ICOT Technical Memorandum: TM-1300

TM-1300

Automatic Generation of Temporal Relations in a Legal Case

by
S. Tojo (MRI) & K. Nitta

June, 1994

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191~5

Institute for New Generation Computer Technology

Automatic Generation of Temporal Relations in a Legal Case

Satoshi Tojo
Mitsubishi Research Institute
tojo@mri.co.jp

Katsumi Nitta
Institute for New Generation
Computer Technology (ICOT)
nitta@icot.or.jp

Abstract

A legal case consists of a number of temporally entangled affairs. To formalize this temporal structure, we need to write down relations between events and states with much effort, while most of the relations may be useless. We propose a method to generate these temporal relations automatically. We define verb types according to their temporal features. According to the type, time intervals are introduced for each affair. We give default assumptions for aligned sequence of affairs, to relate intervals and affairs. A Prolog program is developed to illustrate the feasibility.

1 Introduction

Although logical inference by legal rules (RBR; rule-based reasoning) has played an important role in legal reasoning, legal rules are often too abstract to be applied directly to real problems. Therefore, we have been also required to compare a new problem with precedent cases (CBR; case-based reasoning) [13, 14, 9, 18, 3].

Helic-II system [15] is one of such a hybrid system of RBR and CBR. A case is described by a semantic network like Grebe [3], where temporal relations in affairs in a case are an important part of the network. In Helic-II, Allen's interval logic [1] is used for these temporal relations, however, it was problematic in the following two points. First, combinations of n affairs explodes to the square of n relations although most of them are useless information. Secondly, it has been very hard to write down such an entangled sequence by hand as: an action changes to a state, the state was broken by another action, this action makes another state, and so on.

Our objective is to automate this process. In order to realize this, the method of 'plan script' [19] has been often used, where each verb has a predefined course of subevents. However, to build up plan scripts for so many verbs which can be used in legal matters is another painful job. We propose a method based upon the temporal type of each affair and default assumptions. First, we define a temporal type for each affair, that represents the temporal features of the affair. Therefore, we do not need to attach heavy plan-scripts to each affair. Time intervals are introduced according to these types. Next, temporal relations between these affairs and intervals are generated by given default assumptions. Thus, we are liberated from the tedious work of writing down interval relations.

In the next section, we introduce some fundamental theories in regard to temporal features of affairs, and give formal definitions for them. In the following section, we explain the flow of control as well as default assumptions, and illustrate how temporal relations are made. In the final section, we summarize our contribution and our future plan.

2 Temporal Features of States, Processes, and Events

The simplest way of representing time is to assume a directed line that extends from the eternal past to the eternal future, and to map events on that temporal axis, using time parameter 't.' However this time axis method is too strong for the semantics of natural language, because it is often hard to map events, states, and other temporal ontology on this axis precisely. The interval logic [1] is one method to loosen this strong topology to more coarse-grained time, where time intervals themselves are the temporal ontology instead of time points.

An interval can be interpreted as some time duration where an event occurred. McDermott [7], Kamp [10, 11], Shoham [20] and Kowalski [12] proposed event calculi in different ways, however, they are common in that they can articulate time from a set of events; some of them can define '*an instant*' (point) of time as intersectional intervals of events.

In this paper, we basically observe the interval logic, however we also introduce a pointwise notation just for convenience, so that our logic is not a pure event calculus.

2.1 The Classification of Verbs

As for the classification of verbs with regard to their temporal features, Vendler's one (States/ Activities/ Accomplishments/ Achievements) [21] is hitorically famous. The important distinction here is that several affairs can define its terminative points like Accomplishments and Achievements while others cannot. An affair is called *telic* when its terminative point can be defined, and otherwise it is called *atelic* [4]. This distinction becomes important in relating two affairs, because it decides whether a certain event occurred during the preceding event or after the preceding event. Especially, in legal reasoing, the scope of duration of a deed may affect upon the legal judgement.

Parsons [16] overviewed recent studies on verb classification with this (telic/ atelic) point of view (Fig. 1). Allen's classification (Processes/ Events/ Properties) [1] and McDermott's classification (Fact type/ Event type) [7] can be rather easily mapped into the Parson's chart. Binnick's classification [17] is also almost same with Parson's though more precise.[1] Therefore, in this paper, we observe Parsons' latest classification.

