PGR: A Cooperative Legal IR System on the Web

Paulo Quaresma and Irene Pimenta Rodrigues
pq@dmat.uevora.pt ipr@dmat.uevora.pt

Departamento de Matemática
Universidade de Évora
7000 Évora
Portugal

CENTRIA/AI Center
Universidade Nova de Lisboa
2825 Monte da Caparica
Portugal

Abstract

In this paper we present a web based legal information retrieval system which is able to support cooperative interactions between users and a legal knowledge base.

The legal knowledge base is composed by four levels of knowledge: Text knowledge; Domain knowledge; Information-retrieval knowledge; and Interaction Knowledge.

The Text Level has knowledge about the words and sequence of words that are in each text of the knowledge base. The Domain level includes knowledge about the text domain such as juridical knowledge (for instance, under which conditions a pension for relevant services may be given to someone). It has rules that encoded the domain knowledge, such as: normally those pensions are attributed to militaries or to civilians (firemen, doctors, nurses, etc.) and the conditions to be fulfilled by them are different. The Information Retrieval Level includes knowledge about what we should expect to find in texts about a subject, for instance that in texts about pensions for relevant services, the pension may be attributed or refused. The Interaction level is responsible for the multimodal interaction management. This includes the ability of the system to infer user intentions and attitudes and to build the representation of the user interaction.

Introduction

In this paper we present a web based legal information retrieval system which is able to support cooperative interactions between users and a legal knowledge base.

The legal knowledge base is composed by four levels of knowledge:

1. Text knowledge;
2. Domain knowledge;
3. Information-retrieval knowledge;
4. Interaction Knowledge.

The text level is composed by the set of documents produced by the Portuguese Attorney General since 1940.

The domain level includes knowledge about specific laws. For instance, we have rules to represent under which conditions a pension for relevant services may be given to someone. We also have a taxonomy of legal concepts constructed by our partner the Portuguese Attorney General Office, which has more than 6000 concepts. As another source of domain knowledge we have built a list of relevant expressions from the documents using statistical methodologies.

The information retrieval level includes knowledge about what we should expect to find in texts about a specific subject. The idea is to find classes (or clusters) of concepts. For instance, in texts about pensions for relevant services, one important feature is the fact the pension may be attributed or refused.

The interaction level is responsible for the interaction management. It is composed by rules that allow the system to infer the interaction structure and the user attitudes (beliefs and intentions).

As we have different levels of knowledge we need different representation techniques and inference tools:

1. Text level – Texts are represented by HTML documents and the SINO search engine (from AustLII) is used to perform inferences. SINO was changed in order to deal with the Portuguese Language. Namely, the new system uses the Portuguese lexicon (more than 900,000 words) to handle morphological errors and to obtain the base queried word. Moreover, we are using a Portuguese tagger to disambiguate the morphological categories of the word.
2. Other levels – Knowledge is represented by logic programs and inference is done through the use of dynamic logic programming semantics (Alferes et al.). Dynamic logic programming defines how a logic program can be updated by other logic programs. In our approach, each event is represented by logic facts, which are used to update the previous program and to obtain a new one.

The interaction manager is basically a logic programming module, which communicates with the user and the different knowledge bases. Communication with the user is done through a web interface using CGI and the Java language.

Cooperation is achieved through the inference of the user attitudes behind the user acts.

These inferred attitudes allow the system to interact with the legal knowledge base in order to refine the user query and to be more helpful in the answer.

For instance, if the user act is to search for documents with expression:

User **-** A1: pensions for relevant services

The system may suggest a way for the user to specify the query:

System **-** A2: Pensions that were granted, refused or both?

The user may select any of the suggestions

User **-** A3: Both.

The system will consider that the user does not want to refine its query any more:

System – A4: There are 129 documents, listed in the right frame.

Another cooperative characteristic is the possibility to manage the context of the interaction and to try to join one action with the previous context:

User – A5: drowning

System – A6: There are 45 documents of the 129 above that match the concept, I've listed them in the usual frame.

In this case it is possible to have ambiguous actions and to choose an incorrect option. So, the system should be able to manage corrections:

User – A7: New context.
User A8: drowning.
System - Q10: There are 347 documents listed in the right frame.

In our system each event (action) is represented by logic programming facts which are used to dynamically update the previous logic program. Using this approach it is possible to reason about past events and to represent non-monotonic behavior rules.

