

# Efficient Resolution Mechanism for Fuzzy Temporal Constraint Logic

M. A. Cárdenas, I. Navarrete, R. Marín

Dept. de Informática, Inteligencia Artificial y Electrónica. Universidad de Murcia, Spain

E-mail: [marian@dif.um.es](mailto:marian@dif.um.es)

## Abstract

In this paper we describe a module, named Temporal Solving Module (TSM), for answering the queries sent to a data blackboard which must take into account fuzzy temporal constraints among blackboard variables. The TSM receives the queries written in an interaction language that is translated into a language of a reified [13] many-sorted [9] first-order logic, called Fuzzy Temporal Constraint Logic (FTCL). This logic has been formalized in previous works, together with a sound resolution principle for it [3] [4]. Now, we justify the search strategy used in the implementation of the unit resolution. The main results achieved in this work are the simplification criteria and the heuristic evaluation function given for improving the search process to achieve the refutation. First, we use the fuzzy number ordering functions proposed by Marín et al. [12] for obtaining a simplification on queries that imply an exhaustive search in the data blackboard. Secondly, in order to prune the tree, we use a function to check the possibility of obtaining a fuzzy temporal instant different to empty set, in the unification of two imprecise temporal variables.

**Keywords:** *Temporal logics and ontologies, temporal constraint reasoning, temporal languages and architectures, belief and uncertainty in temporal knowledge.*

## 1. Introduction

This work is part of a research project for designing and implementing an intelligent monitoring system of patients, based on temporal reasoning. The domain of this application focus on Intensive Care Units and the system is applied to patients with severe ischemic

cardiopathy, submitted to continuous monitoring of the electrical and mechanical signals of the heart.

A distributed agent architecture has been implemented in the system. Each agent is associated with a blackboard architecture, formed by a temporal data blackboard, a control mechanism and a set of subagents with the same architecture (recursive architecture). Besides, each agent has a blackboard interface with the Temporal Solving Module (TSM) inside. Actually, the conclusions of this paper derive from the design of this module that replies to the temporal queries sent by the system's agents.

Figure 1 shows the structure of the TSM. Temporal blackboard variables are interconnected by means of fuzzy temporal constraints, which are represented in a Fuzzy Temporal Constraint Network or FTCN [1] [2] [14]. An Interaction Language with the TSM is defined, so that any system agent can communicate with the temporal data blackboard. This language, given in section 2, is user friendly and allows queries and asserts with one or more fuzzy temporal constraints, which may either be absolute or relative. In section 3 we review the Fuzzy Temporal Constraint Logic (FTCL) that has been formalized in previous works [3] [4]. We have implemented a module, named ABSTRAE, for the automatic translation of the Interaction Language into the first order language of the FTCL. The main results of this work are shown in section 4, where we show the simplification criteria and the heuristic evaluation function used for improving the search process in the implementation of the unit resolution for FTCL. Finally, an example is given to illustrate all this process.

## 2. Interaction language with the TSM

In this section an Interaction Language with the Temporal Solving Module is defined, in such a way that any system agent can communicate with the temporal

data blackboard. The syntax of this language is specified in figure 2 using BNF format. In a query or an assertion

written in the Interaction Language appears the observable name, the value taken by the attribute and the

