Temporal Representation for Legal Reasoning *

Lluís Vila†
University of California at Irvine
444 CS, UCI, Irvine CA 92717–3425, USA
`vila@ics.uci.edu`

Hajime Yoshino
Meiji Gakuin University
1-2-37 shirokanedai, minato-ku Tokyo, JAPAN
`yoshino@mh.meijigakuin.ac.jp`

Keywords. Legal knowledge representation, temporal reasoning, legal expert systems.

Abstract

This paper presents a formalism for representing knowledge in the legal domain with an explicit account of time. We do not propose an original contribution on temporal representation, but we present a temporal representation method for *legal reasoning systems* based on some techniques proposed in the area of temporal logics for ai which happen to adequately fulfill some other representational requirements from the legal domain as well.

A central notion of our approach is that of *temporal token*. It allows to make direct reference to specific temporal instances, supports talking about *temporal occurrence* without the technical problems of *reification*, avoids the ontological problems of *temporal types*, allows for a neat separation of the temporal component -which eventually will be processed by an specialized temporal reasoner. Moreover, the notion of temporal token turns out to be very appropriate for expressing references to legal propositions, which is an important issue in representing legal knoweldge.

---

[*] **Acknowledgments.** We thank Eddie Schwalb for interesting discussions. this work has been partially supported by spanish MEC grant EX-94 77909683, by CICYT project *ARREL* (TIC92-0579-C02-01) and by a Japanese Monbusho's grant.

[†] Currently on leave from *Institute for Research in Artificial Intelligence (IIIA), CSIC*, Spain.

1 Introduction

By looking at the knowledge managed in the legal domain, it is rather easy to find references to time. For example, if one is talking about contracts (say between buyer and seller) it is natural to mention the time the contract has been concluded (e.g. "an *escrew*[1] contract between *Eddi* and *Tami* was concluded April the first"), the duration of the contract (e.g. "the contract between *Eddi* and *Tami* last for 3 months"), or the time that the buyer is supposed to pay the seller part (e.g. "*Eddi* has the obligation to pay *Tami* merchandise not more than four weeks later than its delivery time"). Quoting L. Thorne McCarty [9] : "...time and action are both ubiquitous in legal domains. ...".

Surprisingly not much work has been done on providing legal reasoning systems with means to explicitly represent temporal references present in their knowledge. The goal of this paper is to cover this gap by proposing a representation method which supports the expression of temporal references in the legal domain. We envision a system where these expressions will be efficiently processed by an specialized temporal reasoner contributing to the accomplishment of a particular reasoning task.

The structure of the paper is as follows. We first review the representational issues that need to be addressed in order to come up with a temporal representation formalism for legal reasoning. Then, we introduce the idea of temporal token and other key ideas our approach is based on. Next, we define our logical language as a many-sorted first-order logic, presenting syntax and semantics, and we discuss its application to representing legal knoweldge. finally, we outline some extensions to cope with additional specific representational requirements.

2 Representation Issues to Address

A legal reasoning system may involve many representational issues which often appear mixed in real problem solving tasks (see [9] for issues related to *action languages* in the legal domain). In this paper we concentrate on the *representation of time* and *temporal references*, yet we also consider (at basic stage) the *representation and change* and the representation of what we call *propositional references*, since they are interrelated in the way we shall make clear in short.

- **Temporal representation issues.** In general, to define a temporal representation approach one usually must provide a way of expressing two sort of formula [12]:

    1. *Temporal occurrences*: Formula expressing the holding of a certain proposition at a certain time. For example, the statement "an escrew contract between *Eddi* and *Tami* was concluded April the first" requires a sort of formula with related the proposition "escrew contract between *Eddi* and *Tami* is concluded" and the *temporal qualification* "April the first".

---

[1] an *escrew* contract is a temporary contract issued to carry out a complex transaction.

