Temporal Qualification in Artificial Intelligence

Han Reichgelt
Department of Information Technology
Georgia Southern University
Statesboro, GA 30460-8150, U.S.A.

Lluís Vila*
Department of Software
Technical University of Catalonia
Barcelona, Catalonia, Spain

Abstract

We use the term *temporal qualification* to refer to the way a logic is used to express that temporal propositions are true or false at different times. Various methods of temporal qualification have been proposed in the AI community. Beginning with the simplest approach of adding time as an extra argument to all temporal predicates, these methods move to different levels of representational sophistication. In this paper we describe and analyze a number of approaches by looking at the syntactical, semantical and ontological decisions they make. From the ontological point of view, there are two issues: (i) whether time gets full ontological status or not and (ii) what do the temporally qualified expressions represent: *temporal types* or *temporal tokens*. Syntactically time can be explicit or implicit in the language. Semantically a line is drawn between methods whose semantics is based on standard first-order logic and those that move beyond it to either higher-order semantics, possible-world semantics or an *ad hoc* non-standard temporal semantics.

1 Introduction

Temporal reasoning in artificial intelligence deals with *relationships* that hold at some times and do not hold at other times (called *fluents*), *events* that occur at certain times, *actions* undertaken by an actor at the right time to achieve a goal and *states* of the world that are true or hold for a while and then *change* into a new state that is true at the following time. Consider the following illustrative example that will be used throughout the paper:

> "On 1/4/04, SmallCo sent an offer for selling goods g to BigCo for price p with a 2 weeks expiration interval. BigCo received the offer three days later[1] and it has been effective since then. A properly formalized offer becomes effective as of it is received by the offered and continues to be so until it is accepted by the offered or the offer expires (as indicated by its expiration interval). Anybody who makes an offer is committed to the offer as long as the offer is effective. Anybody who receives an offer is obliged to send a confirmation to the offerer within two days."

This narrative contains instances of the temporal phenomena mentioned above:

*Research funded by the Spanish CICYT under grant TIC2002-04470-C03-01 and under "Web-I(2)" grant TIC2003-08763-C02-01.

[1] This trading example probably ought to be updated to an e-treading scenario where the messages are received 2 or 3 seconds after being sent, however we believe that the essential representation issues and results still would be the same.

- *fluents* such as "x being effective from time t to time t'". In this case, the beginning and the end and the duration are not fully determined but the beginning is. Naturally, this fluent may also hold on a set of non-overlapping intervals of time.

- *Actions* such as "an agent x to send an object or message y to agent z at time t". Observe that this also may happen more than once for the same x, y and z, with t being the only distinctive feature.

- *Events* such as "x receives y on time t". Both executed actions and events potentially are causes of some change in the domain. In this case, the event causes the offer to be effective as of the reception time.

- *States* such as the state before "1/Apr/04" and the state right after receiving the offer where the offer is effective and various obligations hold.

Additionally, we observe other kinds of temporal aspects such as:

- Temporal features of an object or the object itself. For instance "the manager of SmallCo" can be a different person at different times or even "SmallCo" could denote different companies at different times depending on our timeframe.

- Temporal relations between events and fluents such as "The offer is effective as of it is received by the offered and will be so until it is accepted by the offered or it expires" or "sending an object <u>causes</u> the destiny to receive it between 1 and 4 days later."

- Temporal relations between fluents such as "the offered is committed to the offer as long as the offer is effective" or "an offer cannot be effective and expired at the same time".

Notice that references to time objects may appear in a variety of styles: absolute ("1/Apr/04"), relative ("two days later"), instantaneous (`now`), durative ("the month of march"), precise ("exactly 2 days"), vague ("around 2 days"), etc.

This example illustrates the issues that must be addressed to design a formal language for temporal reasoning[2], namely the *model of time* i.e. the set or sets of time objects (points, intervals, etc.) that time is made of with their structure, the *temporal ontology* i.e. the classification of different temporal phenomena (fluents, events, actions, etc.), the *temporal constraints language*, i.e. the language for expressing constraints between time objects, the *temporal qualification* method and the *reasoning system*. Research done on these issues is reviewed in the various chapters of this volume. Here we shall focus on **Temporal Qualification**:

> By a temporal qualification method we mean the way a logic (that we shall call the *underlying logic* of our temporal framework) is used to express the above temporal phenomena at specific times.

One may either adopt a well-known logic equiped with a well-defined model and proof theory or define a language with a non-standard model theory and develop a proof theory for it.

Indeed, the temporal qualification method is a central issue in defining a temporal reasoning framework and it is closely related to the other issues mentioned above. The reader must

[2]The presentation is biased towards the standard definition of first-order logic (FOL), although nothing prevents the situation of the elements described here in the context of a different logic, including non-standard semantics for FOL, modal logics and higher-order logics.

be aware that most of these issues are discussed in other parts of this volume. We discuss them here up to level needed to make to make our presentation self-contained and discuss the advantages and shortcomings of each temporal qualification approach.