Knowledge Representation Levels

In this section we describe the four knowledge representation levels and we describe how they are integrated in the dynamic logic programming framework.

Our four representation levels are:

1. Textual Level
2. Domain Level
3. Information Retrieval Level
4. Interaction (dialogue level)

The Text Level has knowledge about the words and sequence of words that are in each text of the knowledge base.

The Domain level includes knowledge about the text domain such as juridical knowledge (for instance, under which conditions a pension for relevant services may be given to someone). It has rules that encoded the domain knowledge, such as: normally those pensions are attributed to militaries or to civilians (firemen, doctors, nurses, etc.) and the conditions to be fulfilled by them are different.

The Information Retrieval Level includes knowledge about what we should expect to find in texts about a subject, for instance that in texts about pensions for relevant services, the pension may be attributed or refused.

The Interaction level is responsible for the multimodal interaction management. This includes the ability of the system to infer user intentions and attitudes and to build the representation of the user and system multimodal acts.

Text level

As it was already pointed out the information retrieval system is based on SINO, a text search engine from the AustLII Institute (Greenleaf, Mowbray and King 1997). SINO is a word based text search engine that allows boolean and free text queries.

We have changed SINO in order to be adapted to the Portuguese Language. Namely, the new system uses the Portuguese lexicon (more than 900,000 words) in order to handle morphological errors and to obtain the base queried word.

Domain Level

This knowledge level is built using the Laws describing the requisites for some juridical condition. For instance the law describing the requisites to obtain a pension for relevant services can be encoded by the following rules:

pension(X) <- military(X), action(X,A), behind_duty(A).
pension(X) <- civilian(X), action(X,A), save_life(Y,A), life_at_risk(X,A), not X=Y.

These rules state that:
1. A military may have a pension for relevant services if he has been the agent of an action, and that action was behind is duty.
2. A civilian may have a pension for relevant services if he has been the agent of an action that saves someone life putting his live at risk.

Information Retrieval Level

This level of knowledge is built with rules that can be obtained by processing the text documents looking for keywords that give rise to disjoint sets of documents. By now we obtain these rules using a thesaurus with keywords for text juridical classification.

Example of rules:

document_about(pension(X)) <-document_about(pension_attributed(X)).
document_about(pension(X)) <- document_about(pension_rejected(X)).

false <- document_about(pension_attributed(X)), document_about(pension_rejected(X)).

These rules state that a document with the concept pension either mentions the concept attributed or rejected.

In order to allow the system to obtain the possible explanations of the user queries, we define attributes as abducible predicates. Using this approach it's possible to obtain the set of non-contradictory logic models that explain the user query.

Interaction Level

This knowledge level will represent rules for the interaction at the multimodal acts level. It includes:

1) the rules for inferring the user intentions necessary to generate the system acts;
2) the rules necessary to build the interaction structure in order to obtain the representation of the user acts and its context.

These two sets of rules will be detailed in the next two sections.

Cooperation with the user is achieved due to the existence of the representation and the inference of user intentions. The system tries to infer the user intentions in order help him to find out the set of documents that the user is looking for.

The system helps the user by informing him about the domain knowledge (juridical) and particularities of the texts in the knowledge base. This way the user is guided by the system in the task of refining his queries.

The interaction representation structure supplies the context for the user and system actions. This representation structure takes into account that an utterance may: specify the information contained in a set of previous utterances; or to open a new context, when the user does not intend to continue refining its query and desires to start a new one.

Dynamic logic Programming framework

Dynamic logic programming (Alferes et al. 1998) defines how a logic program can be updated by other logic programs. In fact it defines a semantic for a sequence of logic program updates *P1,…,Pn*. In the update process, each state (*P1,…,Pn*) may represent a different situation, or even a different time point. This feature allows us to model dialogue events by logic programs (composed only by facts) and to use them to update the previous programs. Inertia rules are also guaranteed by the dynamic logic programming semantics. Alferes et al. proposes in their paper a declarative and a procedural semantics for dynamic logic programming.

In order to describe rules in DLP it is possible to use the keywords **before** and **now** to represent the previous and the actual state.

For instance, the speech act inform may be described by “bel(H,P)<-inform(S,H,P)/before”, meaning that after an inform speech act, the hearer starts to believe in the informed proposition (we have assumed cooperative and sincere users).

