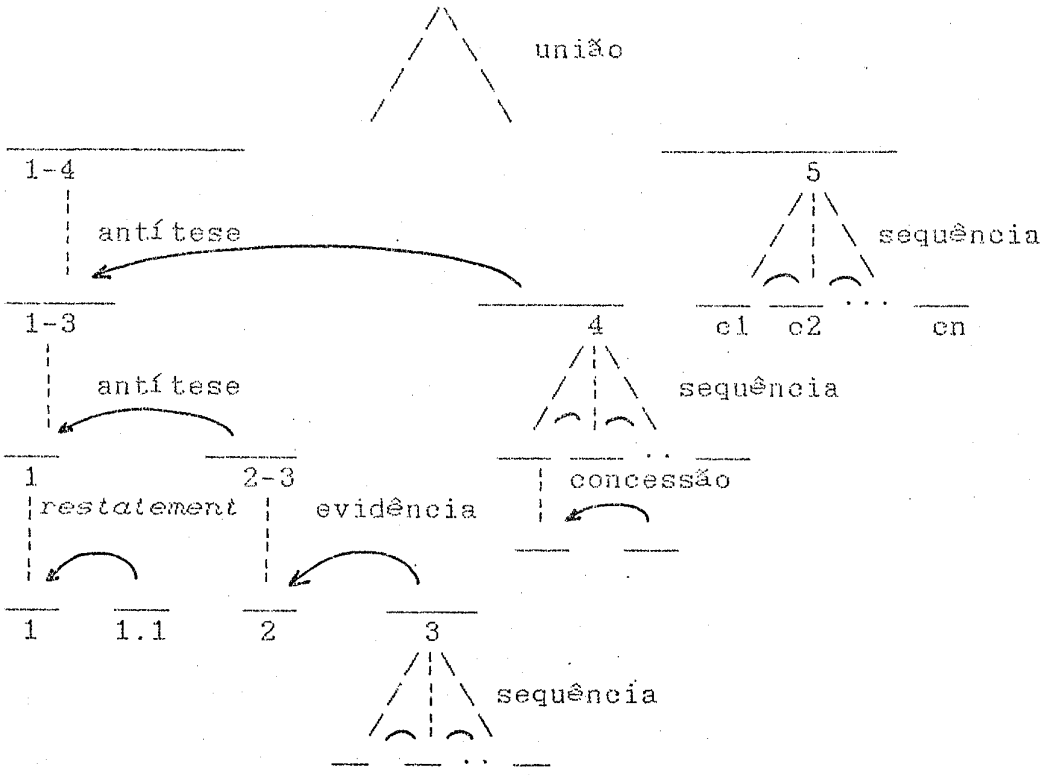



Fig.8. Schema for No Answer

### 3.7.2. DEFINITIONS OF THE RHETORICAL RELATIONS

We have used 8 relations and 9 schemas in our application domain. The majority of them were extracted from [Mann & Thompson-87b]. Let´s see their definitions and examples where they are used.

Following Mann and Thompson, we denote nucleus by N , satellite by S, reader by R and the writer (system) by W.

**1. EVIDÊNCIA (EVIDENCE) Relation**

This expresses a situation where the writer wants the reader to believe something (a claim). To do this he provides information (the evidence) which he feels will make the reader belief the claim. Here the claim is the nucleus and evidence is the satellite.

**constraints on N :** R might not believe N to a degree satisfactory to W.
**constraints on S :** The reader believes S or will find it credible.
**constraints on the N + S combination :** R´s comprehending S increases R´s belief of N.
**the effect :** R´s belief of N is increased.

**locus of the effect :** N.

This relation occurs mainly between parts 1 and 2 of each schema.

Before proceeding, let's define the notion of positive regard. Positive regard refers to the writer's intent with texts and texts spans. Some texts are intended to create belief. Others are intended to create approval or interest. Still others are intended to create desire to act. These - approval, belief and desire to act - are all varieties of positive regard.

## 2. ANTITESE (ANTITHESIS) RELATION

The desired effect of this relation is to cause the reader to have positive regard for the nucleus.