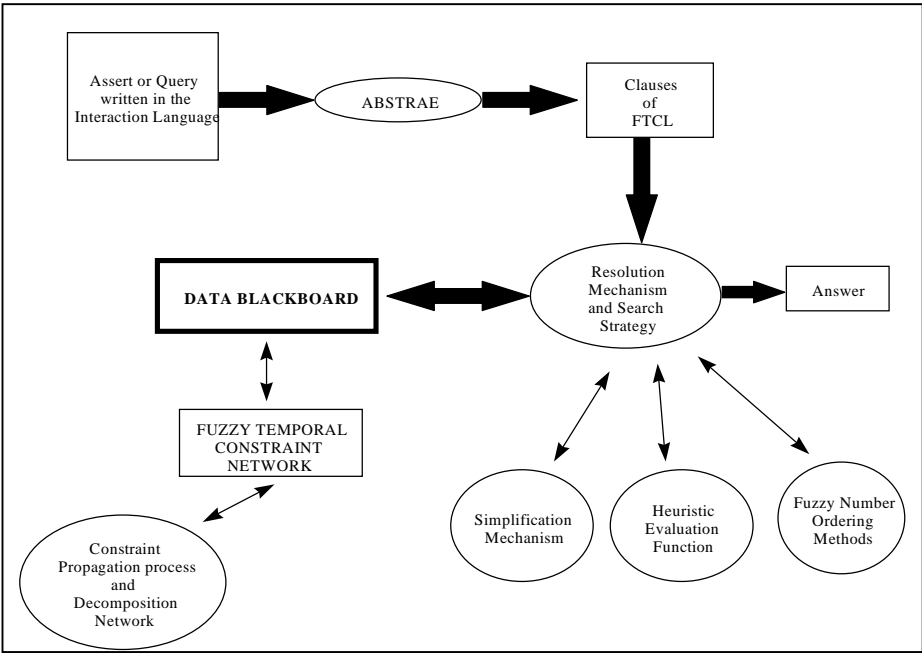


Figure 1 General outline of TSM

<p>           ROOT : ASSERT                    QUERY;            ASSERT : "assert" OBSERVATION;            QUERY : "query" OBSERVATION;            OBSERVATION : OBSERVABLE VALUE TIME;            OBSERVABLE : /* Blackboard observables */;            VALUE : /* Observable values */                    "?";            TIME : INSTANT INSTANTS                    "?";            INSTANTS : /* empty */                    "and" INSTANT INSTANTS;            INSTANT: INS_ABSOLUTE                    INS_RELATIVE;            INS_ABSOLUTE : T                    "[" T "," T "]" ;                    "(" T "," T "," T "," T " )"                    "approximately" T                    DURATION "ago";            T : HOUR DATE                    HOUR                    DATE;            DATE : number                    number "/" number                    number "/" number "number"                    "today"                    "yesterday";            HOUR : number ":"                    number ":" number                    number ":" number ":" number                    "now";         </p>	<p>           DURATION : TQ                    "[" TQ "," TQ "]"                    "(" TQ "," TQ "," TQ "," TQ " )"                    "approximately" TQ;            TQ : number "years" RESTY                    number "months" RESTM                    number "days" RESTD                    number "hours" RESTH                    number "minutes" RESTMIN                    number "seconds";            RESTY : /* empty */                    number "months" RESTM;            RESTM : /* empty */                    number "days" RESTD;            RESTD : /* empty */                    number "hours" RESTH;            RESTH : /* empty */                    number "minutes" RESTMIN;            RESTMIN : /* empty */                    number "seconds";            INS_RELATIVE : DURATION "before" REFERENCE                    "before" REFERENCE                    DURATION "after" REFERENCE                    "after" REFERENCE                    "approximately at the same time of" REFERENCE                    "at the same time of" REFERENCE;            REFERENCE : "last occurrence of event of" OBSERVABLE                    "first occurrence of event of" OBSERVABLE                    "occurrence of maximum value of" OBSERVABLE                    "occurrence of minimum value of" OBSERVABLE;         </p>
---	--

Figure 2 BNF syntax of Interaction Language. The variables are in capital letters and the tokens in small letters (in inverted commas if they are literals)

time of occurrence of this event. This time may be an absolute instant, an instant relative to other or a list of instants (absolute or relative), where all of them may be precise or imprecise.

The purpose of the Interaction Language is allowing an easy and expressive way for the user to interact with the system. Although this is a user friendly interface, the Interaction Language can not take the advantage of the automatic theorem proving that we could have with a formal logic language. That is why we use an intermediate stage for translating the sentences given by the user in the Interaction Language into the first order language of the Fuzzy Temporal Constraint Logic. We have implemented a module, named ABSTRAE, that performs this translation in an automatic and syntax-directed way with the help of the LALR(1) parse generator Yacc [10].

### 3. Fuzzy Temporal Constraint Logic

We review the Fuzzy Temporal Constraint Logic (FTCL) that has been formalized in previous works [3] [4]. It is based on Possibilistic Logic [6] [7] [8]. We give in this section the syntax and informal semantics of this logic. The resolution principle process is briefly described in section 3.1.

Let  $\mathcal{L}$  be a many-sorted language [9].  $\mathcal{L}$  is a first-order language consisting of the following elements: *punctuation marks*, *logical connectives*, *quantifiers*, numerable set of *variables* (that may be observable variables, value variables, instant temporal variables and duration temporal variables), numerable set of *functions* (divided into the same types as the variables) and numerable set of *predicates*. In this latter set, apart from observable predicates, temporal predicates and value predicates, there exists a special one, called *mixed predicate*. Through this predicate we can represent events, in which an observable is associated with a value and with a time.

With the previous elements, and just as in classical logic, we define the notions of *term* and *formula*. There are four types of terms, depending on the type of variable included in them, and the formulas will be heterogeneous, that is to say, they will be formed by predicates and terms of different types.

With this definition of  $\mathcal{L}$ , the syntax of FTCL can be summarized as follows:

Let  $\Omega$  be the set of interpretations associated with  $\mathcal{L}$ . Let  $T$  be the discrete time axis. We consider that  $L=[0,1]^T$  is a lattice of fuzzy subsets of  $T$ . The *fuzzy inclusion* defines an ordering in this lattice. Furthermore, the lattice operators on  $L$  are the *fuzzy intersection*, the

*fuzzy union* and the *fuzzy complementation* [11].  $(L, \cap, \cup)$  is a complete distributive lattice but it is not Boolean since generally  $\tau \cup \bar{\tau} \neq T$  and  $\tau \cap \bar{\tau} \neq \emptyset$ .