2. *Temporal relations*: formula expressing relations (i.e. constraints) between different time elements. For example, from the statement "b has the obligation to pay a's merchandise not more than four weeks later than its delivery time" we need to specify somehow that the "payment time" and the "delivery time" are at most four weeks apart. There a variety of different temporal relations we may encounter: absolute (e.g. "April the first"), relative (the previous one is an example of that), precise (e.g. "three weeks"), approximate ("between two and three weeks", "four weeks at most").

The definition of the formula above will strongly depend on the underlying *model of time* we assume in our approach. A model of time involves deciding on various issues:

- The *temporal primitive(s)*: instants vs. periods vs. both.
- The *structure* of time: discrete vs. dense, bounded vs. unbounded, ordering (partial, linear, circular,...).

- **Propositional references.** By propositional references we mean expressing relations on entities which are themselves propositions. We have examples in first and third sentences. Let us take the later: "the buyer *Eddi* has the obligation to pay the seller *Tami*...". we have a proposition "*Eddi*pay *Tami*" -which could be formalized as $pay(Eddi, Tami)$- and we have a reference to it since this an obligation that b has got $-obligation(Eddi, pay(b, a))$[2]. We see that if we do not want to jump out of first-order logics we must find some alternative way to representing it. This is an important issue in legal reasoning due to the numerous references of this sort that appear in legal knowledge [15].

3 Legal Temporal Knowledge Representation

In this section we present an approach to legal knowledge formalization with an explicit account of temporal information which we belief satisfactorily address the issues above.

3.1 Temporal Tokens

Our approach is based on the notion of *temporal token*. Intuitively a temporal token represents a particular instance of a domain *relationship* at a particular time[3]. For example, consider our first sentence: "an escrew contract between *Eddi* and *Tami* was concluded

---

[2] we are in fact considering only atomic propositions.

[3] Two remarks to be made on this statement. On the one hand, we are considering that our domain can be conceptualized as a set of relationships. A relationship can be formalized as a logical predicate or as something more complex, e.g. a set of predicates. some of them will look as tuples in a *relational database*, others probably will have the appearance of some sort of constraint. On the other hand, by "particular time" we do not commit ourselves to any specific temporal unit: it could be an instantaneous unit or a durative one.

April the first". we are not going to represent the abstract notion of "escrew contract between *Eddi* and *Tami*", which would correspond to the notion of *temporal type* (this is the approach followed by some relevant approaches to temporal reasoning in ai [5, 2, 10, 6]), but we shall represent a particular instance of it, i.e. the specific escrew contract between *Eddi* and *Tami* which was concluded April the first. The idea of "token" is proposed in some philosophical works on action formalization [3] and it has already been used as well within the context of temporal reasoning in ai [8, 4, 7, 11, 14].

Temporal tokens can be introduced by using different techniques. We propose to introduce tokens as additional arguments of the original relations in our logic since it turns out to meet our temporal and also non-temporal representational requirements, yet is the technically simplest way of doing that. We call this approach *temporal token arguments* [11]. In the example above, for instance, we can formalize the "contract" relation between *buyer* and *seller* as a 3-place predicate and the sentence is represented by $contract(escrew, Eddi, Tami)$. Now, to talk about the time related to the contract we introduce a temporal token, tt_1, as an additional argument of *contract* getting

$$contract(escrew, Eddi, Tami, tt_c)$$

The token constant tt_c "labels" the particular temporal instance of a contract of type *escrew* between the buyer *Eddi* and the seller *Tami* at a certain time. Intuitively we can think about it as an index for the various instances of the domain relations at different times. To better convey the intuition behind the idea of temporal tokens and to make the language more *readable* we shall note token symbols in a relation predicate as an index of the relation, so the above example will be written as

$$contract_{tt_c}(escrew, Eddi, Tami)$$

Now that we have temporal tokens as part of our ontology we can think about addressing the issues concerning temporal representation we mentioned in previous section: temporal occurrence and temporal relations. Regarding the first one, we can directly take the token arguments expression to mean its actual temporal occurrence (at the times associated to the token). Or else we can introduce any predicate we wish to express temporal occurrence by merely taking the token (at least) as one of its arguments. We call them *temporal occurrence predicates* (TOP). For instance, we can represent that a token actually occurred by using the OCCURS top to write $\text{OCCURS}(tt_1)$. We discuss the second issue in next subsection.