First, we distinguish *stative* verbs from *active* ones. Next, active verbs can be divided into telic ones and atelic ones. We call the former as *eventual* and the latter as *processive*. We call *affair* for the most general class including states and activities. We will redefine these terms with use of intervals in the following subsection.

2.2 Definition of Intervals

In order to formalize the distinction of states, events, and processes, we need to define the following two states (intervals) and one time point [16].

Definition 1 (States of an affair) *An affair consists of the following terms.*

1. *in-progress state:*
   In-progress state is the interval from the inceptive point of progressive activity, to the culminative (terminative) point. This is also called development portion.

2. *time of culmination:*
   The culminative point is the point where the objective of deed is achieved.

[1] Binnick divided first states from non-states, then non-states into telic ones and atelic ones, and then telic ones into development and punctual occurrence, that corresponds to accomplishments and achievements, respectively.

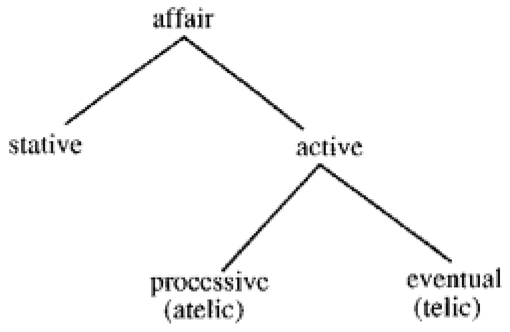

Figure 1: Verb classification

3. holding (target) state:
Target state is the interval from the culminative point, to the point to recover to the original state. The state is also called holding state if it is not instantaneous, viz. the achieved state is held for some time. [2]

We call the in-progress state and the holding (target) state, as **IP-state** and **HL-state**[3] for short respectively, from now on. [4]

In general, an affair begins its IP-state from some beginning time, and ascends to the culminative point. Then the target state is held for its HL-state, as is shown in Fig. 2. In that figure, the IP-state is shown as the light gray interval and the HL-state as the dark gray interval; two intervals are contiguous at the culminative point.

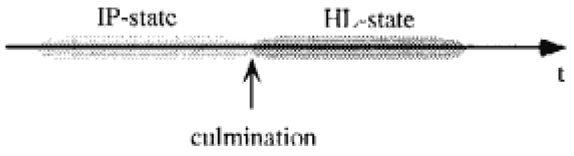

Figure 2: General *affair* type

The variety of temporal features of affairs is given rise to by the following two attributes.

- Length of each interval:
  If IP-state has zero-length then it happened *instantaneous*-ly, and otherwise it was *durative*. If HL-state has zero-length then the state recovered to the original state immediately.

- Both ends of each interval:
  IP-state begins with the inceptive point and finishes with the culminative point, and HL-state begins with the culminative point and finishes with the point to recover to the original state.

[2]Some linguists may be sensitive to the usage between *target* state and *holding* state, however, the discussion is out of the scope of this paper.

[3]As our position is based upon interval logic, we adopt the name of holding state.

[4]Parsons also introduced *resultant* state that is the scope of the influence of the deed, regardless whether the target state is recovered to the original state or not, however we do not mention this state furthermore in this paper because the concept only concerns with the interpretation of perfective.

These points are often unknown, and for such an unknown point we call the end of the interval is *open*.

We give the definitions of verb types in terms of the above interval features.

Definition 2 (Classification) *Verb types are classified in terms of states as follows:*

1. *An affair is* **stative** *iff it is in HL-state. Both ends of HL-states are open.*
2. *An affair is* **processive** *iff it is in IP-state. Both ends of IP-states are open.*
3. *An affair is* **eventual** *iff it culminates. IP-state and HL-state meets at the culminative point. The recovering point of HL-state is open.*

In Fig. 3, we show interval relations of such verb types.

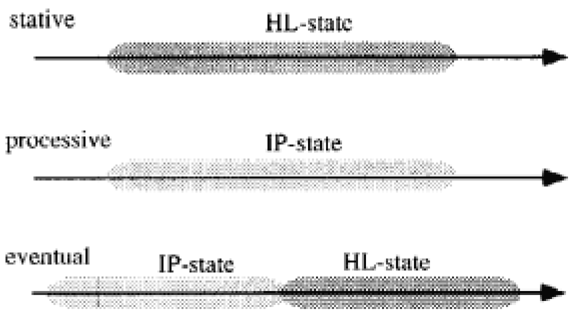

Figure 3: Various verb types

Here, we define functions IP, HL, and $\mathcal{C}$, which retrieve IP-state, HL-state, and the culminative point respectively, given an affair. Actually these functions refer different parts of an affair, so that we call them *reference* functions.