Example:

1.Every rule has exceptions,

2.but the tragic and too-common tableaux of hundreds or even thousands of people snake-lining up for any task with a paycheck illustrates a lack of jobs,

3. not laziness.

In unit 3, the writer considers the thesis that unemployment can be explained in terms of laziness, but he clearly favors (i.e., has positive regard for) the proposition in unit 2 : unemployment has its root in a lack of jobs.

**constraints on N :** W has positive regard for the situation presented in N.
**constraints on S :** none.
**constraints on the N + S combination :** the situations presented in N and S are in contrast; because of an incompatibility that arises from the contrast, one cannot have positive regard for both the situations presented in N and S; comprehending S and the incompatibility between the situations presented in N and S increases R's positive regard for the situation presented in N.

**the effect :** R's positive regard for N is increased
**locus of the effect :** N.

We have an antithesis relation between the correct answer and the misconceptions detected. Ex : No, Antonio is the crime agent (nucleus); João is not the agent although he is Pedro' enemy and was there when Pedro was killed (satellite).

## 3. CONCESSÃO (CONCESSION) RELATION

This relation shares the antithesis property that the desired effect is to cause the reader positive regard for the nucleus. They differ in that Concession is not a subtype of Constrast, while Antithesis is.

One obvious way to signal a Concession relation is an *although* clause. Example :

1. Although this material is toxic to certain animals,
2. evidence is lacking that it has any serious long-term effect on human beings.

The writer both signals that units 1 and 2 are compatible and acknowledges their potential incompatibility. That is, the material's toxicity to certain animals is compatible with the lack of evidence that it is harmful to humans, but it is also potentially incompatible with it, since toxicity to animals often implies toxicity to humans.

constraints on N : W has positive regard for the situation presented in N.
constraints on S : W is not claiming that the situation presented in S doesn't hold.
constraints on the N + S combination : W acknowledges a potential or apparent incompatibility between the situations presented in N and S; W regards the situations presented in N and S as compatible; recognizing the compatibility between the situations presented in N and S increases R's positive regard for the situation presented in N.

the effect : R's positive regard for the situation presented in N is increased.
locus of the effect : N and S.

We have concession relation between the conditions of a rule that defines a potential agent, for example, (nucleus) and the negation of this fact : João is not the crime agent although he is Pedro's enemy and he was at crime locality when Pedro was killed (that is, João is a potential agent ).

4. CONDIÇÃO (CONDITION) RELATION

The realization of the nuclear situation depends on the positive realization of the satellite situation.

constraints on N : none.
constraints on S : S presents a hypothetical, future or otherwise unrealized situation (relative to the situational of S).
constraints on the N + S combination : Realization of the situation presented in N depends on realization of that presented in S.
the effect : R recognizes how the realization of the situation presented in N depends on the realization of the situation presented in S.
locus of the effect : N and S.

We have conditional relation between the conditions (satellite) of a Prolog definition rule. When the system shows a trace of a proof, Prolog clauses are presented as conditonal schemas and, in this case, it can be grammaticized in Portuguese by an *se (if)* clause.

## 5. RAZÃO (VOLITIONAL CAUSE) RELATION

Volitional cause involves the action of an agent, typically a person, who controls an action that yields the nuclear situation. It is performed because the agent prefers the outcome or possibly the action itself. Non-volitional cause is the residue-consequentiality without a chosen outcome.

**constraints on N :** presents a volitional action or else a situation that could have arisen from a volitional action.
**constraints on S :** none.
**constraints on the N + S combination :** S presents a situation that could have caused the agent of the volitional action in N to perform that action; without the presentation of S, R might not regard the action as motivated or know the particular motivation; N is more central to W's purpose in putting forth the N-S combination than S is.
**the effect :** R recognizes the situation presented in S as a cause for the volitional action presented in N.
**locus of the effect :** N and S.

We use this relation when the crime motive is to be presented. The event arguments constitute the nucleus and the motive is the satellite. Example : João killed Pedro (nucleus) because Pedro betrayed João (satellite).