Given a clause  $\zeta$  of  $\mathcal{L}$ , in which the instant temporal variable  $p_i$  appears, we propose the following definitions:

**Definition 1** A *temporal possibility distribution* for the instant temporal variable  $p_i$  in  $\mathcal{L}$  is a function,  $\pi_i: \Omega \rightarrow L$ , which will be associated with this temporal variable.

**Definition 2** We define a *context temporal possibility function*  $\tilde{\Pi}_i: \mathcal{L} \rightarrow L$ , induced by each  $\pi_i$ , as

$$\tilde{\Pi}_i(\zeta) = \cup \{ \pi_i(\omega), \omega \text{ is a model of } \zeta \}$$

**Definition 3** We define a *context temporal necessity function*  $\tilde{N}_i: \mathcal{L} \rightarrow L$ , induced by each  $\pi_i$ , as

$$\begin{aligned} \tilde{N}_i(\zeta) &= \overline{\cup \{ \pi_i(\omega), \omega \text{ is a model of } \neg \zeta \}} \\ &= \cap \{ \pi_i(\omega), \omega \text{ is a model of } \neg \zeta \} \end{aligned}$$

**Definition 4** We define a *fuzzy temporal constraint clause*  $\mathcal{C}$ , as a pair  $(\zeta, \tilde{\mathcal{N}})$ , where  $\zeta$  is a Horn clause of  $\mathcal{L}$ , which consists of  $k$  predicates in which  $n$  temporal variables appear;  $\tilde{\mathcal{N}}$  is a tuple  $(\tilde{\tau}_1, \dots, \tilde{\tau}_n)$ , where each  $\tilde{\tau}_i$  is a subset of  $[0,1]^T$ , or a fuzzy subset of  $T$ , being  $T$  the discrete time axis.  $\tilde{\mathcal{N}}$  tells us that each instant temporal variable  $p_i$  occurs in a time  $\tau_i \in \tilde{N}_i(\zeta)$ .

**Definition 5** We define the *temporal truth degree* of a predicate  $\rho_k$ , in which the instant temporal variables  $p_i$  and  $p_j$  are involved, and we will denote it by means of  $\tilde{V}_k^{ij}$ , as a function  $\tilde{V}_k^{ij}: \Omega \rightarrow L$  defined as:  $T$  if  $\rho_k$  is completely true,  $\emptyset$  if  $\rho_k$  is completely false, and a fuzzy subset  $\tilde{v}_k^{ij}$  of  $T$  otherwise.

$\tilde{v}_k^{ij}$  will measure the fuzzy set of instants to which a predicate in question can certainly be referred, bearing in mind the relationship with the instant temporal variables  $p_i$  and  $p_j$ .

We shall associate to each fuzzy temporal constraint clause  $\mathcal{C}$ , a tuple  $\tilde{\mathbf{V}} = (\tilde{V}_1, \dots, \tilde{V}_K)$  of true values of each predicate; and a tuple  $N = (\alpha_1, \dots, \alpha_n)$ , which gives us lower bounds for the certainty value that each one of the temporal variables  $p_i$  of  $\zeta$  takes its value at given times with  $\tau_i$ , and it is going to depend on  $\tilde{N}_i$  and  $\tilde{V}_k^{ij}$ .

In [3] a full description of the FTCL-semantics can

be found, including the proofs of the deduction and refutation theorems.

### 3.1 Resolution principle

The objective of this section is to summarize the FTCL-resolution principle [3] [4]. We are going to use two notions of the necessity function for each one of the instant temporal variables  $p_i$ : the context temporal necessity function  $\tilde{N}_i$ , already defined before, and the conventional necessity function, denoted with  $N_i$ . There may be various temporal variables in a clause, thus it is important to introduce a necessity value associated with each one of them. This tells us the *uncertainty*, which will be associated with each one of the temporal variables in a clause. On the other hand, we consider that in the clause there may be fuzzy or *imprecise* predicates, which in our case are limited to the temporal predicates (for example, *before*). Thus, a temporal truth degree,  $\tau_k^{ij}$ , is associated with each predicate, being  $p_i$  and  $p_j$  the temporal variables involved in the fuzzy temporal predicate, as defined before.