1.1 Temporal Reasoning Issues

1.1.1 The Model of Time

Modeling time as a mathematical structure requires deciding (i) the class or classes of basic objects time is made of such as instants, intervals, etc. (i.e. the *time ontology*) and (ii) deciding what are the properties these time sets: dense vs. discrete, bounded vs. unbounded, partial/total/...order, etc. (i.e. the *time topology*).

This issue is discussed in chapter *Theories of Time and Temporal Incidence* in this handbook and we shall remain silent on what the best model of time is. When introducing a temporal qualification method we shall merely assume we are given a *time structure*

$$\langle \mathcal{T}_1, \ldots, \mathcal{T}_{n_t}, \mathcal{F}_{time}, \mathcal{R}_{time} \rangle$$

where each $\mathcal{T}_i$ is a non-empty set of time objects, $\mathcal{F}_{time}$ is a set of functions defined over them, and $\mathbf{P}_{time}$ is a set of relations over them. For instance, when formalizing our example we shall take a time structure with three sets: a set of time points $\mathcal{T}_1$ that is isomorphic to the natural numbers (where the grain size is one day), the set of ordered pairs of natural numbers $\mathcal{T}_2$ and $\mathcal{T}_3$ a set of durations or temporal spans that is isomorphic to the integers. $\mathcal{F}_{time}$ contains the functions $\texttt{begin}, \texttt{end} : \mathcal{T}_2 \mapsto \mathcal{T}_1, - : \mathcal{T}_1 \times \mathcal{T}_1 \mapsto \mathcal{T}_3$ and $\mathcal{R}_{time}$ contains relations like $<_1$ and $=_1$ for $\mathcal{T}_1$ $<_3$ and $=_3$ for $\mathcal{T}_3$ and the known qualitative interval relations like $\texttt{Meets}, \texttt{Overlaps}, \ldots \subseteq \mathcal{T}_2 \times \mathcal{T}_2$.

The decision made on the model of time is "fairly" independent of the temporal qualification method but it has an impact on the formula we one can write and the formula we can proof. On the one hand,

The reader must bear in mind that the temporal qualification method one selects will determine how the model of time adopted will be embedded in the temporal reasoning system. The completeness of a proof theory depends on the availability of a theory that captures the properties of the model of time and allows the proof system to infer all statements valid in the time structure. Such a theory, the *theory of time*, may have the form of a set of axioms written in the temporal language that will include symbols denoting functions in $\mathcal{F}_{time}$ and relations in $\mathcal{R}_{time}$. For example, the transitivity of ordering relationship (denoted by $<_1$) over $\mathcal{T}_1$ can be captured by the axiom

$$\forall\, t_1, t_2, t_3\ [t_1 <_1 t_2 \wedge t_2 <_1 t_3) \rightarrow t_1 <_1 t_3]$$

Depending on the time structure and the expressive power of *underlying logic* it can be the case that such complete set of axioms cannot be written in our language.

1.1.2 Temporal Constraints Language

The language of expressions denoting constraints between temporal objects. Formulas are logical combinations of atoms built on *time constants* (possibly of different nature) e.g. "1/Apr/04" or "2 days", that denote time objects in $\mathcal{T}_1, \ldots, \mathcal{T}_{n_t}$, function symbols that denote functions in $\mathcal{F}_{time}$ (we call them *time functions*), and predicates, we call them *time predicates*, that denote relations in $\mathcal{R}_{time}$ whose arguments are all *time terms*.

1.1.3 Temporal Ontology and the Theory of Temporal Incidence

As discussed in two previous chapters in this book ("Eventualities" by A. Galton and also partly in Vila's "Theories of Time and Temporal Incidence"), temporal statements can be classified in various classes (as illustrated by the different temporal phenomena in our example) each associated with a pattern of temporal incidence. Different temporal ontologies have been proposed in different contexts such as natural language understanding and common-sense reasoning. In most cases, the result of such ontological studies is a classification of temporal relations into a number of classes (e.g. fluents, events, etc.). For our purpose here we shall refer to these classes as

$$\textit{temporal entities} = \{\mathcal{E}_1, \ldots, \mathcal{E}_{e_n}\}$$

and assume that each class is accompanied by a pattern of temporal incidence that sometimes is characterized by one or more axioms written in our logical language through some sort of *temporal incidence meta-predicates*. We call the set of these axioms the *theory of temporal incidence* $\mathcal{T}_i$.

For instance, when formalizing our example we shall have the following temporal entities: (i) *events* or *accomplishments* such as "send a legal object on time t", that occur either at a time point i.e. one day, or during a time span (several days) and (ii) *fluents* such as "the offer is effective as of t" that hold homogeneously throughout a number of days. Whereas the occurrence of an event over an interval is *solid* (if it holds on a interval it does not hold on any interval that overlaps with it), the holding of a fluent over an interval is *homogeneous* (if it holds during an interval it also holds over any subinterval). For example, if we would have the meta-predicate HOLDS[3], then for each fluent $R^k \in \mathcal{E}_{fluents}$

$$\begin{array}{l} \forall\, t_1, t_2, x_1, \ldots, x_k\ [\ \textsc{Holds}(t1, t2, R^k(x_1, \ldots, x_k)) \rightarrow \\ \quad \forall\, t_3, t_4\ [(t_3, t_4) \subseteq (t_3, t_4) \rightarrow \textsc{Holds}(t3, t4, R^k(x_1, \ldots, x_k))]\ ] \end{array}$$

Although these issues are out of the scope of this study, we must bear in mind that the temporal qualification method determines how are the temporal incidence axioms written and the formula derived from them.