## 6. RESULTADO (RESULT) RELATION

This relation expresses the converse of the volitional cause relation.

Here the causing situation is nuclear and the caused and situation is less central.

constraints on N : none.
constraints on S : presents a situation that could have arisen from an action.
constraints on N + S combination : N presents a situation that caused the situation in S; presentation of N is more central to W's purposes in putting forth the N-S combination than is presentation of S.
the effect : R recognizes that the situation presented in N could have caused the situation presented in S.
locus of the effect : N and S.

The result relation occours between the description of a crime fact and effects such as the actual situation of the agent and of the victim.

Although the last two definitions distinguish the focussed element between nucleus and satellite for writer's purposes, the system chooses one or the other depending on what information it is able to add : if it is a motive, the relation is *Razão* ; if it is an effect of the event, the relation is *Resultado*.

7. RESTATEMENT RELATION

constraints on N : none.
constraints on S : none.
constraints on N + S combination : S restates N.
the effect : R recognizes S as a restatement of N.

8. SEQUENCIA (SEQUENCE) NON-NUCLEATED RELATION

Mann and Thompson's definition for sequence relation imposes an order on the spans the relation covers, that is, the relation among the spans is their order of presentation. The simplest example is a food recipe.

Our sequence relation is rather different : it indicates that the involved information does not have any significant relation to each other; they simply were decided at the same moment to be present but they have the goal of complementing the user crime information and because it depends on what the user already knows, this set of information may be quite heterogenous.

constraints on N : multi-nuclear.
constraints on the combination of nuclei : A succession relationship between the situations is presented in the nuclei.
the effect : R recognizes the succession relationships among the nuclei.
locus of the effect : each N.

9. UNIÃO (JOINT) SCHEMA

The joint schema has no corresponding relation. The schema is multinuclear, and no relation is claimed to hold between the nuclei.

Remember we use joint schema with the answer and the complement answer as nuclei. The complement is related only to what the user already knows. Although they are related in the sense that nothing said in correct answer can not appear in complement part, this kind of relation is not intended to be clear to the user.

## 3.8. EXAMPLES

In what follows we provide examples of system answers to input queries about hypothetical crimes.

A corresponding natural language text is shown with the purpose of showing what the system generates. At present we are not dealing with the final format of the text.

We indicate the focussed item by underlining it.

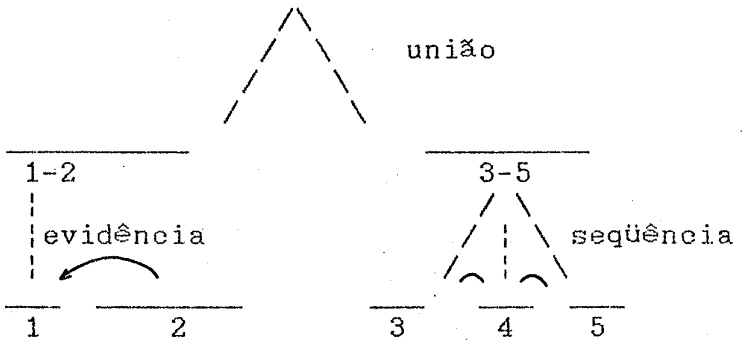
Case 1. Pedro and José were walking in Leme beach, on June 21,when they were assaulted by an unknown man. They reacted to it

and Pedro was hurt by the assailant's knife. The assailant ran away without any money.