Suppose we have:

P0 = {int(s,X) <- bel(s,int(u,X)), bel(H,P)<-inform(S,H,P)/before}, meaning that the system intends to do an action X if he believes the user intends to do that action;

P1 = {inform(u,s,int(u,search_documents))}

Then, in state P2 we’ll have:

{int(s,search_documents)}

Inference of user Intentions

In order to be collaborative our system needs to model user attitudes (intentions and beliefs). This task i**s** achieved through the use of logic programming framework rules and the dynamic LP semantics.

The system mental state is represented by an extended logic program that can be decomposed in several modules (see QL95 for a more complete description of these modules):

1 Description of the effects and the pre-conditions of the speech acts in terms of beliefs and intentions;

2 Definition of behavior rules that define how the attitudes are related and how they are transferred between the users and the system (cooperatively).

For instance, the rule which describes the effect of an inform and a request speech act from the point of view of the receptor (assuming cooperative agents) is:

bel(A,bel(B,P)) <- inform(B,A,P)/before.

bel(A,int(B,Action)) <- request(B,A,Action)/before.

In order to represent collaborative behavior it is necessary to model how information is transferred from the different agents:

bel(A,P) <- bel(A,bel(B,P))/now, (not neg bel(A,P))/before.
int(A,Action) <- bel(A,int(B,Action))/now, (not neg int(A,Action))/before.

These two rules allow beliefs and intentions to be transferred between agents if they are not inconsistent with the previous mental state (neg stands for the explicit negation and not stands for the negation by omission).

After each event (for instance a user question) the agents' model (logic program) needs to be updated with the description of the event that occurred. The interaction system recognizes the multimodal act and it constructs the associated speech act (request or inform). The speech act will be used to update the logic program in order to obtain a new model. Using this new model it is possible to obtain the intentions of the system.

Interrogation Context

The Interaction structure (IS) is made of segments that group sets of acts (user and system sentences). The Interaction structure reflects the user intentions, it is built taking into account the user and system intentions. The Interaction segments have precise inheritance rules defining how segments heritage their attributes from the attributes of their multimodal acts.

The Interaction structure is built by recognizing the user intentions and using them in order to enable the system to intervene in the dialogue using pertinent multimodal acts.

In order to define our Interaction structure we first present the Interactions segments and their attribute inheritance rules, and finally we present the rules that enables the system to build the Interaction structures.

Interaction Structure

In this paper we shall present the following 4 segments that enable us to built the interaction structure of our example session 1 presented in the next section:

- Empty , []- an empty structure. It is what we have initially in a new context of interrogation.

- Basic - has 2 arguments:
  Speaker; Action

- New - has 2 arguments:
  Interaction Structure; Interaction Structure

  This Interaction Structure inherits its attributes from their second argument
  ex: New([],basic(User,Q1))

- Specify - has 2 arguments
  Interaction Structure; Interaction Structure

  This Interaction structure inherits their attributes from both Interaction structure
  ex: Specify(Basic(User,[],Q1),Basic(System, Q2))

By now we may consider that the attributes are the representation of the user and system multimodal acts.

Rules to build the Interaction structure

Given act(S1,Speaker) where S1 is the first act representation, the update of the new act in the interaction is:

act(basic(Speaker,S1)).

This fact gives rise to the update of the new IS according to the above rules

```
is(specify(Old_is,Is3))/now <-   is(Old_is)/before,
                                 act(Is3)/now,
                                 possible(specify(Old_is,Is3).

is(new(Old_is, Is))/now <-       is(Old_is)/before,
                                 act(Is)/now.
```

where:

possible(specify(Is,basic(User,S1)))<- search_sino(semantic(specify(Is,basic(User,S1))),Y),not Y=[].
possible(specify(Is,basic(System,S1))).

These rules encode that it is possible for an user act to specify a Interaction structure if the resulting structure gives rise to a SINO query that match one or more documents. A system utterance always specifies the previous Interaction.