Definition 3 (Reference functions) *Given interval type and point type of time,*

1. $\lambda x.IP(x)$: *affair* $\rightarrow$ *interval*
2. $\lambda x.HL(x)$: *affair* $\rightarrow$ *interval*
3. $\lambda x.\mathcal{C}(x)$: *affair* $\rightarrow$ *point*

where '$\lambda x.f(x) : \alpha \rightarrow \beta$*' means that* f *is a function that bites a variable* x *of type* α*, and that retrieves an object of type* β.

According to the verb types above, reference functions works as below:

a	stative	processive	eventual
$IP(a)$	-	w.d.	w.d.
$HL(a)$	w.d.	-	w.d.
$\mathcal{C}(a)$	-	-	w.d.

where 'w.d.' means well-defined, and '-' means unknown.

2.3 Time - from the situation theoretic point of view

Situation Theory, proposed by Barwise and Perry [2], is a paradigm to represent the *situatedness* of meaning, like as possible world semantics [6], DRT [11], and mental space [8]. We use this situation-theoretic notations because it seems to have richer notions than possible world and DRT and sounder logical foundation than mental space.

2.3.1 General terms

A unit of information is called *infon*, and an infon has the following form:

$$\ll rel, a_1, a_2, \cdots ; p \gg$$

where *rel* is a relation (that corresponds to a predicate of predicate logic), a_i's are parameters for this relation, and p is a polarity which indicates positive/ negative (1/ 0). When a situation s supports an infon σ, we denote:

$$s \models \sigma.$$

2.3.2 Situation vs. spatio-temporal location

The first version of situation theory distinguished the notion of situation from that of spatio-temporal location; the former was an abstract one and the latter was a part of our physical world [2]. There have been some arguments for this distinction thereafter, though it is also possible to regard spatio-temporal location as a kind of situation. In this paper, we would like to regard temporal locations as situations. Namely, we would regard time intervals as situations.

2.3.3 *soa* vs. *coe*

The next source of confusion in situation theory, when we apply it to temporal matters, is the definitions of *soa* (state of affairs) and *coe* (course of events). By definitions, a situation is a *soa*, and a *coe* is a function from a location to a situation; therefore if one location is fixed, a *coe* becomes a *soa* [2]. This implies that only *stative* type relations are allowed as *rel*'s of infons.

Our objective here is to denote various types of verbs as infons, so that the definition becomes a barrier to our further formalization of temporal features.

We introduce a new notion of *coa* (course of affairs) in this paper. A *coa* is a sequence of infons, including affairs of various types. Namely, we admit eventual type and processive type infons. However this new definition, that is a relaxation of the definition of infon, gives rise to a serious new problem; we are now required to give meaning for the supporting relation '$s \models \sigma$' if σ is processive or eventual. We will discuss this matter in the following subsection.

2.3.4 Supporting relation

There is a problem in interpreting the supporting relation if we regard a situation as a spatio-temporal location. Cooper [5] pointed out the problem by the notion of temporal well-/ ill-foundedness as follows. What is the natural definition of supporting relation '$l \models \sigma$', for some time interval l and an infon σ? Suppose that we can define an interval of an affair as $||\sigma||$. If an infon σ is *stative*, then '$l \subset ||\sigma||$' looks proper for the definition of '$l \models \sigma$.' However, on the contrary, if σ is *eventual* then '$l \supset ||\sigma||$' seems suitable for '$l \models \sigma$.'[5]

[5]The problem should be mentioned more precisely with use of the inclusion relation between intervals. Shoham formalized this matter in a more generalized way as upward/ downward hereditary [20].

In order to avoid this problem, we should not mention the length of the interval of an affair. Instead, we just mention the culminative point of an affair. We give the definition as below.

Definition 4 (Supporting relation) $s \models e$ *iff* $s \ni \mathcal{C}(e)$.

Note that only eventual type affairs can come to the right-hand side of the supporting relations because only they can define the culminative points.

3 Generation of Temporal Relations

In the previous section, we introduced the fundamental theory for the classification and temporal features of affairs. In this section, we introduce the flow of processing and default assumptions for the organization of temporal relations.