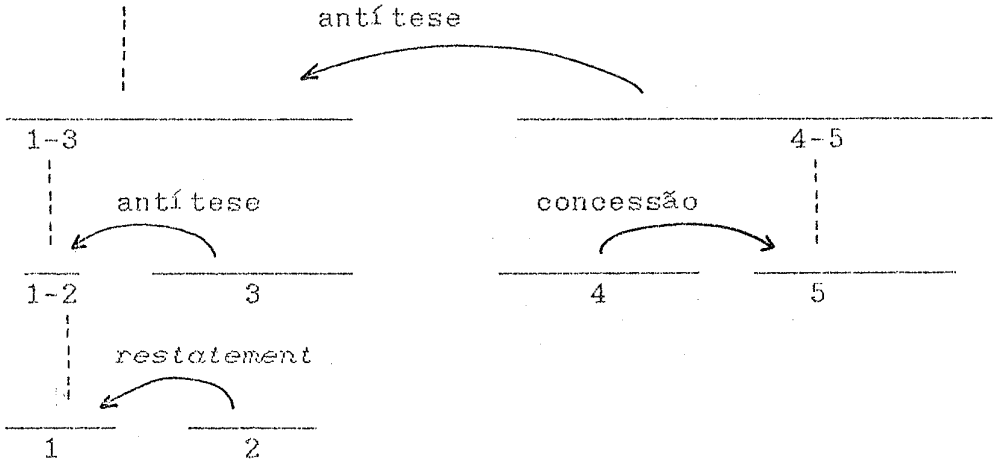
Suppose the user knows the crime motive - robbery, and the crime victim - Pedro.

**U.Q.1.** : Is Pedro hurt ?

**S.A.** : 1. Yes.
2. Pedro was hurt by stabs
3. The assailant run away
4. and he is still unknown.
5. The crime occoured in Leme, on June 21.



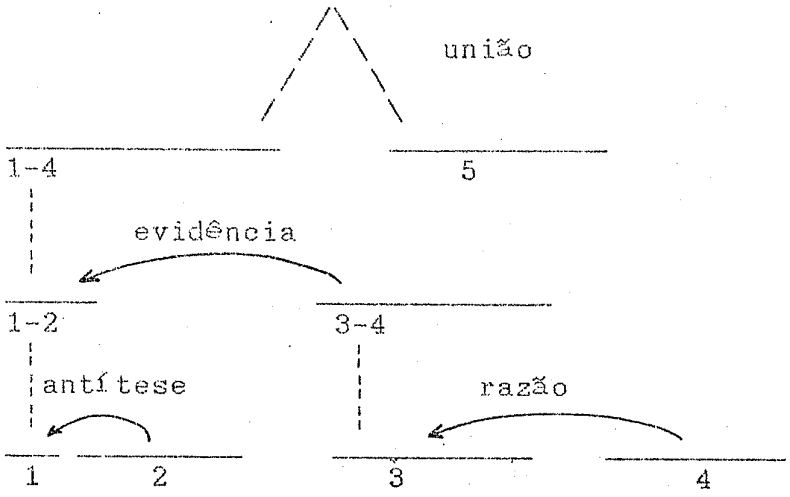

**U.Q.2.** : Is José suspected of having hurt Pedro ?
**S.A.** : 1. No,
2. he isn't.
3. José was a witness.
4. Although José was there at crime local and day,
5. he isn't suspect.




Case 2 : João killed his wife Ana in Copacabana, firing three shots. He declared she was unfaithful.

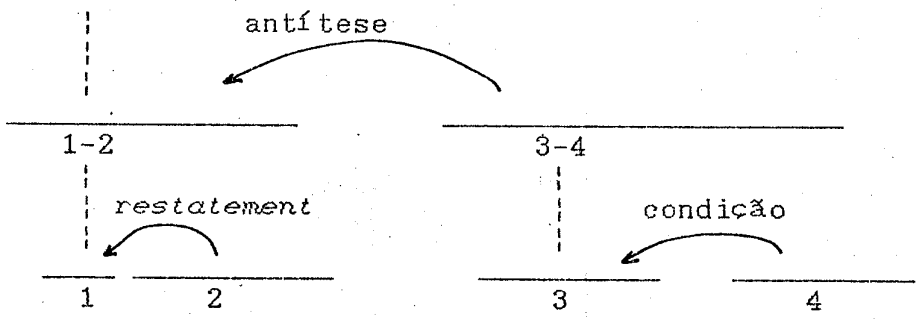
Suppose the user knows the action - homicide, and the crime location -Copacabana.