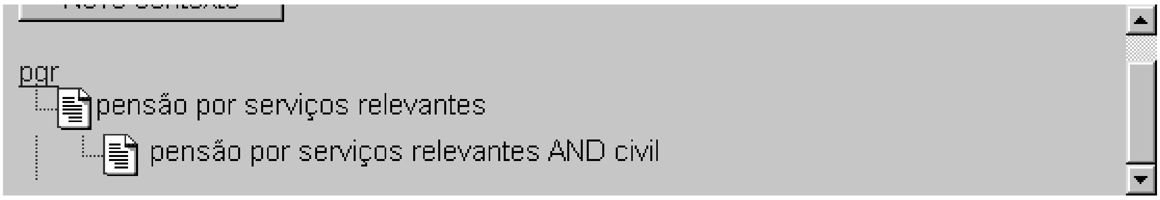
With these rules the system always intends to specify the user query, either by answering it or by asking the user to give more information.

The Interaction structure has preference rules over the segments to build when it is possible to built more than one. The preference are: specify segment and new.

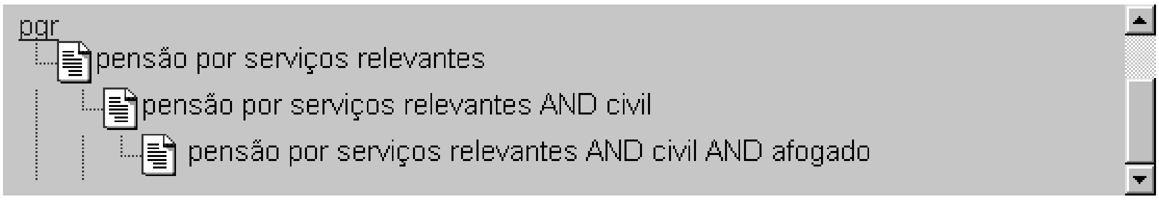
Detailed Example of a Collaborative Interaction

Dialogue 1:

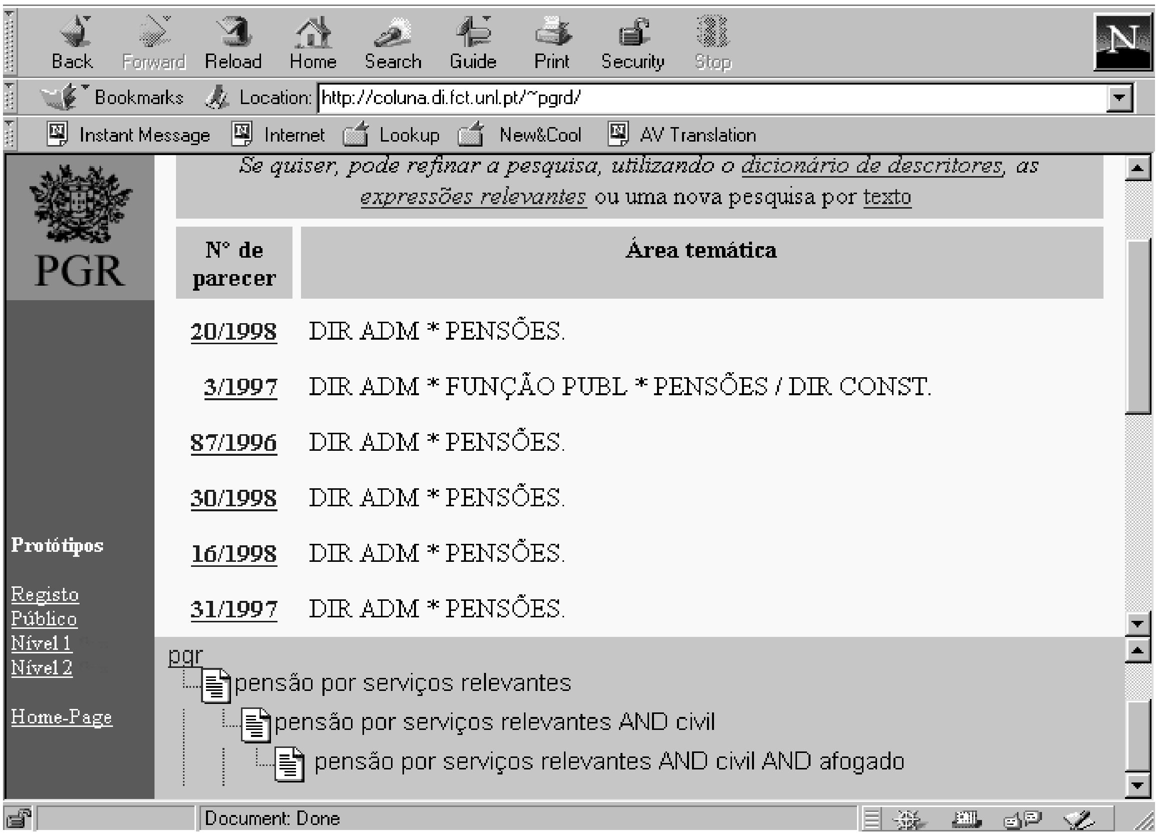
User - A1: pensions for relevant services?
System - A2: Pensions that were given or refused?
User - A3: Both.
System - A4: Pensions for militaries or civilians?
User - A5: Civilians
System - A6: There are 129 documents, listed in the right frame.