The process of refutation will be as follows:

1. In each clause, a lower bound of  $\alpha_i$  and another of  $\tau_i$  is assigned to each one of the temporal variables  $p_i$ . The first ones is set to 1, and the others are set to the universal constraint  $T$ . When, instead of a variable, we have a constant, it will be started at its value, if it is given as a trapezoidal fuzzy number, or at  $L_{0k}$ , if it is given at a node  $P_k$  of the temporal constraint network.
2. In the unification process, for each  $\tau_i$ , the following formula will be applied:  

$$\tau_i = \tau_i \cap \tau_i' \cap \tau_k^{ji} \cap \tau_k^{ji},$$
being  $\tau_i'$  the value of context temporal certainty which is associated with the unified variable or unified constant with  $p_i$ ,  $k$  the index of predicate which is being manipulated in this step of resolution and  $\tau_k^{ij} = \tau_i \oplus \tau_k^{ij}$ .
3. Each time we make an intersection with the  $\tau_i$ , which belong to the predicate  $\rho_k$ , we update the  $\alpha_i$  of this variable according to the following formula:  $\alpha_i = \min(\alpha_i, \alpha_i', \text{cm}_i)$ , being

$$\text{cm}_i = \begin{cases} 1 - \sup_{t \in T} \inf \left[ 1 - \mu_{\tilde{\tau}_i}(t), \mu_{\tilde{\tau}_i'}(t) \right] & \text{if } \rho_k \text{ is not temporal} \\ 1 - \sup_{t \in T} \inf \left[ 1 - \mu_{\tilde{\tau}_k^{ji}}(t), \mu_{\tilde{\tau}_k^{ji}}(t) \right] & \text{if } \rho_k \text{ is temporal} \end{cases}$$

If  $\rho_k$  is temporal, we suppose that the instantiated variables  $p_i$  and  $p_j$  appear in  $\rho_k$ . On the other hand,

$\mu_{\tilde{\tau}_i}(t), \mu_{\tilde{\tau}_i'}(t), \mu_{\tilde{\tau}_k^{ji}}(t)$  y  $\mu_{\tilde{\tau}_k^{ji}}(t)$  are the membership functions of each one of these fuzzy sets.

4. Finally, with all the  $\alpha_i$ , we calculate a limit  $\alpha$  for the conventional necessity value according to the expression  $\alpha = \min_i \alpha_i$ .

The soundness of refutation by resolution of FTCL is showed in [3] and [4]. Furthermore, in the second paper, the previous resolution principles proposed in [6] and [7], are generalized.

### 4. Resolution optimization in the search process

In this section we show the simplification criteria and the heuristic evaluation function used for improving the search process in the implementation of the unit resolution for FTCL. First, we use the fuzzy number ordering functions proposed by Marín et al. [12] for obtaining a simplification on queries that imply an exhaustive search in the data blackboard. Secondly, in order to prune the tree, we use a compatibility function to check the possibility of obtaining a fuzzy temporal instant different to empty set, in the unification of two imprecise temporal variables.

We use the *unit resolution* in the search strategy. Each unit clause has just one literal. If one of the parents of a resolvent is a unit clause and the other a clause  $C$ , this resolvent must contain less literals than  $C$  and this way the resolvent is closer to the empty clause. Therefore, a trivial heuristic is using only unit resolutions in the refutation process. In order to make this process complete [5], we use a slightly restricted form of clauses with at most one positive literal which are the Horn clauses. In our case using only Horn clauses does not reduce the expressivity power of the language.

So the strategy will be complete, as duly, and moreover we will prune the tree in such a way that the pruned branches would not lead to any possible refutation.

For the sake of simplicity, in sections 4.1, 4.2 y 5, we omit the  $\tilde{\mathcal{N}}$  tuple of clauses.

## 4.1 Simplification criteria

There are three kind of queries that the TSM can support:

1. If one asks about the *time of occurrence* of an event then the query is translated into a unit clause in this way

$$observation(OBS1, VALUE1, p_a),$$

where *OBS1* is an observable constant, *VALUE1* is a value constant, and *p<sub>a</sub>* is an instant temporal variable.

No simplification strategy is applied in this case due to the simplicity of the query.

2. If one asks about the *value* taken by an observable of the blackboard at a certain time, three cases are considered:
  - a) The time is *absolute*. In this case we have again a unit clause like this:  $observation(OBS1, v_a, TIME1)$ , where *OBS1* is an observable constant, *TIME1* is an instant temporal constant, and *v<sub>a</sub>* is a value variable.
  - b) The time is *relative* to an instant. Here we have to apply some simplifications and this will be discuss later.
  - c) The query in the Interaction Language has *multiple temporal labels* linked with the token "and" (see figure 2). In this case the size of the query is bigger but its complexity is the same as in the two previous cases.
3. If one asks if an attribute value of an observable happened at certain time, then the query is similar to the previous queries taken into account that now all variables are instantiated.