U.Q.1.: Was Ana killed by an unknown person ?
S.A. : 1. No.
2. João killed her
3. firing three shots
4. because according to him, she was unfaithful.
5. She was his wife.



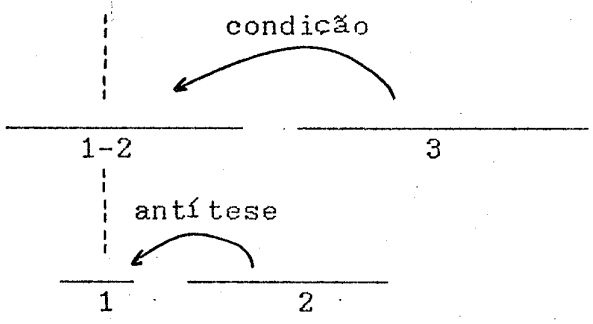

U.Q.2. : Is João's crime "homicidio-culposo" ?

S.A. : 1. No,
2. it isn't.
3. His crime is "homicídio_simple_culposo".
4. since in this type of crime the author's risk is assumed.



U.Q.3. : Is João's punishment five years imprisonment ?

S.A. : 1. No.
2. His punishment is between 6 and 20 years imprisonment
3. because this is the basic punishment for this type of crime.

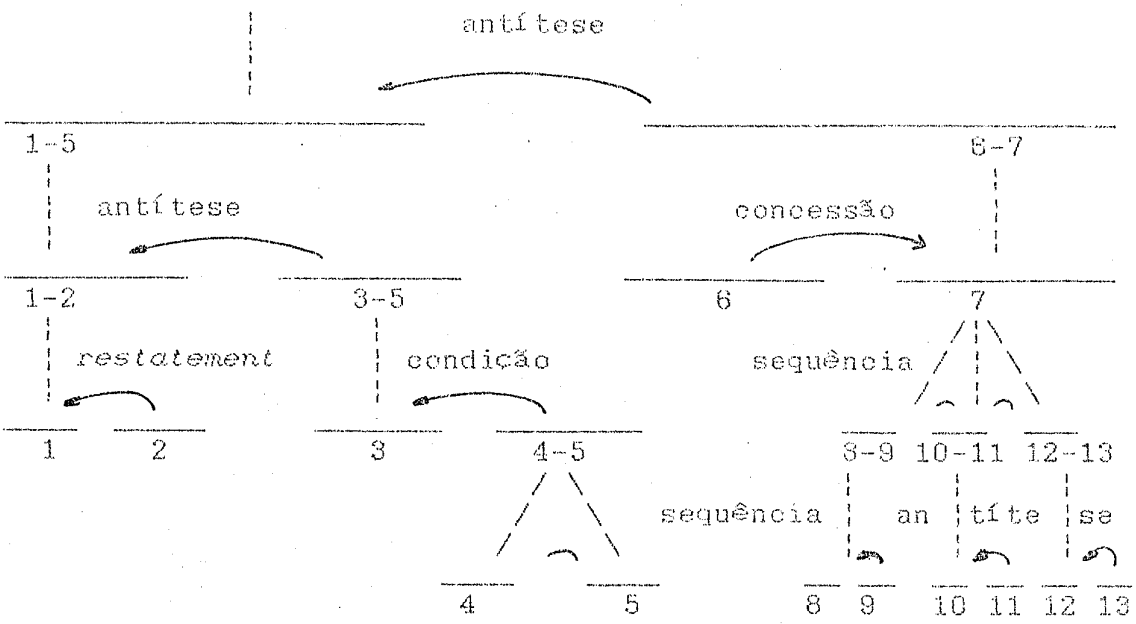



Case 3. Maria, 18 years old, had an abortion because she was pregnant resulting from a rape. With her permission, her doctor performed the abortion, and Maria died.

Suppose the user knows what is an abortion, the agent - Maria, and the agent situation - died.