1.2 Temporal Qualification Issues

Having the previous issues out of the way, let's point out the issues determined by a temporal qualification method[3]:

- *The distinction between temporal and atemporal individuals.* As illustrated by the example, a distinction ought to be made between atemporal individuals (i.e. individuals that are independent of time such as color green, number 3, ...) and individuals whose existence depends on time such as "contract c1-280-440" or "the SmallCo company".

- *The distinction is made between temporal and atemporal functions.* The introduction of time also leads to the need of making a semantic distinction between temporal functions and classical functions possibly co-existing in the same logic. We define a *temporal function* as a function whose value can be different at different times, for example "the manager of"(SmallCo). We shall call $\mathcal{F}_t$ the set temporal function symbols and $\mathcal{F}_\infty$ the set of atemporal predicate symbols.

[3] As a matter of fact, it can be argued that any method of temporal qualification method can be regarded as the set of decisions made with respect to these issues.

- *The distinction between temporal and atemporal relations.* Similarly, a temporal logic ought to have a semantic distinction between relations whose truth-value can be different at different times, such as "agent a_1 sends an offer to a_2 of selling g" and those whose truth-value is independent of time as in classical logics such as "a contract is a legal document" (from the domain ontology) and "an offer is properly formalized". Notice that the *time relations* mentioned above are in fact atemporal relations.

- *The distinction between temporal occurrences and temporal types of occurrences.* By a temporal occurrence (namely **temporal token**) we mean a particular temporal relation when is true at a specific time (e.g. "on time t agent a_1 sends an offer to a_2 of selling g") as opposed to the term *temporal type* that means the set of all occurrences of a temporal relation (e.g. the set of all specific sending events of type "agent a_1 sends an offer to a_2 of selling g").

- *The specification of time and temporal incidence theories.* As we explained, the time and temporal incidence theories are do not condition the temporal qualification an representational issue for our temporal qualification method is the flexibility and expressiveness provided to specify the axioms of these theories.

- *The specification of "nested" temporal relations.* A temporal relation that relates objects or other relations that in turn are temporal. For example, "an agent is committed for a period to send a confirmation of a certain offer". The commitment, the send action and the offer are all temporal relations.

- *The specification of relations between temporal relations or their occurrences.* The paradigmatic example of this is the causal relation between a temporal relation whose occurrence causes another relation to hold. Other examples are incompatibility between temporal relations, correlation of temporal relations, or class-subclass links bounds between temporal relations.

As a matter of fact, temporal qualification is an issue in any formal temporal representation, no matter its context. In this section we quickly overview temporal qualification in different areas, ending with temporal qualification in AI where we briefly introduce the approaches that will be discussed in detail in the following sections.

1.3 Temporal Qualification in Logic

Classical Logics. Classical logics have proven useful for reasoning about domains that are atemporal (such as mathematics) or in domains where time is not a relevant feature and can be abstracted away (e.g. a diagnostic system in a domain where the times of the relevant symptoms do not affect the result of the diagnosis). However, in many domains time cannot be disregarded if we want our logical system to be correct and complete. Logicians have studied different theories to model time and designed various temporal logics. In such logics, statements are no longer timelessly true or false but are true or false at a certain time. Temporality may be inherent in any component of the formula, functions, predicates or logical connectives, in particular, as soon as we have a time domain it is natural to *quantify* over time individuals.

A simple approach to formulating a temporal logic is as a particular FOL with a time theory. Temporal functions and predicates are supplemented with an additional argument representing the time at which they are evaluated and time is characterized by a set of first-order axioms. Standard FOL syntax and semantics are preserved and, therefore, standard FOL proof theory is also valid however we must be careful with the time axioms. On the one hand, as discussed above, the completeness of the theorem prover depends on the existence of a complete first-order axiomatization for the intended time structures. On the other hand, the time axioms may easily lead to an explosion of the search space to be explored by the theorem prover.

It is convenient to move to a **many-sorted logics** [6, 24, 12, 5] since it naturally allows to distinguish between time and non-time individuals. Many-sorted logics do not extend FOL's expressive power (it is well-known that a many-sorted first-order logic can be translated to standard FOL) but it provides several advantages. Notation is more efficient as formulas are more readable, more "elegant" and some $\rightarrow$ can be dropped yielding more compact formulas. Semantics also can be regarded as a simple FOL extension, therefore many-sorted logic preserves the most interesting logical properties of FOL while it provides some potential for making reasoning more efficient. A formula parser can perform "sort checking" and some of the reasoning involving the sortal axioms can be moved into the unification algorithm. Although this leads to a more expensive unification step, this is typically more than off set by the reduction in the search space that can be achieve through the elimination of the sortal axioms from the theory.