User - A7: drowning?



System - A8: Since there are 45 documents of the 129 above that match the concept, I've listed them in the usual frame.



User - A9: New context..
User - A10: drowning.

System - A11: There are 347 documents listed in the right frame.

User - A7': drugs
System - A8': I assume that you want a new query about drugs since there are no documents with the concept drugs and pensions for relevant services for civilians. Do you want to see documents about the use of drugs while driving?
User: - Q10' yes.

System reasoning steps

A1 causes an update of:

sentence(basic(user,[x,y:documents(x),concept(y), y=pension,about(x,y)]))

which gives rise to the update of

IS2=Is(new([],basic(user,[x,y:documents(x), concept(y), y=pension,about(x,y)]))

and

request(u,s,inform(s,u,[document(X),concept(Y),Y=pension, about(X,Y)])).

Which gives rise to the following update

int(s,inform(s,u, [document(X), concept(Y), Y=pension, about(X,Y)])).

Due to the speech act rule for requests in the Interaction knowledge level:
int(S,A) <- request(S,H,A)/before

In order to execute the inform action we must obtain values for the free variables in the semantic representation.

A non collaborative version will simple launch the query: "sino> search Y" at the Text knowledge level.
Our collaborative version will use the Information Retrieval and the Domain level to predict the user goals.

The steps of the reasoning process are:

- Update the IR level with the concepts to be searched: Y (pension)
- Obtain models at IR level: {pension, pension_attributed, military},{pension, pension_refused, military},{pension, pension_attributed, civilian},{pension, pension_refused, civilian},{pension, military},{pension, pension_attributed}, {pension}
- Choose the most suitable model:
- Obtain the predicate list: {pension_attributed, military}
- Ask the user to supply the value of those predicates:
  request(s,u,inform(u,s,[document(Y),concept(Z),Z=military, about(Y,Z)])) e
  request(s,u,inform(u,s,[document(Y),concept(Z),Z=pension_attributed, about(Y,Z)])).

This will cause the system to generate actions A2 and A4.

A2 and A4 will be incorporated in the Interaction structure in a specify segment as well the user actions A3, A5 and A6.

When the user poses question A7.

User - A7: drowning

The system will update:

sentence(basic(user,[x,y:documents(x),y=drown,about(x,y)]))

Which gives rise to the update of:

IS2= is(specify(specify(basic(user,[x,y:documents(x),y=pension,about(x,y)], ...(sentences A2 to A6)),basic(user,[x,y:documents(x),y=drown,about(x,y)])).

Since possible(IS2) is a logic consequence because there are some documents in our text base that are about pensions given to someone that saves another from drowning in the sea.

Which according to our segment inheritance rule gives rise to the following semantic representation:
[x,y,z: documents(x),concept(y) y=pension,about(x,y), ... semantic of A2-A6 ..., concept(z),z=drown,about(x,z)]

having as consequence the update of the following speech act:

request(u,s,inform(s,u,[document(X),concept(Y),Y=pension, about(X,Y), ...A2-A6...., concept(Z), Z=drown, about(X,Z)])).

Since in both knowledge levels this query will have only a model the system will generate the answer A8:

System - A8: Since there are 45 documents of the 129 above that match the concept, I've listed them in the usual frame.

But user may be want to start a new context with A7, so it will utter A9:

User - A9: New context.

The system will recognize this act as a clarification of Interaction segment IS2. And it will start a new context by creating the interaction structure IS3:

IS3= new(IS2, [v,w: documents(v),concept(w) w=drown,about(v,w)])

If instead of A7 the user had uttered A7':

User – A7’: Drugs.

The interaction structure that will be obtained will be a

IS4= new(IS2, [v,w: documents(v),concept(w) w=drug,about(v,w)])

Since the evaluation of sino_search> drug and pension will give an empty set of documents.