3.2 Temporal Constraints

Having temporal tokens in our ontology we can comfortably talk about the temporal qualification itself. The sort of knowledge we wish to express will have the form of a constraint between the different times of (possibly different) tokens, being particular cases constraints on one single time, i.e. *unary* or *absolute* constraints. In order to do that we introduce a number of additional elements:

1. *Temporal functions:* these are functions which will map tokens to their relevant times, the times which will be used to construct temporal constraints.
2. *Pure temporal forms:* these are the formulas we will use to represent constraints among the time elements coming from the temporal tokens.

The basic temporal primitives upon which these forms are based has been determined according to the needs identified in the domain. In the examples we have been analyzing we have found both relations between time points as well as relations between time intervals. Therefore, we take both instants and periods as the times of our approach. We will have temporal functions to get both, the extreme instants as well as the period related to a token, and we will incorporate both instant-to-intant constraints as well as period-to-period ones. For the purposes of this work we did not need to decide on any specific structure of time.

4 The Representation Language

In this section we describe the syntax and semantics of the representation language we propose and then we discuss its usage in modelling knowledge from a certain domain.

4.1 Syntax

We define our logical language starting from a given sorted first-order logic lacking of an explicit representation of time. We proceed as follows:

- **Sorts.** To those given sorts we add the following ones:
    - $\mathcal{TT}$, time tokens
    - $\mathcal{I}$, time instants
    - $\mathcal{P}$, time periods
    - $\mathcal{D}$, time distances

  These are a sort of tokens which are the key piece of our approach, two temporal sorts -one for instantaneous pieces of time and the other for durative ones- and finally the time distances needed to construct our temporal constraints.

- **Functions.** We introduce four one-place function symbols whose domain is of sort $\mathcal{TT}$ and whose range is of sort $\mathcal{T}$, namely:

$$\begin{aligned} \textsc{Instant}, \textsc{Begin}, \textsc{End} &: \mathcal{TT} \longrightarrow \mathcal{I} \\ \textsc{Period} &: \mathcal{TT} \longrightarrow \mathcal{P} \end{aligned}$$

  the function INSTANT is used to refer to the time of an instantaneous occurrences, whereas BEGIN, END and PERIOD denote the beginning time point, the end time point and the time interval of a durative occurrence.

- **Predicates.** Each n-ary predicate symbol of a certain signature from the given set of predicates is transformed into a n+1-ary predicate symbol where the added argument is of sort $\mathcal{TT}$. A particular case of it are the so-called *temporal occurrence predicates* (top). Once time is introduced explicitly, we can introduce some predicates to assert the temporal occurrence of a certain proposition. So far there is only one we need here: the 1-place predicate OCCURS which is used to express the occurrence of the event described by a certain token. In addition, now that we have introduced tokens in our language we have the possibility of talking about them regarding relations which we were not able to express before within a standard first-order logic.

- **Temporal constraints.** We introduce a number of predicates to represent temporal constraints over time points. To meet the requirements from the legal domain we introduce two different types of temporal constraints:

  - *Instant-to-instant metric constraints*: they express a constraint on the set of possible metric temporal distances between two time points. their form is:

    $$\mathrm{t}_j - \mathrm{t}_i \in [d_1^-, d_1^+], \ldots, [d_n^-, d_n^+]$$

    where t_i, t_j are instant terms and d_i's are time distance terms. Obviously the expressive power of temporal constraints lies on the capability of expressing uncertain or imprecise knowledge. However, in the case that the knoweldge is precise, although it can be expressed without problems, the resulting expression is not much natural. Therefore we propose some specific forms (which are syntactic sugar of the previous one) to express precise knowledge:

    $$\mathrm{t}_i = d$$
    $$\mathrm{t}_j - \mathrm{t}_i = d$$

  - *Period-to-period qualitative constraints*:

    $$p_i \ \mathrm{IR} \ p_j$$

    where ir is a subset of allen's thirteen interval relations [1] and p_i, p_j are period terms. A period term can be either a period constant symbol, a period variable symbol, or the function PERIOD applied to a either token term or to a pair of instant terms[4].