Modal Logics. An alternative way to incorporate time is by complicating the model theory, along the lines of modal logic. Using the common Kripke-style possible world semantics for **modal logics**, each *possible world* represents a different time while the *accessibility relationship* becomes a temporal ordering relationship between possible worlds. Different modal temporal logics are obtained by (i) imposing different properties on the accessibility relationship, and (ii) choosing different domain languages (e.g. propositional, first-order, ...). In order to provide an efficient notation, modal varieties of temporal logic use a number of temporal modal operators, operators that are applied to propositions in the domain logic and change the time with respect to which the proposition is to be interpreted. Traditionally, four primitive *modal temporal operators* are defined: F (at some future time), P (at some past time), G (at any future time) and H (at any past time). Hence $F\varphi$ denote that the formula φ is true at some future time. Other common temporal modalities are p UNTIL q (p is true until q is true), p SINCE q (p has been true since q has been true) or AT(t) p (p is true at time t).

1.4 Temporal Qualification in Databases

From a purely logical point of view, classical database applications [1, 22, 4] have followed the first approach outlined in the previous section. In addition to the original relations and a data domain for the values of the attributes, the temporal database includes a temporal domain. Typically, temporal databases use an instant-based approach to time. Some kind of mathematical structure is imposed on instants: usually one that is isomorphic to the natural numbers. A temporal database can be abstractly defined in a number of different ways[4].

The Model-theoretic View. A database is abstractly viewed as a two-sorted first-order language. Each relation P of arity n gives rise to a predicate R with arity $n + 1$, where the additional argument is a time argument. Its intended meaning is as follows:

$$(a_1, \ldots, a_n, t) \in R \text{ if and only if } P(a_1, \ldots, a_n) \text{ holds at time } t$$

All a_i are constant symbols denoting elements in a data domain. The set of constant symbols is possibly extended with some symbols denoting elements in the temporal domain. The theory may also add some time function and relation symbols, such as a function symbol $t+1$ to denote the time immediately following t or the relation $<$ to denote temporal ordering.

Some databases require multiple temporal dimensions. The usual case is that a single temporal domain is assumed. The relational predicates are then given two temporal arguments to indicate that the relation holds between two points in time (interval timestamps), or a number of time arguments used to model multiple kinds of time. For example, in the so-called *bi-temporal databases*, one set of temporal arguments refers to the *valid time* (the time when the relation is true in the real world) and another to the *transaction time* (the time when the relation was recorded in the database) [20]. The different interpretations of multiple temporal attributes databases are captured by *integrity constraints*. For example, a constraint may state that the beginning of an interval always precedes its end or that transaction time is not before valid time.

The Timestamp View. Moving to concrete databases (database that are to be implemented and therefore must allow for a finite representation), the most useful view is the timestamp view. In this view, each tuple is supplemented with a timestamp formula possibly representing an infinite set of times. A timestamp formula is a first-order formula with one free variable in the language of the temporal domain, e.g. $0 < t < 3 \vee 10 < t$. Different temporal databases result from different decisions about what subsets can be defined by timestamp formulas. An interesting temporal domain is the Presburger arithmetic as it allows one to describe periodic sets and therefore has obvious application in calendars and repeating events.

It is not clear whether timestamps could be defined in a language richer than the first-order theory of the time domain [4]. However, there are some approaches that extend the timestamp view by associating timestamps not to tuples but to attribute values [21]. Such approaches increase data expressiveness and temporal flexibility but pay for this through increased query complexity, and hence decreased efficiency.

Temporal Query Languages. While the temporal arguments approach has been predominant in temporal databases a wide variety of languages have been explored for querying them. These range from logic programs with a single instantaneous temporal argument to temporal logics with modal operators such as SINCE , UNTIL , etc. Readers interested in temporal query languages are referred to the the chapter on this subject in this volume.

1.4.1 Temporal Qualification in Computer Systems

Computer systems can be regarded as a sequence of states. Each state is characterized by a set of propositions stating what is true at that time. Interesting reasoning tasks such as system specification, verification and synthesis can be states in terms of logical properties that must hold at some times/states in the future when the system starts at a certain initial state.

In this context, it is appropriate to model time as an ordered, discrete sequence of time points and the dominant temporal qualification approach is modal logics. The reasons are that temporal modal operators allow one to easily express relative temporal references (e.g.,"the value of variable a is x until this assignment statement is executed"). Modal operators also provide a very efficient notation for various levels of nested temporal references (e.g. "p will have been true until then"). Also the semantics fits the discrete time model very well. Since modal temporal logic is discussed at length in other chapters in this volume we will not expand on this discussion here and merely refer the reader to these other chapters.

1.5 Temporal Qualification in AI

It has been recognized that AI problems such as natural language understanding, common-sense reasoning, planning, autonomous agents, etc. make greater demands on the expressive power of temporal logics than many other areas in computer science. For example, the temporal reasoning that autonomous agents are required to undertake typically require both relative and absolute temporal references. Autonomous agents also often require reasoning about different possible futures and, if they are to engage in abductive reasoning, they may have to consider different possible pasts in order to determine which past is the best explanation for the current state of affairs.