3.1 Flow of Processing

In this subsection, we summarize the flow of processing that gives temporal relations to affairs in a case. First, we give a sample case as below:

> "**Mary's case:** On a cold winter's day, Mary abandoned her son Tom on the street because she was very poor. Tom was just 4 months old. Jim found Tom crying on the street and started to drive Tom by car to the police station. However, Jim caused an accident on the way to the police station. Tom was injured. Jim thought that Tom had died in the accident and left Tom on the street. Tom froze to death."

We give a situation-theoretic description in accordance with the story above, as the initial stage of the flow of processing.

1. Given *coa* (course of affairs) $\cdots$

$$\begin{array}{ll}\{ & \ll poor, mary; 1 \gg, \\ & \ll abandon, mary, tom; 1 \gg, \\ & \ll find, jim, tom; 1 \gg, \\ & \ll pick\_up, jim, tom; 1 \gg, \\ & \ll driving, jim, tom, to\_police; 1 \gg, \\ & \ll make\_traffic\_accident, jim; 1 \gg, \\ & \ll misunderstand, jim, dead(tom); 1 \gg, \\ & \ll alive, tom; 1 \gg, \\ & \ll leave, jim, tom, on\_the\_road; 1 \gg, \\ & \ll dead, tom; 1 \gg \quad \}\end{array}$$

2. Make virtual ordering $\cdots$

$$\begin{array}{ll} at\ 0 & \ll poor, mary; 1 \gg \\ at\ 10 & \ll abandon, mary, tom; 1 \gg, \\ at\ 20 & \ll find, jim, tom; 1 \gg, \\ at\ 30 & \ll pick\_up, jim, tom; 1 \gg, \\ at\ 40 & \ll driving, jim, tom, to\_police; 1 \gg, \\ at\ 50 & \ll make\_traffic\_accident, jim; 1 \gg, \\ at\ 60 & \ll misunderstand, jim, dead(tom); 1 \gg, \\ at\ 70 & \ll alive, tom; 1 \gg, \\ at\ 80 & \ll leave, jim, tom, on\_the\_road; 1 \gg, \\ at\ 90 & \ll dead, tom; 1 \gg \end{array}$$

3. set up temporal locations $\cdots$

driving	: processive	$\Rightarrow$	IP-state
poor, live, dead	: stative	$\Rightarrow$	HL-state
abandon, find, pick_up, $\cdots$	: eventual	$\Rightarrow$	IP-state, HL-state

4. make support relations $\cdots$

$$\{IP(a_1) \models a_2,\ \ HL(a_1) \models a_3,\ \ HL(a_3) \models a_4, \cdots\}$$

5. and make inter-situation relations.

$$IP(a_1) \| HL(a_1),\ \ IP(a_2) \prec IP(a_3) \prec IP(a_4), \cdots$$

In the final stage, we introduced several inter-situation relations. '$s_1 \| s_2$' means that s_1 and s_2 meets (or, they are contiguous); '$s_1 \prec s_2$' means that s_1 precedes s_2 temporally, sharing no common time. Although we did not introduce the inclusion relation ('$\subset$') explicitly, the supporting relation ('$\models$') plays the role.

3.2 Default Assumptions

In this subsection, we will discuss several default assumptions. In some cases, the adequateness of these assumptions becomes doubtful, so that we must replace them for other information if necessary; this is the reason we say that these assumptions are mere default rules.

Assumption 1 (Disjoint IP-states of eventual affairs) *IP-states of eventual affairs are aligned in the same order as the eventual affairs, sharing no common time.*

Let us look back the preceding example. The two consecutive eventual affairs:

$at\ 10\ \ll abandon \gg$
$at\ 20\ \ll find \gg$

should obey the assumption, and the system produces the following statement.

$$IP(\ll abandon \gg) \prec IP(\ll find \gg).$$

This claims that IP-state of $\ll abandon \gg$ temporally precedes IP-state of $\ll find \gg$.