Next we describe the mechanism of simplification and we focus on the case 2.b) above, since the other cases where relative temporal variables appear can be resolved in the same manner. A query of this form written in the Interaction Language has the structure

$$query\ obs1\ ?\ duration\ relation\ reference \quad (1)$$

where *obs1* denotes an observable of the blackboard, ? means that a value variable is being consulted, *duration* is a time extent, *relation* can be one of the following expressions: *BEFORE*, *AFTER*, *APPROXIMATELY AT THE SAME TIME OF* or *AT THE SAME TIME OF*, and *reference* may be: *LAST*

*OCCURRENCE OF EVENT OF obs2*, *FIRST OCCURRENCE OF EVENT OF obs2*, *OCCURRENCE OF MAXIMUM VALUE OF obs2* or *OCCURRENCE OF MINIMUM VALUE OF obs2*.

The query (1) can be expressed in  $\mathcal{L}$  as

$$\begin{aligned} &\exists_p p_g, p_c \exists_v v_a, v_b (observation(obs1, v_a, p_g) \wedge \\ &observation(obs2, v_b, p_c) \wedge \\ &relation(duration, p_g, p_c) \wedge \\ &\forall_p p_e \forall_v v_d (observation(obs2, v_d, p_e) \wedge \\ &\neg(Q \Rightarrow P)), \end{aligned}$$

where *P* is one of the following expressions: *after*(*d<sub>f</sub>*, *p<sub>c</sub>*, *p<sub>e</sub>*) if reference is *LAST OCCURRENCE OF EVENT OF obs2*, *before*(*d<sub>f</sub>*, *p<sub>c</sub>*, *p<sub>e</sub>*) if reference is *FIRST OCCURRENCE OF EVENT OF obs2*,  $\geq(v_b, v_d)$  if reference is *OCCURRENCE OF MAXIMUM VALUE OF obs2* or  $\leq(v_b, v_d)$  if reference is *OCCURRENCE OF MINIMUM VALUE OF obs2*. Moreover *Q* is the expression *at\_the\_same\_time\_of*(*p<sub>e</sub>*, *p<sub>c</sub>*) if *P* is *after*(*d<sub>f</sub>*, *p<sub>c</sub>*, *p<sub>e</sub>*) or *before*(*d<sub>f</sub>*, *p<sub>c</sub>*, *p<sub>e</sub>*) and the expression  $\neg(v_b, v_d)$  if *P* is  $\geq(v_b, v_d)$  or  $\leq(v_b, v_d)$ .

Therefore, the negation of the query (1) is translated into three clauses of the FTCL language,

$$\begin{aligned} &\neg observation(obs1, v_a, p_g) \vee \\ &\neg observation(obs2, v_b, p_c) \vee \\ &\neg relation(duration, p_g, p_c) \vee \\ &observation(obs2, g(v_a, v_b), f(p_g, p_c)), \\ &\neg observation(obs1, v_a, p_g) \vee \\ &\neg observation(obs2, v_b, p_c) \vee \\ &\neg relation(duration, p_g, p_c) \vee \\ &\neg Q, \\ &\neg observation(obs1, v_a, p_g) \vee \\ &\neg observation(obs2, v_b, p_c) \vee \\ &\neg relation(duration, p_g, p_c) \vee \\ &\neg P, \end{aligned} \quad (2)$$

where *P* is one of the following expressions: *after*(*d<sub>f</sub>*, *p<sub>c</sub>*, *f*(*p<sub>g</sub>*, *p<sub>c</sub>*)) if reference is *LAST OCCURRENCE OF EVENT OF obs2*, *before*((*d<sub>f</sub>*, *p<sub>c</sub>*, *f*(*p<sub>g</sub>*, *p<sub>c</sub>*))) if reference is *FIRST OCCURRENCE OF EVENT OF obs2*,  $\geq(v_b, g(v_a, v_b))$  if reference is *OCCURRENCE OF MAXIMUM VALUE OF obs2* or  $\leq(v_b, g(v_a, v_b))$  if reference is *OCCURRENCE OF MINIMUM VALUE OF obs2*. Moreover, *Q* is the expression *at\_the\_same\_time\_of*(*f*(*p<sub>g</sub>*, *p<sub>c</sub>*), *p<sub>c</sub>*) if *P* is *after*(*d<sub>f</sub>*, *p<sub>c</sub>*, *f*(*p<sub>g</sub>*, *p<sub>c</sub>*)) or *before*((*d<sub>f</sub>*, *p<sub>c</sub>*, *f*(*p<sub>g</sub>*, *p<sub>c</sub>*))), and the expression  $\neg(v_b, g(v_a, v_b))$  if *P* is  $\geq(v_b, g(v_a, v_b))$  or  $\leq(v_b, g(v_a, v_b))$ . Furthermore, *f* and *g* are Skolem functions introduced when existential quantifiers are eliminated. Realize that in many-sorted logic like this, the quantifiers (existential or universal) are specific for each kind of variables.