And the system will answer A8' that includes the prediction of the user goals with the query A7'.

Conclusions

In this paper we have proposed a system which is able to cooperatively participate in multimodal interactions via the web, namely in information-seeking interactions.

The system uses dynamic logic programming to represent and to reason about events. Four levels of knowledge are described using DLP extended with explicit negation: Interaction, Domain, Information Retrieval and Text.

The interaction level is responsible for the interaction management. The Domain level includes knowledge about the text domain. The Information Retrieval Level includes knowledge about what we should expect to find in texts about a specific subject. Finally the Text Level has knowledge about the words and sequence of words in each text of the knowledge base.

Cooperation is achieved through the inference of user attitudes using the knowledge representation.

References

[AKPT91] James Allen, Henry Kautz, Richard Pelavin, and Josh Tenenberg. Reasoning about Plans. Morgan Kaufman Publishers, Inc., 1991.

[AP96] José J. Alferes and Luís Moniz Pereira. Reasoning with Logic Programming, volume 1111 of Lecture Notes in Artificial Intelligence.Springer, 1996.

[ALPPP98] José J. Alferes, João Leite, Luís Moniz Pereira, H. Przymusinska and T. Przymuzinski. Dynamic Logic Programming, Proceedings of KR'98- Knowledge Representaion, 1998.
[Bra90] Michael Bratman. What is Intention?, in Intentions in Communication. MIT, 1990.
[CL90a] P. Cohen and H. Levesque. Intention is choice with commitment. Artificial Intelligence, 42(3), 1990.
[Car88] Sandra Carberry. Modelling the user's plans and goals. Computational Linguistics, 14(3):23--37, 1988.
[GS86] Barbara Grosz and Candice Sidner. Attention, intention, and the structure of discourse. Computational Linguistics, 12(3):175--204, 1986.
[HM87] S. Hanks and D. McDermott. Nonmonotonic logic and temporal projection. Artificial Intelligence, 33, 1987.
[KR93] Hans Kamp and Uwe Reyle. From Discourse to Logic: An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory. Dordrecht: D. Reidel., 1993.
[KS86] Robert Kowalski and Marek Sergot. A logic-based calculus of events. New Generation Computing, 4:67--95, 1986.
[LA87] D. Litman and J. Allen. A plan recognition model for subdialogues in conversations. Cognitive Science, (11):163--200, 1987.
[LA91] Alex Lascarides and Nicholas Asher. Discourse relations and defeasible knowledge. In Proceedings of the 29th Annual Meeting of ACL, pages 55--62, 1991.
[PR93] J.Pinto and R.Reiter. Temporal reasoning in logic programming: A case for the situation calculus. In D.S. Warren, editor, Proceedings of the 10th ICLP. MIT Press, 1993.
[PP91] F.Pereira and M.Pollack. Incremental interpretation. Artificial Intelligence, 50:40--82, 1991.
[PQ98] Luís Moniz Pereira and Paulo Quaresma. Modeling Agent Interaction in Logic Programming. In Proceedings of the 11th International Conference on Applications of Prolog. Tokyo, Japan, 1998.
[QL95] Paulo Quaresma and José Gabriel Lopes. Unified logic programming approach to the abduction of plans and intentions in information-seeking dialogues. Journal of Logic Programming, (54), 1995.
[QR98] Paulo Quaresma and Irene Pimenta Rodrigues. Keeping Context in Web Interfaces to Legal Text Databases. In Proceedings of the 2nd French-American Conference on AI&LAW, Nice, France, 1998.
[RL92] Irene Pimenta Rodrigues and José GabrielPereira Lopes. Discourse temporal structure. In Proceedings of the COLING'92, 1992.
[RL93] Irene Pimenta Rodrigues and José Gabriel Lopes. Building the text temporal structure. In Progress in Artificial Intelligence: 6th Portuguese Conference on AI. Springer-Verlag, 1993.
[RL97] Irene Pimenta Rodrigues and José Gabriel Lopes. AI5, An Interval Algebra for the temporal relations conveyed by a text. In Mathematical Linguistics II, Eds Carlos Martin-Vide, John Benjamins, 1997.
[Son91] F.Song. A Processing Model for Temporal Analysis and its Aplication to Plan Recognition. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 1991.