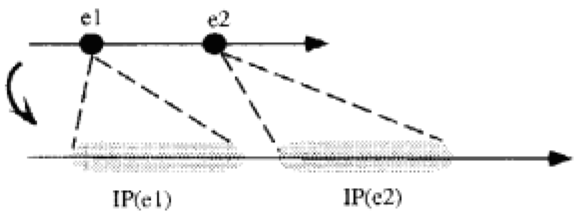

Figure 4: Disjoint IP-states

Assumption 2 (Stative) *HL-states, introduced by state type verbs, support all the culminative points of eventual affairs in the given coa.*

The first infon of the given example, $\ll$ *poor* $\gg$, is stative. Therefore $HL(\ll poor \gg)$ supports all the other eventual infons in the case. According to this assumption, "Mary may be poor all through this case." However let us see other two stative infons $\ll$ *alive* $\gg$ and $\ll$ *dead* $\gg$. The HL-states of these two infons must not support same infons. In such a case, we need to refute this assumption and need to claim other proper support relations.

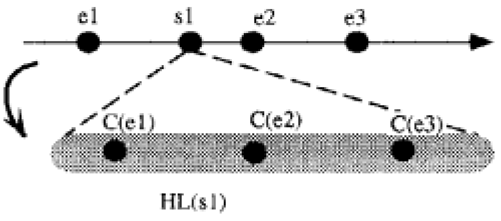

Figure 5: State

Assumption 3 (Processive) *The IP-state, introduced by a processive type verb, supports the culminative point of the following eventual affair.*

In the given example, $\ll$ *driving* $\gg$ is a processive affair. It culminates when "Jim arrives at the police station," however, the deed is still under way. The sequence of:

> *at* 40 $\ll driving \gg$
> *at* 50 $\ll make\_traffic\_accident \gg$

should obey the assumption. viz.:

$$IP(\ll driving \gg) \models \ll make\_traffic\_accident \gg .$$

Different from the preceding *stative* type, this *processive* type supports only one following event. Because the both ends of IP-state are open, *processive* IP-state may extend longer and may support more events. However we set up the default in this way just for practical reason: having surveyed several case descriptions, we judged that the range of IP-state seems not so long.

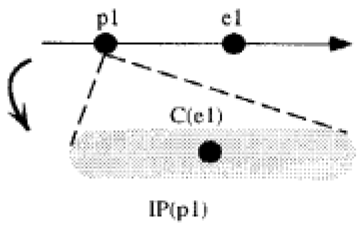

Figure 6: Process

Assumption 4 (Sequence of eventual affairs) *A culminative point of an event is supported by the HL-state introduced by the preceding event.*

Let us see again the two consecutive eventual affairs: $\{\ll abandon \gg, \ll find \gg\}$. The assumption claims that the culminative point of $\ll find \gg$ occurred in HL-state of $\ll abandon \gg$, viz.:

$HL(\ll abandon \gg) \models \ll find \gg$.

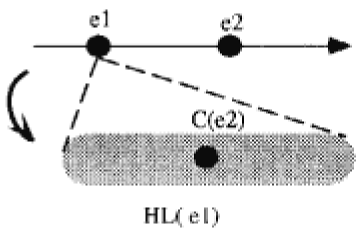

Figure 7: Sequence of events

3.3 Sample program

A inference system based upon the above specification is implemented in IF/Prolog on SUN-4/SPARC-330 station. A part of the program is shown below.

(1) Toplevel function

```
%%% toplevel function

relate_between_affairs(Coa) :-
        add_virtual_time(Coa,10,10,Timed_coa),
        create_event_agents(Timed_coa),
        set_up_primitive_situations(Timed_coa),
        arrange_eventual_ip,
        make_support_relations,
        display_temporal_relations.
```

The first line of the body, `add_virtual_time`, adds virtual time on affairs. After that, the third line, `set_up_primitive_situations`, introduces IP-states and HL-states of each affair. The situation introduction from the virtual order is shown in Fig. 8, where light gray ovals are IP-states and dark gray ones HL-states, on Mary's case. Those two characters in the figure:

po, ab, fi, $\cdots$

are taken from corresponding infons' relation names, and represent:

$\ll poor \gg$, $\ll abandon \gg$, $\ll find \gg$, $\cdots$

respectively.

(2) Assumption 1

Next, we show the contents of the fourth line of the toplevel function, `arrange_eventual_ip`, which is the predicate for Assumption 1.