In order to make the search efficient it is convenient to locate the reference time  $p_c$  first, since we must look for the last, first, maximum or minimum occurrence of a value in the blackboard. Next we find  $p_g$  using relation. This is better than if one first finds consistent values for  $p_c$  and  $p_g$  and after checks, for any pair of these values, if  $p_c$  is really the reference required. Moreover, if we assign suitable semantic interpretations to  $f$  and  $g$  then the set of the former clauses becomes shorter and the search process in the blackboard is improved considerably. Now we show how to define  $f$  and  $g$ .

Let  $f$  be above Skolem function and  $p_c$  the second variable of the domain of this function. With the following definitions we give a semantic interpretation to the function  $f$ .

**Definition 10** If  $P$  is  $after(d_f, p_c, f(p_g, p_c))$  then  $f$  returns the last fuzzy temporal instant of occurrence of an event of  $obs2$ .

**Definition 11** If  $P$  is  $before(d_f, p_c, f(p_g, p_c))$  then  $f$  returns the first fuzzy temporal instant of occurrence of an event of  $obs2$ .

Let  $g$  be the other Skolem function and  $v_b$  the second variable of the domain of this function. The following definitions give us a semantic interpretation to the function  $g$ .

**Definition 12** If  $P$  is  $\geq(v_b, g(v_a, v_b))$  then  $g$  returns the maximum value of the observable  $obs2$ .

**Definition 13** If  $P$  is  $\leq(v_b, g(v_a, v_b))$  then  $g$  returns the minimum value of the observable  $obs2$ .

**Corollary 14** With the former definitions of the Skolem functions  $f$  and  $g$ , the third clause of the set (2) can be eliminated.

It is worth noting that the above discussion would not have any sense without efficient procedures for computing functions  $f$  and  $g$ . Efficient fuzzy number ordering methods are required in order to compute the function  $f$ . A wide study about this ordering functions with good results can be found in [12]. The computing of the function  $g$  is a trivial implementation problem which we are not going to deal with.

## 4.2 Heuristic evaluation function

In our resolution process, the resolvents are taking necessity values in the way explained in section 3. However these values do not provide a criterion to prune a branch, since one can has a null necessity although there may exist a possible refutation. So, if we use the necessity functions the search process would be incomplete. Instead, we base our heuristic evaluation function on the function

$$\Pi(p, p') = \sup_{t \in T} \mu_{P \cap P'}(u) = \sup_{u \in U} \min [\mu_P(u), \mu_{P'}(u)]$$

defined by Zadeh [15] to show to what extent a value compatible with  $P$  and  $P'$  can be found.

We call this function  $h$  whose domain is the set of clauses (resolvents, to be precise) and for every clause  $\zeta$ ,  $h$  is defined as

$$h(\zeta) = \begin{cases} \sup_{t \in T} \inf \left[ \mu_{\tilde{\tau}_i}(t), \mu_{\tilde{\tau}_i'}(t) \right] & \text{if } \rho_k \text{ is not temporal} \\ \sup_{t \in T} \inf \left[ \mu_{\tilde{v}_k^{ji}}(t), \mu_{\tilde{v}_k^{ji}'}(t) \right] & \text{if } \rho_k \text{ is temporal} \end{cases}$$

$\rho_k$  is the predicate used in the unification. In both cases, variables  $p_i$ ,  $p_j$  appears in the predicate and  $\mu_{\tilde{\tau}_i}(t), \mu_{\tilde{\tau}_i'}(t), \mu_{\tilde{v}_k^{ji}}(t)$  y  $\mu_{\tilde{v}_k^{ji}'}(t)$  are the membership functions to each one of the fuzzy sets defined in section 3.

The objective is to evaluate, whenever two temporal variables are unified, the possibility of finding some precise instant of the time axis that belongs to both variables. If this possibility is null then the refutation can not be reached with the pruned branch. In this way, the completeness of the resolution process is maintained. We could relax the condition of null possibility to have a certain limit of possibility so that beneath this limit the search is stopped. This would increase the efficiency but the completeness is lost, on the other hand. The theoretic completeness is studied in [3].

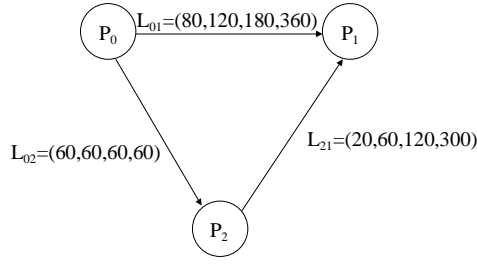
## 5. Example

We are going to describe an example belonging to an application on the domain of Intensive Care Units. The observations come from a patient with severe ischemic cardiopathy, submitted to continuous monitoring of the electrical and mechanical signals of the heart.