U.Q.1. : Did Maria commit "provoked-abortion" ?
S.A. : 1. No,
2. she didn't.
3. She commited "aborto_resultante _de_estupro"
4. since the abortion was motivated by a rape-caused pregnancy
5. and was made by a doctor.
6. Although both types of abortion involve a foetus death before parturition,
7. they differ in that
8. a) while the doctor performs the abortion in "aborto_resultante_de_estupro",
9. the mother performs in "aborto_provocado";
10. b) while "aborto_provocado" is a crime,
11. "aborto_resultante_de_estupro" is not,
12. c) the motive for "aborto_resultante_de_estupro" is a rape-caused pregnancy
13. and the motive for the other type is different.

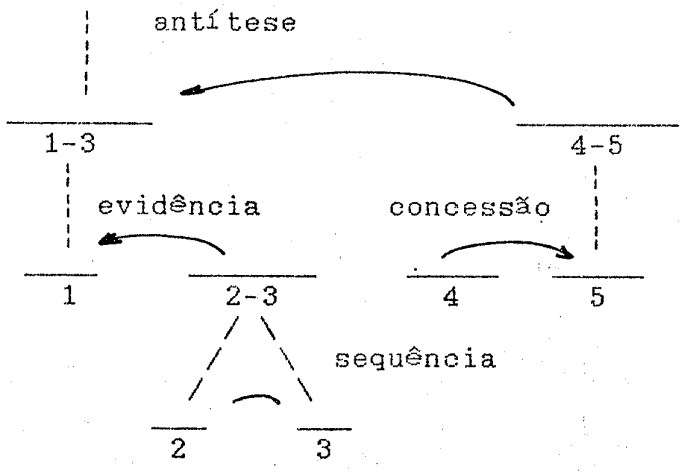
See the figure of the generated structure in the next page.




Case 4. Carlos killed Márcio after a strong discussion at a bar. They were brothers. Carlos was arrested. Márcio was being pursued by the police because he had hurt José a month ago.

Suppose the user knows that Márcio is dead, Carlos and Márcio were brothers and Márcio had hurt José.

U.Q.1. : Was Márcio killed by Carlos ?
S.A. : 1. Sim.
2. The motive was a strong discussion.
3. Consequently, Carlos was arrested.
4. Although had motive,
5. José is not the murder.




Obs.: 2 and 3 are related to 1 via reason and result relation respectively.

## 4. CONCLUSION

We have proposed a deep generation component of a text generator for a knowledge-based system about crimes.

In relation to our initial goals :

**a) Influence of different knowledge sources :**

We have used five different knowledge sources - User Model, Legal Knowledg , Crime Knowledge, Discourse Information and Common Sense Knowledge about the Domain.

All five sources were found to be necessary for providing a cooperative answer. The contribution of each one in determining what to say is :

**Commom Sense Model** : in the treatment of misconceptions and in the provision of additional relevant information.

**User Model** : determining which relevant information would be best left unsaid.

**Domain Model** : provides information about Brazilian Penal Code and about particular criminal events.

**Discourse Model** : although we do not make explicit use of a Discourse Model, we provide the mechanism for utilizing the information that this module would produce (i.e. focus). We also provide a (domain-dependent) mechanism for deducing focus in cases where the Discourse Model fails to find it.

**b) Misconceptions Detection and Justification :**

The proposed method of detecting and justifying misconceptions is strictly based on common-sense rules about the domain and on our very simple User Model. More sophisticated user models could lead to better results. The more the system knows about the user and the domain, the more accurate the misconception treatment will be.

**c) RST as a generation tool :**

We have shown that it is possible to generate a coherent structure for the answer, based on RST, that this structure is cohesive and that it will provide the required information about the types of relations that exist between the elements of the structure. Furthermore, as a welcome side effect, this structure also provides information about how its contents affect the user's (reader's) beliefs, thereby indicating how the User Model should be updated as a result of the response.

At this point, we feel we have achieved our initial goals. However, we would like to add as a caveat that we do not yet have final proof of the amenability of RST as a generation tool. The final proof will only be reached when we are able to produce Portuguese text. This work has just begun.