All techniques that have been employed in temporal databases and/or computer science have also been applied in AI:

- The method of *temporal arguments* has been an appealing method to many AI researchers because of its simplicity, the availability of interesting results on FOL theorem proving, and the fact that its expressiveness is not as limited as has commonly been claimed[3] if we allow temporal arguments in functions as well as in predicates.
- *Temporal Modal logics* have been appealing to those interested in formalizing natural language (the so-called *tense logics*) and formal knowledge representation.

However, it is a third family of techniques that attracted much of the attention from AI researchers, specially during the 80s and 90s, namely the *reified approach.* In the reified approach, one "reifies" temporal propositions and introduces names for them. One then uses predicates such HOLDS or OCCURS to express that the named proposition is true at a certain time, or over a certain interval. Classical examples of this approach are the *situation calculus* [13, 17, 18], McDermott's logic for plans[14], Allen's interval logic [2], *event calculus* [11, 18], the *time map manager* [8], Shoham's logic for time and change [19] Reichgelt's temporal reified logic [16], Token reified logics [23], action languages [?], etc.

The attraction of the reified approach is to a large extent due to the fact that the inclusion of names for such entities as actions, events, properties and states in the formalism allows one to predicate and quantify over such entities, something that is not allowed in either the method of temporal arguments or in temporal modal logic. This expressive power is important in many AI applications. Even our seemingly simple example includes examples of propositions that require quantification. The proposition "An offer remains valid until it either expires or is withdraw" is most naturally regarded as involving a quantification over expiration and withdrawal events. Other examples of propositions that are best regarded as involving quantification over events and/or states include propositions such as "whenever company X is in need of cleaning services, it issues a tender document", or "State-funded agencies can only issue contracts after an open and transparent tendering process".

Although reified logics have proven very popular, they have come under attack from different angles. First, temporal reified systems have been presented without a *precise formal* *semantics.* While temporal reified logics in general remain first-order, the introduction of names for events and states, and some meta-predicates tot assert their temporal occurrence, means that one cannot simplistically rely on the standard semantics for first-order logic to provide a rich enough semantics for a temporal reified logic. In some cases, like Shoham's reified logic, the apparent increased expressive power is not superior to that of the standard, easy-to-define method of temporal arguments [3] Second, in the cases in which the expressiveness advantage is clear, the price to pay is a logic that may end being far too complex. Third, reified temporal logics also received criticisms from the ontological point view, Galton [9], for example, considers them *"philosophically suspect and technically unnecessary"*, as they seem to advocate the introduction of *temporal types* in the ontology. One way to escape from this criticism is to move to an ontology of temporal propositions based on *temporal tokens*. A temporal token is not to be interpreted as a type of temporal propositions but as a particular temporal instance of a temporal proposition. Such ontology has been used as the basis for some alternative temporal qualification methods such as *temporal token arguments* or *temporal token reification.*

1.6 Paper Outline

In the following sections we describe in detail the most relevant methods of temporal qualification in AI that we briefly introduced in the previous subsection. We look at the syntactical, semantical and ontological decisions they make. As we have seen, syntactically we distinguish between those that represent times as additional arguments and those that introduce specific temporal operators. Semantically, the main distinction is between those methods that stay within standard first-order logics and those that move to some sort of non-standard semantics, either defined from scratch or by adapting some known non-classical semantics such as modal logics. Finally, from the ontological point of view, we distinguish between the methods that only give full ontological status to time from the ones that, in addition, include in the ontology denotations for temporal propositions: these can be either *temporal types* or *temporal tokens* if they represent particular temporal occurrences of a temporal expression.

Each method is illustrated by formalizing our trading example. The reader should recall that we assume we are given the following:

- **The model of time.** The *time structure* composed of the three time subdomains: a set of time points $\mathcal{T}_1$ that is isomorphic to the natural numbers (where the grain size is one day), the set of ordered pairs of natural numbers $\mathcal{T}_2$ and $\mathcal{T}_3$ a set of durations or temporal spans that is isomorphic to the integers. $\mathcal{F}_{time}$ contains the functions $\texttt{begin}, \texttt{end} : \mathcal{T}_2 \mapsto \mathcal{T}_1, - : \mathcal{T}_1 \times \mathcal{T}_1 \mapsto \mathcal{T}_3$ and $\mathcal{R}_{time}$ contains relations like $<_1$ and $=_1$ for $\mathcal{T}_1$ $<_3$ and $=_3$ for $\mathcal{T}_3$ and the known qualitative interval relations like $\texttt{Meets}, \texttt{Overlaps}, \ldots \subseteq \mathcal{T}_2 \times \mathcal{T}_2$.

- **Temporal Entities and Temporal Incidence Theory.** We have two temporal entities:

  1. *events* or *accomplishments* that occur either at a time point i.e. one day, or during a time span (several days) and the occurrence is *solid*: if it holds on a interval it does not hold on any interval that overlaps with it.

  2. *fluents* that hold homogeneously throughout a number of days: if it holds during an interval it also holds over any subinterval.