4.2 Semantics

The semantics is a slight variation from the standard many-sorted first-order logic one. Our interpretations will contain disjoint sets of objects according to the syntactic sorts defined,

---

[4] to be rigorous we should have two different symbols in place of our PERIOD to avoid having a single function with two different signatures. We do so for the sake of readability of our language. We could do it without having any significant difference from our development.

and functions and predicates will be interpreted in the appropriate sorts according to their signature. Logical connectives and quantifiers will be interpreted as usual. The only non-standard element comes on the temporal part. The interpretation has no set of times but only a set of time distance $\mathcal{D}_{\mathcal{D}}$ over which a linear order relation $\leq$ is defined as well as the addition $(+)$ operation with which forms a conmutative group structure. The intuition behind such a definition is that both times and time distances are semantically elements of the same set. Any instant can always be regarded as the distance between time 0 and its time, and any period can always be regarded also as an ordered pair of instants, i.e. of time distances. So, an interpretation will be composed of a meaning function $\mathcal{M}$ such that:

$$\begin{aligned} \mathcal{M} &: \mathcal{I} \longrightarrow \mathcal{D}_{\mathcal{D}} \\ \mathcal{M} &: \mathcal{P} \longrightarrow \{\langle d, d' \rangle : d, d' \in \mathcal{D}_{\mathcal{D}}, d \leq d'\} \\ \mathcal{M} &: \mathcal{D} \longrightarrow \mathcal{D}_{\mathcal{D}} \end{aligned}$$

and the temporal constraints will be interpreted, according to their definition, as follows: given an interpretation $\mathcal{I}$:

$$\begin{aligned} \mathcal{I} \models t_j - t_i \in [d_1^-, d_1^+], \ldots, [d_n^-, d_n^+] \text{ iff } & \ d_1^- \leq \mathcal{M}(t_j) - \mathcal{M}(t_i) \leq d_1^+ \text{ or} \\ & \ \vdots \\ & \text{ or } d_n^- \leq \mathcal{M}(t_j) - \mathcal{M}(t_i) \leq d_n^+ \end{aligned}$$

the interpretation for the qualitative constraints will be similar, taking for each *interval relation* its interpretation as a *cnf* of point-to-point qualitative relations which in turn can be viewed as particular cases of a metric one. For example, the MEETS relation between two periods, $p_i \text{MEETS} p_j$, would be satisfied by an interpretation iff

$$\text{second}(\mathcal{M}(p_i)) = \text{first}(\mathcal{M}(p_j))$$

4.3 Conceptualizing and Representing Knowledge

Let us take a particular example to illustrate how our language can be used for knowledge representation and to demonstrate its adequacy. Consider the following narrative describing an specific story about contracts as well as some general rules that are applicable to them.

Example 1 (simplified contract scenario)

1. *"Eddi got the idea of a* escrew *contract with Tami".*
2. *"Eddi offered that contract to Tami at time t_1".*
3. *"Tami accepted the contract from Eddi at time t_2".*
4. *"A contract is* concluded *when the* offeree *accepts it".*

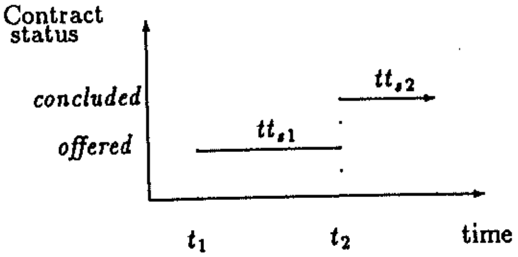

Figure 1: *History* of the *contract status* fluent on contract tt_c.