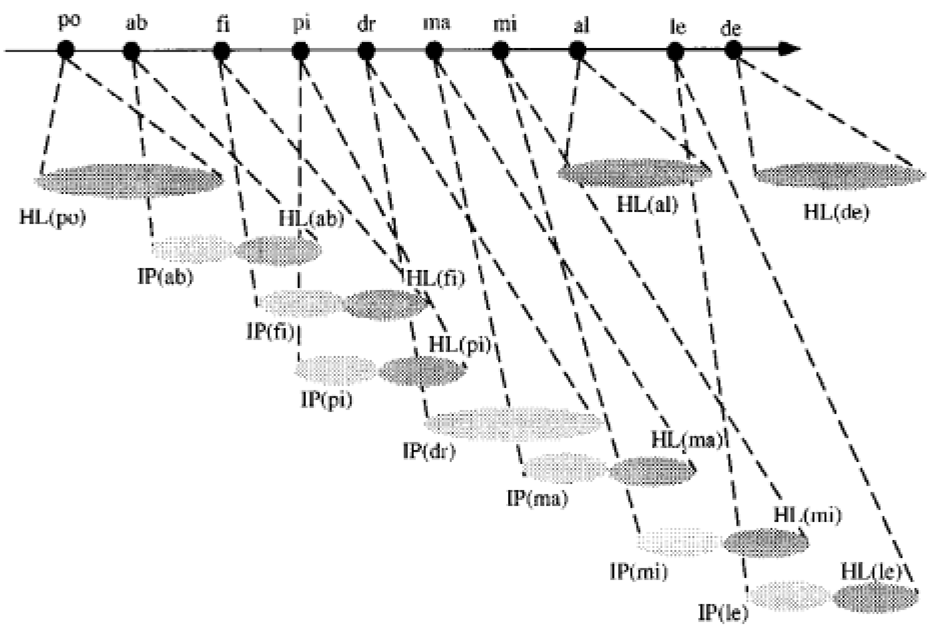

Figure 8: Initial situation setting

```
%% Assumption 1

arrange_eventual_ip :-
        '$field'(world, W),
        filtrate(W,eventual, Eventual_field),
        who_is_first(Eventual_field, First),
        arrange_eventual_ip0(Eventual_field, First).

arrange_eventual_ip0(_,none) :-!.
arrange_eventual_ip0(Eventual_field, Agt1) :-
                make_contiguous_relations(Agt1),
        who_is_next(Agt1, Eventual_field, Agt2),
                make_contiguous_relations(Agt2),
        get_state(Agt1, ip, S1),
        get_state(Agt2, ip, S2),
        relate(prec, S1, S2), !,
        arrange_eventual_ip0(Eventual_field, Agt2).

make_contiguous_relations(Agt) :-
        get_state(Agt, ip, IP),
        get_state(Agt, hl, HL),
        relate(contiguous, IP, HL).
```

In arrange_eventual_ip, if an affair is eventual, its IP-state and HL-state are related as **contiguous**.

(3) Assumption 2-4

The fifth line of the toplevel function, make_support_**relations**, is for other assumptions below:

```
%% support relations

make_support_relations :-
        '$field'(world, W),
        who_is_first(W, First),
        who_is_next(First, W, Next),
        make_support_relations0(W,First,Next).

make_support_relations0(_,Agt1,none) :-
        say_type(Agt1,state),!,
        make_support_relations1(Agt1, state, _, _).
make_support_relations0(_,_,none) :- !.
make_support_relations0(W,Agt1,Agt2) :-
        say_type(Agt1,Type1),
        say_type(Agt2,Type2),
        make_support_relations1(Agt1, Type1, Agt2, Type2), !,
        who_is_next(Agt2,W,Agt3),
        make_support_relations0(W,Agt2,Agt3).

%% Assumption 2

make_support_relations1(Agt, state, _,_) :- !,
        get_state(Agt, hl, H),
        '$field'(world, W),
        filtrate(W, process, Proc_set),
        filtrate(W, eventual, Eventual_set),
        support_all(H, Proc_set),
        support_all(H, Eventual_set).

%% Assumption 3

make_support_relations1(Agt, process, Next, eventual) :- !,
        get_state(Agt, ip, IP),
        support(IP, Next).
make_support_relations1(_,process,_,_) :-!.

%% Assumption 4

make_support_relations1(Agt, eventual, Next, eventual) :- !,
        get_state(Agt, hl, H),
        support(H, Next).
make_support_relations1(_,eventual,_,_) :- !.
```

In the program above, say_type retrieves the type of an affair. make_support_relations0 asks two consecutive affairs their types, and if the first affair is of type state then Assumption 2 is applied, if the first affair is of type process and the second one is of type eventual then Assumption 3 is applied, and if both of them are of type eventual then Assumption 4 is applied.