We analyze the advantages and shortcomings of each method according to a set of representational, computational and engineering criteria Among the representation criteria, we shall first look at the **expressiveness** of the language. In particular, it is important for our temporal qualification method to be able to represent the various types of propositions and axioms indicated in previous sections. The comparison will be informal and illustrated by our example. Second, we shall look at the **notational efficiency**. For a host of reasons, it is important that one is able to formalize knowledge into formulas that are compact, readable and elegant. Third, it is desirable to have an **ontology** that is clean and not unnecessarily complex. One wants to make sure that one avoids undesirable entities in one's ontology. For example, an ontology that requires one to postulate the existence of both types and tokens is suspect. On the other hand, one also wants to make sure that the entities that one postulates in one's ontology are rich enough to enable one to express whatever temporal knowledge one wants to express. A second type of criteria are **theorem proving** criteria such as soundness and completeness of the proof theory, efficiency of any theorem provers, as well as the possibility of using implementation technique to improve the efficiency of the theorem prover. Finally, we also bear in mind what one might call "engineering criteria", such as **modularity** of the method. Often temporal reasoning is but one aspect of the reasoning that the system is expected to undertake. For example, an autonomous agent needs to be able to reason not only about time but also about the intentions of other agents that it is likely to have to deal with. It would therefore be advantageous if the method of temporal qualification allows one to extend the reasoning system to include reasoning about other modalities as well.

2 Temporal Modal Logic

One possible approach to temporal qualification in AI is the adoption of modal temporal logic (*MTL*). We already briefly discussed modal temporal logic in section 1.3. Moreover, there is a chapter in this handbook devoted to modal varieties of temporal logic by H. Barringer and D. Gabbay, and our discussion of this approach is therefore extremely condensed.

2.1 Definition

Temporal modal logics are a special case of modal logic. Starting with a normal first order logic, one adds a number of modal operators, sentential operators which, in the case of temporal modal logic, change the time at which the proposition in its scope is claimed to be true. In other words, the problem of temporal qualification is dealt with by putting a modal operator in front of a non-modal proposition. For example, one may introduce a modal operator P ("was true in the Past"). When applied to a formula ϕ, the modal operator would change the claim that ϕ is true at this moment in time to one which states that ϕ was true some time in the past. Thus, the statement "SmallCo sent offer o1 to BigCo some time in the past" would be represented as $P\ Send(sco, o1, bco)$.

Modal temporal logic, as traditionally defined by philosophical logicians, is not particularly expressive. In its simplest form, modal temporal logic only allows existential and universal quantification over the past and the future. In other words, in its simplest form, modal temporal logic contains only four modal operators, namely P ("was true in the past"), H ("has always been true"), F ("will be true sometimes in the future") and G ("is always going to be true"). Clearly, this is insufficient for Artificial Intelligence, or indeed Computer Science. For example, none of the propositions in our example could be expressed in such an expressive poor formalism. It is for this reason that a number of authors (e.g., Fischer, 1991; Reichgelt, 1989) have introduced a number of additional modal operators, such as UNTIL , SINCE and a model operator scheme AT, which takes a name for a temporal unit as argument and returns a modal operator. Alternatively, one can, as Barringer and Gabbay do in an earlier chapter in this handbook, introduce a unary predicate $p()$ for each proposition p in the original -propositional- language and stipulate that $p(t)$ holds if p is true at time point t. Thus, $p(t)$ is essentially a different notation for $\mathrm{AT}(t)p$. One advantage of the AT operator is that it is easier to see how it can be used in a full first-order logic.

Modal temporal logic inherits its model theory from generic modal logic. The standard model theory for such logics relies on the notion of a possible world, as introduced in this context by Kripke (1963). In Kripke semantics, primitive expressions, such as constants and predicates, are evaluated with respect to a possible world. Non-modal propositions can then be assigned truth value with respect to possible worlds as in standard models for first-order logic. The semantics for modal propositions is defined with the help of an accessibility relation between possible worlds. In modal temporal logic, an intuitive way of defining possible worlds is as points in time, and the accessibility relation between possible worlds as an ordering relation between possible worlds. We then say that for example the proposition Pp is true in a possible world w if there is a possible world w', which is temporally before w and in which p is true. With this in mind, the definition of the semantics for other modal operators is relatively natural.

The only complication to this picture is caused by an introduction of a possible AT operator scheme. Since this operator requires a name for a temporal unit as an argument, the language has to be complicated to include names for such temporal units, and the semantics has to be modified to ensure that such temporal units receive their proper denotation. Obviously, the most appropriate way to deal with this complication is to assign possible worlds as the designation of names for temporal units, and to include an additional clause in the semantics that states that the proposition $\mathrm{AT}(t)p$ is true if p is true in the possible world denoted by t.

2.2 Analysis

We defined a number of representational desiderata on any temporal logic. One of the criteria is the **notational efficiency** (conciseness, naturalness, readability, elegance, ...). Compared to other temporal formalisms discussed in this paper, modal temporal logic scores well on this criterion since the temporal operators produce concise and natural temporal expressions. Another issue is the **modularity** with respect to other knowledge modalities such as knowledge and belief operators. It is straightforward to combine the syntax and semantics of a modal temporal logic with a modal logic to represent, say, knowledge. Syntactically, such a change merely involves adding a knowledge modal operator; semantically, it involves adding an accessibility relation for this new modality. The model theory now contains two accessibility operators, one used for temporal modalities, the other for epistemic modalities.