## 5. REFERENCES


[Allen & Perrault-80] Allen, J.F. & C.R. Perrault, *Analysing Intention in Utterances*. Artificial Intelligence, 15. 1980.

[Appelt-85] Appelt, D.E., *Planning and language generation in problem-solving systems*. Proceedings of the National Computer Conference. Vol. 54. 1985.

[Carberry-87] Carberry, S., *Plan Recognition And Its Use in Understanding Dialogue*. To appear in *User Models in Dialogue System*. Alfred Kobsa and Wolfgang Walhster, eds. Spring Verlag. 1987.

[Chin-87] Chin, D.N., *KNOME : Modeling What the User Knows in UC*. To appear in *User Models in Dialogue System*. Alfred Kobsa and Wolfgang Walhster, eds. Spring Verlag. 1987.

[Grice-75] Grice, H.P., *Logic and Conversation*, in *Syntax and Semantics*. Vol.3. eds. P. Cole and J. Morgan, Academic Press, New York, 1975.

[Mann-87] Mann, W.C., *Text Generation : The Problem of Text Structure*. In *Natural Language Generation Systems*. eds. D. McDonald and L. Bolc, Springer-Verlag, New York, 1987.

[Mann & Thompson-85] Mann, W.C. & S.A. Thompson, *Assertions from Discourse Structure*. Proceedings of the Eleventh Annual Meeting of the Berkeley Linguistics Society. California. February.1985.

[Mann & Thompson-86a] Mann, W.C. & S.A. Thompson, *Rhetorical Structure Theory : Description and Construction of Text Structures*. Presented at the Third International Workshop on Text Generation. Nijmegen, The Netherlands. August. 1986.

[Mann & Thompson-86b] Mann, W.C. & S.A. Thompson, *Relational Propositions in Discourse*. In Discourse Processes 9, (1), January-March 1986.

[Mann & Thompson-87] Mann, W.C. & S.A. Thompson, *Rhetorical Structure Theory : A Theory of Text Organization*. In *The Structure of Discourse*. Livia Polanyi, ed. Norwood, N.J. Alex Publishing Corporation. 1987.

[McCoy-87] McCoy, K.F., *Highlighting a User Model to Respond to Misconceptions*. To appear in *User Models in Dialogue System*. Alfred Kobsa and Wolfgang Walhster, eds. Spring Verlag. 1987.

[McDonald-85] McDonald, D.D., *Surface generation for a variety of applications*. Proceedings of the National Computer Conference. Vol. 54. 1985.

[McDonald & Pustejovsky-85] McDonald, D.D. & J.D. Pustejovsky, *Description-Directed Natural Language Generation*. Proceedings

of the 9th. IJCAI. Vol.2. 1985.

[McDonald & Vaughan & Putejovsky-86] McDonald, D.D. & M.M. Vaughan & J.D. Putejovsky, *Factors contributing to efficiency in natural language generation.*

[McKeown-82] McKeown, K., *Generating Natural Language Text in Response to Question about Database Structure*. Ph.D. dissertation, University of Pennsylvania, 1982.

[McKeown-85] McKeown, K., *The need for text generation.* Proceedings of the National Computer Conference. Vol. 54. 1985.

[McKeown & Wish & Matthews-85] McKeown, K.R. & M. Wish & K. Matthews, *Tailoring Explanations for the User*. Proceedings of the 9th. IJCAI. Vol 2. 1985.

[Paris-87] Paris, C.L., *Tailoring Object Descriptions To a User's Level of Expertise.* To appear in *User Models in Dialogue System.* Alfred Kobsa and Wolfgang Walhster, eds. Spring Verlag. 1987.

[Quilici-87] Quilici, A., *Detecting and Responding to Plan-Oriented Misconceptions.* To appear in *User Models in Dialogue System.* Alfred Kobsa and Wolfgang Walhster, eds. Spring Verlag. 1987.

[Rich-83] Rich, E., *Users are individuals : individualizing user models.* Int. J. Man-Machine Studies, 18. 1983.