(4) Output

The bare output produced by the function, display_temporal_relations, that is the bottom line of the toplevel function, on Mary's case, becomes as follows:

```
IF/Prolog Version 4.0.4 SPARC OS4.1 created 14/11/90
Copyright (C) 1984,90 InterFace Computer GmbH

?- [coa,test,flage,temporal].
consult: file coa.pro consulted in 0 sec.
consult: file test.pro consulted in 0 sec.
consult: file flage.pro consulted in 0 sec.
consult: file temporal.pro consulted in 0 sec.

yes
?- relate_between_affairs(mary).
relation(contiguous,ip(leave),hl(leave)) .
relation(prec,ip(misunderstand),ip(leave)) .
relation(contiguous,ip(misunderstand),hl(misunderstand)) .
relation(prec,ip(accident),ip(misunderstand)) .
relation(contiguous,ip(accident),hl(accident)) .
relation(prec,ip(pick_up),ip(accident)) .
relation(contiguous,ip(pick_up),hl(pick_up)) .
relation(prec,ip(find),ip(pick_up)) .
relation(contiguous,ip(find),hl(find)) .
relation(prec,ip(abandon),ip(find)) .
relation(contiguous,ip(abandon),hl(abandon)) .

supporting(hl(dead),abandon) .
supporting(hl(dead),find) .
supporting(hl(dead),pick_up) .
supporting(hl(dead),accident) .
supporting(hl(dead),misunderstand) .
supporting(hl(dead),leave) .
supporting(hl(alive),abandon) .
supporting(hl(alive),find) .
supporting(hl(alive),pick_up) .
supporting(hl(alive),accident) .
supporting(hl(alive),misunderstand) .
supporting(hl(alive),leave) .
supporting(hl(misunderstand),leave) .
supporting(ip(driving),accident) .
supporting(hl(accident),misunderstand) .
supporting(hl(pick_up),accident) .
supporting(hl(find),pick_up) .
supporting(hl(abandon),find) .
supporting(hl(poor),abandon) .
supporting(hl(poor),find) .
supporting(hl(poor),pick_up) .
supporting(hl(poor),accident) .
supporting(hl(poor),misunderstand) .
supporting(hl(poor),leave) .


yes
?-
```

In the result above, both of ≪ *alive* ≫ and ≪ *dead* ≫ support every eventual infons. In this case, we need to refute the assumption and need to remove the supporting relations of ≪ *dead* ≫ manually.

Fig. 9 is the revised result, where dotted arrows show supporting relations, solid arrows show the temporal order between situations, and two contiguous ovals show that they have contiguous relations.