As far as cleanness of the **ontology** is concerned, the main concern is the notion of a possible world. There is a significant amount of philosophical literature on whether possible worlds are ontologically acceptable or suspect. Without wanting to delve into this literature, it seems to us that a possible world can simply be regarded as a model for a non-modal first order language, and that this makes the notion ontologically unproblematic. There are of course additional arguments about the identity of individuals across possible worlds, but it again seems to us that this problem can be solved relatively easily by insisting that the same set of individuals be used for each possible world.

Where modal temporal logic is less successful is in its ability to represent the various sentences and axioms in our example. Let's consider few statements from our example. To formalize the statement "An offer becomes effective when is received by the offered and continues to be so until it is accepted by the offered or the offer expires" we introduce several predicates. Let $E(x)$ denote "the offer x is effective", $R(x)$ denote "the offer x is received" $A(x)$ denote "the offer x is accepted" and $X(x)$ denote "the offer x expires". The classic since-until tense logic can be used to express the example as

$$\begin{array}{l} \forall\ x_a, y_a, x_o \\ \quad [\ E(o(x_a, y_a, x_o))\ \textsc{Since}\ R(y_a, o(x_a, y_a, x_o)) \wedge \\ \quad E(o(x_a, y_a, x_o))\ \textsc{Until}\ (A(y_a, x_o(x_a, y_a, x_o)) \vee E(x_o(x_a, y_a, x_o)))] \end{array}$$

The problem is that modal temporal logic does not allow one to quantify over occurrences of a particular event. Thus, a proposition like "every time a company makes an offer, it is committed to that offer until it either expires or has been accepted" would be impossible to express.

Although the semantics for modal temporal logics is well understood, it has to be admitted that the implementation of **automated theorem provers** for modal temporal logic is not straightforward. One could of course try to adopt a theorem prover developed for general modal logic. However, such theorem provers in general do not allow for particularly complex accessibility relationships between possible worlds. Most merely allow accessibility relations to be serial, transitive, reflexive or some combination of these. However, such properties are clearly not enough if one were to introduce intervals as one's temporal units. In other words, using a general theorem prover as a reasoning mechanism for modal temporal logic is only likely to be successful if one uses points as one's temporal units. A more promising approach would be to develop theorem provers specifically for temporal modal logic which is a topic of ongoing research as discussed on other chapters in this volume.

3 Temporal Arguments

The oldest and probably most widely used approach to temporal qualification is the method of temporal arguments (*TA*) as introduced in section 1.3. The idea of the temporal arguments approach is to start with a traditional logical theory but to add additional arguments to predicates and function symbols to deal with time. In order to reflect the fact that the domain now contains both "normal individuals" and times, the theory is often formulated as an instance of a *many-sorted first-order logic with equality.*

3.1 Definition

As stated in section 1.1 we assume (i) a given time structure $\langle \mathcal{T}_1, \ldots, \mathcal{T}_{n_t}, \mathcal{F}_{time}, \mathcal{R}_{time} \rangle$ with a FOL time axiomatization , and (ii) a classification of temporal entities $\{\mathcal{E}_1, \ldots, \mathcal{E}_{e_n}\}$ each class accompanied by a temporal incidence axiomatization.

Given these, we define the temporal arguments method as a many-sorted logic with the time sorts $T_1, \ldots, T_{n_t}$, one for each time set, and a number of non-time sorts $U_1, \ldots, U_n$.

Syntax. The vocabulary is composed of the following symbols:

- a set of function symbols $\mathbf{F} = \{f^{\langle D_1,\ldots,D_n \mapsto R\rangle}\}$. If $n = 0$, f denotes a single individual from sort R, otherwise f denotes a function $D_1 \times \ldots \times D_n \mapsto R$ and depending of the nature of the D_i, we distinguish between:
  - *Time functions* $\mathbf{F}_{time}$ whose domain and range are time sorts.
  - *Temporal functions* $\mathbf{F}_t$ whose range is a non-time sort and whose domain include both, time and non-time sorts.
  - *Atemporal functions* $\mathbf{F}_\infty$ whose range is a domain sort and whose domain do not include any time sort.

  Time, temporal and atemporal terms are defined in the usual way.

- a set of predicates $\mathbf{P} = \{P^{\langle D_1,\ldots,D_n\rangle}\}$. If $n = 0$, P denotes a propositional atom, otherwise P denotes a relation defined over $D_1, \ldots, D_n$ and depending on whether D_i are time or a non-time sorts we distinguish between:
  - *Time predicates* $\mathbf{P}_{time}$ whose arguments are all time sorts.
  - *Temporal predicates* $\mathbf{P}_t$ whose arguments include both time and domain sorts.
  - *Atemporal predicates* $\mathbf{P}_\infty$ whose arguments do not include any time sort.

- a set of variable symbols for each sort.

We have three classes of basic formula: atomic temporal formula, atomic atemporal formula and temporal constraints.