The formalization of the first three statements would be as follows:

$$\begin{array}{lll}
1. & contract_{tt_c}(escrew, Eddi, Tami) & \\
2. & offer_{tt_o}(Eddi, tt_c, Tami) & \textsc{Instant}(tt_o) = t_1 \quad \textsc{Occurs}(tt_o) \\
3. & accept_{tt_a}(Tami, tt_c, Eddi) & \textsc{Instant}(tt_a) = t_2 \quad \textsc{Occurs}(tt_a)
\end{array}$$

From the occurrence of the offering and acceptance events we would like to somehow complete the information about the status of the contract. Specifically we would like to derive facts such as (see figure 1):

$$\begin{array}{l}
contract\_status_{tt_{s1}}(tt_c, offered) \\
contract\_status_{tt_{s2}}(tt_c, concluded) \\
tt_{s1}\textsc{Meets}\ tt_{s2}
\end{array}$$

The questions of how can we obtain them and what is the knowledge needed to derive them can be answered by following the same intuition than in the original *Event Calculus* approach, but taking technical advantage of having temporal tokens and temporal constraints in the language. Briefly, to derive the facts on the left hand side we would need some expression of the *causal relationship* between event occurrence and proposition holding[5]. In our case such a relationship will be between tokens. For instance,

$$\begin{array}{lll}
\forall TT_a, TT_c, X, Y & accept_{TT_a}(X, TT_c, Y) \wedge \textsc{Occurs}(TT_a) & \\
& \longrightarrow & \\
& \exists TT_{co} & contract\_status_{TT_s}(TT_c, concluded) \wedge \\
& & \textsc{Instant}(TT_a) = \textsc{Begin}(TT_s) \wedge \\
& & \textsc{Causes}(TT_a, TT_s)
\end{array}$$

To derive additional information that "completes" the intended picture of the temporal description of the example, i.e. the picture that one can expect from common-sense , we should need to apply techniques related to *temporal persistence* and *temporal clipping*. These functionalities can be adequately reformulated by using the the metric temporal constraints of our language as shown in [13][6].

---

[5] The *Event Calculus* introduces the predicates INITIATES and TERMINATES to do so.

[6] Although the language defined in [13] is not the same than here, they share the main representational features, namely temporal tokens and temporal constraints.

Let us now have a look again on the way we *conceptualized* the example, i.e. decided on the predicates and tokens coming out from the given narrative statements. One may observe that the notion of *contract status* could have been formalized merely as another argument of the predicate *contract* as the TYPE of the contract is. What is the criterium to follow in order to determine when a predicate argument must be considered a different predicate by its own ? The answer is based on the pattern of change through time: an attribute may deserve being conceptualized as a different fluent and represented as a different predicate if its pattern of change is significantly different from the one of the predicate its is argument of. In the example, contract status has been taken out of the concept of contract since changes with higher frequency than the other attributes of contract. Since the status attribute is taken as a different fluent we get tokens for it describing its change through time (tokens tt_{s1} and tt_{s2}), plus a token for contract (tt_c) representing its life time. Temporal tokens turn out to be very well-suited to support attribute splitting since it is just another instance of *propositional reference.*

5 Future Work

Upon the basis of the legal knowledge representation formalism presented in this paper we are working on several directions to address problems clasical in knowledge representation.

One of them is *information incompleteness.* The information relevant to a problem case is often not complete in the legal domain. In particular, descriptions of temporal events do not always include information about each of its features. Our proposal can be easily extended to account for such a sort of incompleteness by following *Semantic Case Decomposition*, an idea that we borrow from Kowalski & Sergot who used it in the *Event Calculus* [8] for a similar purpose. It basically consist of deciding on a set of descriptor or attributes which will be used to refer to each paramenter for each argument of our initial predicates. Then, every attribute is taken as a predicates whose attributes are a token and an attribute value. The advantage here is that the language user has the flexibility of representing just the pieces of information that are known ignoring the rest.