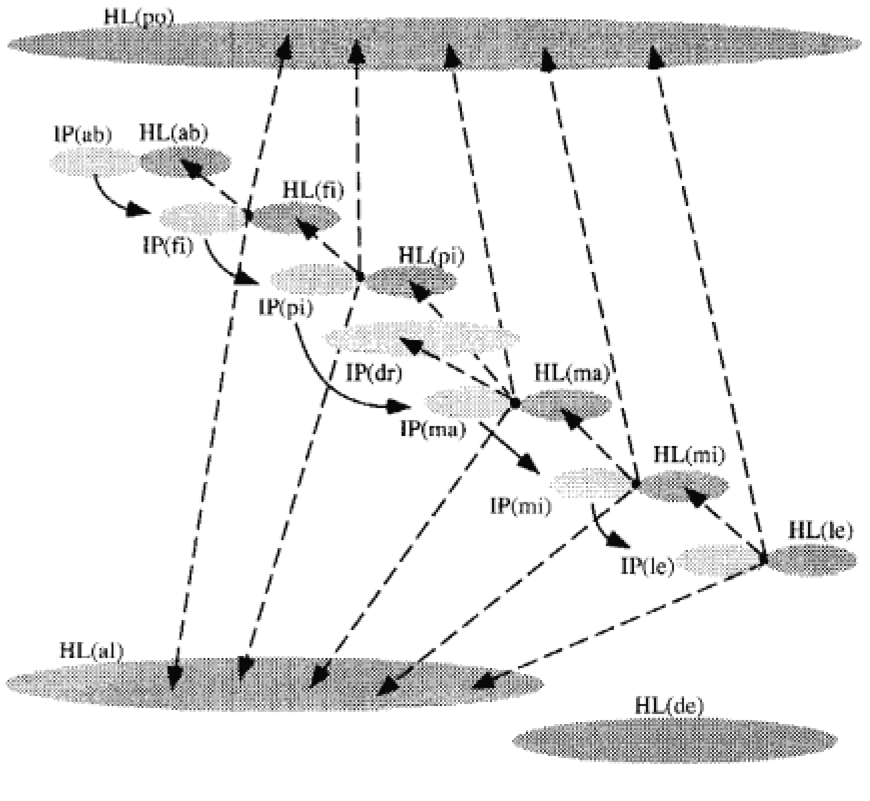

Figure 9: Revised result

4 Discussion

We introduced a theory of verb types, and according to this, we attached each verb its temporal features. With these temporal types, we proposed a method to make temporal relations automatically in a legal case. The system was implemented in Prolog, and we showed a simple example.

However, there are still several problems for our method. First, the verb classification does not strictly depend upon the verb itself, but upon the usage of the verb. Namely, one verb may be used both as a process and as an event. In our current sample system, we attached verb types in the dictionary. However, these type declarations must be refuted freely when a case is encoded to a formal description.

Secondly, default assumptions are not general principles but heuristic rules dependent upon the notion of *coa*, that is a sequence of affairs aligned in a order beforehand. Actually, most of the temporal relations are the complication of the given simple order. Thus, the notion of assumptions lacks sound foundation. However, there is another aspect for this assumptions with regard to *coa*. They also can be our norms for us to write down a course of affairs. Namely, as far as we observe the manner of assumptions for *coa*, we can obtain faithful descriptions of cases. In order to use them in this way, we need to polish up assumptions more, and also need to set up an environment to refute those defaults easily.

By the way, our final end was to use these temporal relations for matching of cases. Namely, we need to compare two sets of temporal relations, each of which represents a case. We are now considering an efficient method to embed a new case to precedent cases, also from situation theoretic point of view.

References

[1] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

[2] J. Barwise and J. Perry. *Situations and Attitudes*. The MIT Press, 1983.

[3] K. Branting. *Integrating rules and precedents for classification and explanation: automating legal analysis*. PhD thesis, University of Texas at Austin, 1990.

[4] B. Comrie. *Aspect*. Cambridge University Press, 1976.

[5] R. Cooper. Tense and discourse location in situation semantics. *Linguistics and Philosophy*, 9(1):17–36, February 1986.

[6] D. Dowty, R. Wall, and S. Peters. *Introduction to Montague Semantics*. D. Reidel, 1981.

[7] D.V.McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155, 1982.

[8] G. Fauconnier. *Espas Mentaux*. Editions de Minuit, 1984.

[9] A. Gardner. *An Artificial Intelligence Approach to Legal Reasoning*. PhD thesis, Stanford University, 1984.

[10] H. Kamp. *Events, Instants, and Temporal References*, pages 376–417. Springer Verlag, 1979. in Semantics from Different Points of View.

[11] H. Kamp. A theory of truth and semantic representation. In J. Groenendijk, T. Jansson, and M. Stockhof, editors, *Methods in the Study of Language Representation*. Math Carter, Amsterdam, 1981.

[12] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 1986.

[13] L.C.McCarty. Computing with prototypes. In *Proc. of the Bar-Ilan Symposium on the Foundations of Artificial Intelligence*, 1989.

[14] L.C.McCarty. Ai and law: How to get there from here. In *Workshop notes of the legal reasoning workshop of the eighth national conference on Artificial Intelligence*, 1990.

[15] K. Nitta, Y. Ohtake, S. Maeda, M. Ono, H. Ohsaki, and K. Sakane. Helic-ii: A legal reasoning system on the parallel inference machine. In *FGCS '92*, pages 1115–1124, 1992.

[16] T. Parsons. *Events in the Semantics of English.* MIT press, 1990.

[17] R.I.Binnick. *Time and the Verb.* Oxford University Press, 1991.

[18] E. Rissland and D. Skalak. Cabaret: rule interpretation in a hybrid architecture. *Man-Machine Studies*, 1991.

[19] R. C. Schank and R. P. Abelson. *Scripts, Plans, Goals, and Understanding.* Lawrence Erlbaum Associates Publishers, 1977.

[20] Y. Shoham. *Reasoning about Change.* The MIT Pres, 1988.

[21] Z. Vendler. Verbs and times. *Philosophical Review*, 66:143–60, 1957.