We want to know if there was relative low blood pressure (RLBP) approximately 90 minutes after the administration of oral nitrates (OR\_NIT), supposed that in the blackboard we have just the following facts:

*observation*(RLBP, YES,  $P_1$ )  
*observation*(RLBP, YES,  $P_0$ )  
*observation*(OR\_NIT, YES,  $P_2$ ),

being  $P_0$ ,  $P_1$  and  $P_2$  the nodes of the fuzzy temporal constraint network of figure 3.



**Figure 3** Fuzzy Temporal Constraint Network of example

On the other hand, the query is expressed by means of these three clauses in FTCL language:

$\neg \text{observation}(\text{RLBP}, v_a, p_g) \vee$   
 $\neg \text{observation}(\text{OR\_NIT}, \text{YES}, p_c) \vee$   
 $\neg \text{after}(\text{approximately}(90\text{MINUTES}), p_g, p_c) \vee$   
 $\text{observation}(\text{OR\_NIT}, g(v_a, \text{YES}), f(p_g, p_c)),$   
 $\neg \text{observation}(\text{RLBP}, v_a, p_g) \vee$   
 $\neg \text{observation}(\text{OR\_NIT}, \text{YES}, p_c) \vee$   
 $\neg \text{after}(\text{approximately}(90\text{MINUTES}), p_g, p_c) \vee$   
 $\neg \text{at\_the\_same\_time\_off}(f(p_g, p_c), p_c),$   
 $\neg \text{observation}(\text{RLBP}, v_a, p_g) \vee$   
 $\neg \text{observation}(\text{OR\_NIT}, \text{YES}, p_c) \vee$   
 $\neg \text{after}(\text{approximately}(90\text{MINUTES}), p_g, p_c) \vee$   
 $\neg \text{after}(d_f, p_c, f(p_g, p_c))$

The search tree is shown if figure 4, where we can see the values taken by the variables in each step of the resolution, following the simplification mechanism of section 4.1 and making the cut according to the heuristic

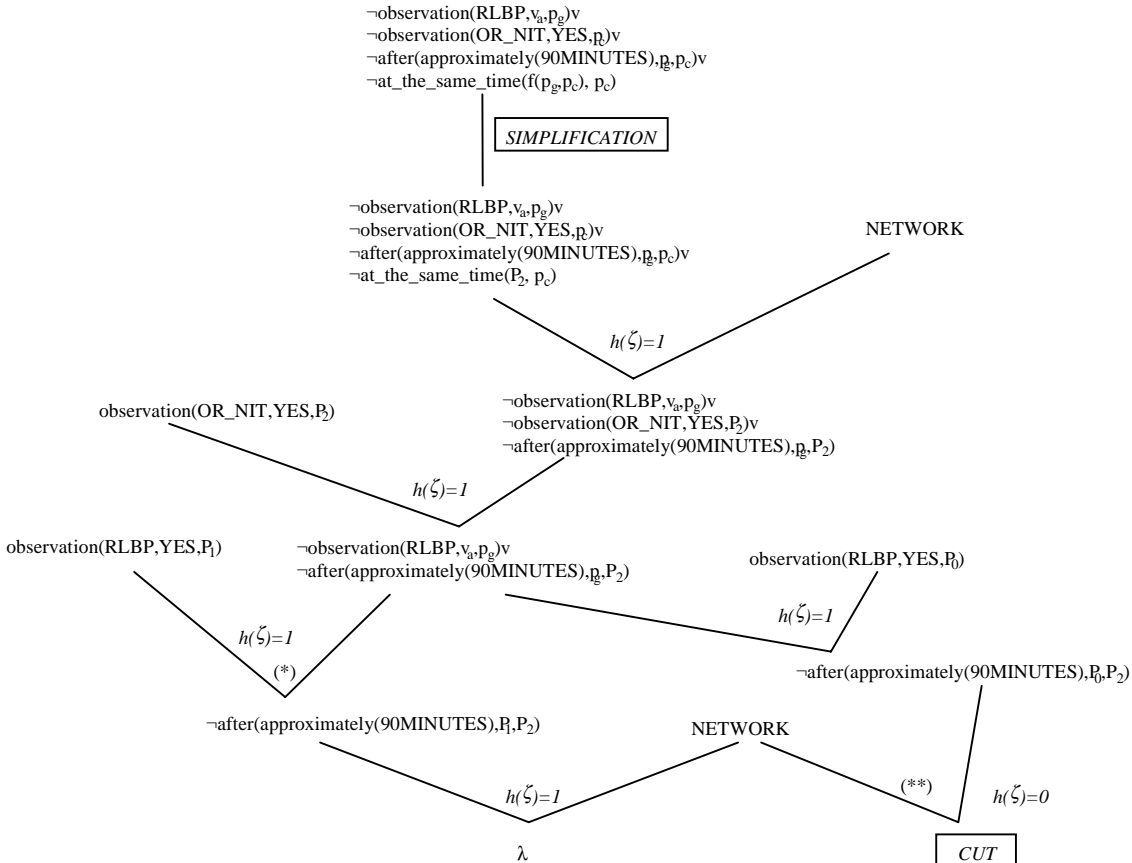
evaluation function of section 4.2. We suppose that the third clause has already been eliminated. Moreover, for the sake of clarity, the only clause used is the second one which leads to a refutation. In this case, with the resolvent (\*) the refutation is achieved, whereas a cut is done in the resolvent (\*\*). This is because the heuristic function  $h$  returns zero in this case since