Example 2 *In the introductory example, the piece of knowledge "A* escrew *contract between* Eddi *and* Tami *has been concluded at time* t_1*" is formalized as*

$$\begin{array}{ll}
\mathrm{relation}(tt_1, \mathrm{contract}) & \mathrm{relation}(tt_{11}, \mathrm{contract\_status}) \\
\mathrm{contract\_type}(tt_1, \mathrm{escrew}) & \mathrm{contract}(tt_{11}, tt_1) \\
\mathrm{buyer}(tt_1, \mathrm{Eddi}) & \mathrm{status}(tt_{11}, \mathrm{concluded}) \\
\mathrm{seller}(tt_1, \mathrm{Tami}) & \mathrm{BEGIN}(tt_{11}) = t_1
\end{array}$$

An additional expressiveness advantage is that now we can quantify over relations such as *contract* to express, for instance, general properties about any relation on a certain individual or object.

Another matter of future work is default reasoning not only in the classical sense but in the sense of making and monitoring assumptions about the temporal distances between relevant times, specially on the duration (i.e. distance from BEGIN to END) of relation occurrences.

Finally we are working on the identification of particular subdomains where the sort of temporal algebra required can be restricted in order to obtain computational complexity advantages.

6 Conclusions

We proposed an approach to temporal representation for legal reasoning systems based on the method of *temporal token arguments* and the embedding of temporal constraints.

Moreover, the notion of token introduced as an additional argument to domain predicates turns out to be very adequate to express *propositional references* which is an important representational issue in legal knowledge. In conclusion, we can satisfactorily address two different requirements by applying the clear and technically simple idea of having (temporal) tokens as arguments.

References

[1] J. F. ALLEN. Maintaining knowledge about temporal intervals. Technical Report TR86, Department of Computer Science University of Rochester, 1983.

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

[3] D. DAVIDSON. The logical form of action sentences. In N. Rescher, editor, *The Logic of Decission and Action*. University of Pittsburgh Press, 1967.

[4] T. L. DEAN and D. V. M. DERMOTT. Temporal data base management. *Artificial Intelligence*, 32:1987, 1987.

[5] D. M. DERMOTT. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155, 1982.

[6] A. GALTON. A critical examination of Allen's theory of action and time. *Artificial Intelligence*, 42:159–188, 1990.

[7] A. GALTON. Reified temporal theories and how to unreify them. In *Proc. IJCAI'91*, pages 1177–1182, 1991.

[8] R. KOWALSKI and M. SERGOT. A logic-based calculus of events. *New Generation Computing*, 3, 1986.

[9] L. T. McCARTY. Some requirements on an action language for legal discourse (position paper). In *Spring Symposium Series'95: Extending Theories of Action*, pages 136–138. AAAI, 1995.

[10] Y. SHOHAM. Temporal logics in AI: Semantical and ontological considerations. *Artificial Intelligence*, 33:89–104, 1987.

[11] L. VILA. *On Temporal Representation and Reasoning in Knolwedge-based Systems.* Phd thesis on computer science, Dept. of Computer Languages and Systems, Technical University of Catalonia, C/ Pau Gargallo 5, 08028 BARCELONA, 1994.

[12] L. VILA. A survey on temporal reasoning in artificial intelligence. *AI Communications*, 7(1):4–28, March 1994.

[13] L. VILA. Reformulating event calculus in terms of metric temporal constraints. In H. Yoshino, editor, *Symposium on Research for the Development of Legal Expert Systems*, pages 254–271. Meiji Gakuin University, 1995. Invited talk.

[14] L. VILA and H. REICHGELT. The token reification approach to temporal reasoning. *Artificial Intelligence*, (in press) 1995.

[15] H. YOSHINO. Representation of legal knowledge by compound predicate formula. In D. T. C. Biagioli, G. Sartor, editor, *Proc. of the Workshop on Legal Application of Logic Programming*, pages 128–137. ICPL'94, 1994.

松尾資料（2）

Legal Expert System Association(LESA)

The 3rd International Workshop on a Legal Expert System for the CISG

Proceedings of the Workshop

May 20, 1995

Edited by
Kevin D. Ashley and Hajime Yohsino