$$h(\zeta) = \sup_{t \in T} \inf \left[ \mu_{\tilde{v}_k^{20}}(t), \mu_{\tilde{v}_k^{20'}}(t) \right]$$

where  $\tilde{v}_k^{20} = (30, 60, 120, 150)$  and  $\tilde{v}_k^{20'} = (-60, -60, -60, -60)$

## 6. Conclusion

In this work we have described a Temporal Solving Module (TSM) for adding assertions and answering queries in a data blackboard which contains facts that



**Figure 4** Search tree of example

may be related together by means of fuzzy temporal constraints. An Interaction Language with the TSM has been defined to provide an easy but expressive interface. And in order to take the advantage of the automatic theorem proving, we have implemented a parsing module to translate the sentences written in the Interaction Language into Horn clauses of the first order language of a Fuzzy Temporal Constraint Logic (FTCL).

The main results achieved in this work are the simplification criteria and the heuristic evaluation function given for improving the search process to achieve the refutation without losing the completeness of the resolution process.

## Acknowledgements

This work has been partially supported by CICYT-FEDER and EC project 1FD97-0255-C03-01.

## References

- [1] S. Barro, R. Marín, R.P. Otero, R. Ruíz, J. Mira. *On the Handling of time in intelligent monitoring of CCU patients*. Proceedings of the 14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (1992) 871-873.
- [2] S. Barro, R. Marín J. Mira, A.R. Patón. *A model and a language for the fuzzy representation and handling of time*. Fuzzy Sets and Systems, 61 153-175, 1994.
- [3] M.A. Cárdenas, M. Balsa, J.T. Palma, R. Marín. *Lógica de restricciones temporales borrosas: sintaxis y semántica*. Proceedings of 6<sup>th</sup> Iberoamerican Conference on Artificial Intelligence IBERAMIA-98, Lisboa (Portugal), 6 a 9 de Octubre de 1998, Progress in Artificial Intelligence. Helder Coelho ed., pag. 147-158.
- [4] M.A. Cárdenas, R. Marín, I. Navarrete. *Fuzzy Temporal Constraint Logic: A Valid Resolution Principle*. Fuzzy Sets and Systems. In press, 1999.
- [5] R.D. Dowsing, V.J. Rayward-Smith, C.D. Walter, *A first Course in Formal Logic and its Applications in Computer Science*, Blackwell Scientific Publications, 1986.
- [6] D. Dubois, H. Prade. *Resolution principles in possibilistic logic*. International Journal Approx. Reasoning, 4 (1990) 1-21.
- [7] D. Dubois, J. Lang, H. Prade. *Timed possibilistic logic*. Fund. Inf. XV (1991) 211-234.
- [8] D. Dubois, J. Lang, H. Prade. *Possibilistic Logic*. Handbook of Logic in Artificial Intelligence and Logic Programming, D. M. Gabbay, C.J. Hogger, J.A. Robinson, eds., 1994, 3: 439-513.
- [9] H. B. Enderton, *A Mathematical Introduction to Logic*, Academic Press, New York, 1972.
- [10] S.C. Johnson, *Yacc: Yet Another Compiler-Compiler*, Computing Services Technical Report No. 32, Bell Laboratories, Murray Hill, N.J., 1975.
- [11] Kaufmann A., Gupta M., *Introduction to Fuzzy Arithmetic*, Van Nostrand Reinhold, New York, 1985.
- [12] R. Marín, M.A. Cárdenas, M. Balsa, J.L. Sánchez. *Obtainig Solutions in Fuzzy Constraint Networks*. International Journal of Approximate Reasoning 16 3-4, p. 261-288, 1997.
- [13] Y. Shoham, *Reified Temporal Logics: Semantical and Ontological Considerations*, in: Proc. 1986 European Conference on Artificial Intelligence, Society for the Study of Artificial Intelligence and Simulation of Behavior (ALSB), pp. 390-397, 1986.
- [14] L. Vila, L. Godo. *On fuzzy temporal constraint networks*. Mathware and Soft Comput., 1(3), 315-334, 1994.
- [15] Zadeh, L.A., *Fuzzy sets as a basis for a theory of possibility*, Fuzzy Sets and Systems, 1, 3-28, 1978.