Semantics. The semantics is the standard semantics of many-sorted logics. Notice that time gets full ontological status as we have one or more time sorts, yet temporal entities and temporal formulas receive no special treatment.

3.2 Formalizing the Example

Having assumed the models of time and temporal incidence indicated in 1, we define the following **sorts**: T_{point} for time points, T_{int} for time intervals, and T_{span} for time spans or durations, A for agents, O for legal objects, G for trading goods, S for legal status and \$ for money. Our **vocabulary** includes the following symbols:

- a set of constants for each sort: day constants = $\{1/8/04, \texttt{now}, \ldots\}$, time interval constants = $\{3/04, 2004, \ldots\}$, time span constants = $\{3d, 2w, 1y, \ldots\}$, the constant `now`, agent constants = $\{\mathit{john}, \mathit{jane}, \mathit{bco}, \mathit{sco}, \ldots\}$, legal object constants = $\{o_1, o_2, \ldots\}$, etc.

- the following sets of function symbols:
  - $\mathbf{F}_{time} = \{+_T^{\langle T_{\text{point}}, T_{\text{span}} \mapsto T_{\text{point}}\rangle}, -_T^{\langle T_{\text{point}}, T_{\text{point}} \mapsto T_{\text{span}}\rangle}, \ldots\}$
  - $\mathbf{F}_t = \{\mathit{manager}^{\langle T_{\text{int}}, A \mapsto A\rangle}\}$
  - $\mathbf{F}_\infty = \{\mathit{sale}^{\langle G,P \mapsto O\rangle}, \mathit{offer}^{\langle A,A,O,T_{\text{span}} \mapsto O\rangle}\}$

- the following sets of predicates:

- $\mathbf{P}_{time} = \{\leq^{\langle T_{\text{point}}, T_{\text{point}} \rangle}, =^{\langle T_{\text{point}}, T_{\text{point}} \rangle}, \ldots\}$
- $\mathbf{P}_t = \mathbf{P}_{event} \cup \mathbf{P}_{fluent}$
  - $\mathbf{P}_{event} = \{\mathit{Send}^{\langle T_{\text{point}}, A, A, O \rangle}, \mathit{Receive}^{\langle T_{\text{point}}, A, O \rangle}, \mathit{Accept}^{\langle T_{\text{point}}, A, O \rangle}, \mathit{Expire}^{\langle T_{\text{point}}, O \rangle}\}$
  - $\mathbf{P}_{fluent} = \{\mathit{Effective}^{\langle T_{\text{point}}, T_{\text{point}}, O \rangle}, \mathit{Accepted}^{\langle T_{\text{point}}, T_{\text{point}}, O \rangle}, \mathit{Expired}^{\langle T_{\text{point}}, T_{\text{point}}, O \rangle}\}$
- $\mathbf{P}_\infty = \{\mathit{Correct\_form}^{\langle O \rangle}, \leq_{\$}^{\langle P, P \rangle}$ (that denotes the $\leq$ relation between prices)$\}$

- and a set of variable symbols for each sort.

The statements in the example can be formalized as follows:

1. "On 1/4/04 SmallCo sent an offer to BigCo for selling goods g for price p with a 2 weeks expiration interval."
   $\mathit{Send}(1/4/04, \mathit{sco}, \mathit{bco}, \mathit{offer}(\mathit{sco}, \mathit{bco}, \mathit{sale}(g, p), 2w))$

2. "BigCo received the offer three days later and it has been effective since then."
   $\mathit{Receive}(1/4/04 + 3d, \mathit{bco}, \mathit{offer}(\mathit{sco}, \mathit{bco}, \mathit{sale}(g, p), 2w)) \ \wedge$
   $\mathit{Effective}(1/4/04 + 3d, \texttt{now}, \mathit{offer}(\mathit{sco}, \mathit{bco}, \mathit{sale}(g, p), 2w))$

3. "A properly formalized offer becomes effective when it is received by the offered ..."
   $\forall\ t_1 : T_{\text{point}}, x_a, y_a : A, x_o : O, ts : T_{\text{span}},$
   $[\ \mathit{Correct\_form}(\mathit{offer}(x_a, y_a, x_o, ts)) \ \wedge \ \mathit{Receive}(t_1, y_a, \mathit{offer}(x_a, y_a, x_o, ts) \ \rightarrow$
   $\exists t_2 : T_{\text{point}}\ [\ \mathit{Effective}(t_1, t_2, \mathit{offer}(x_a, y_a, x_o, ts) \ \wedge \ t_1 \leq t_2\ ]$
   $]$

4. "... (an effective offer) continues to be so until it is accepted by the offered or the offer expires (as indicated by its expiration interval)."
   $\forall\ t_1, t_2 : T_{\text{point}}, x_a, y_a : A, x_o : O, ts : T_{\text{span}}$
   $[\ \mathit{Effective}(t_1, t_2, \mathit{offer}(x_a, y_a, x_o, ts)) \wedge t_1 \leq t_2 \rightarrow$
   $\exists t_3 : T_{\text{point}}\ [\mathit{Accept}(t_3, y_a, \mathit{offer}(x_a, y_a, x_o, ts)) \wedge t_1 < t_3 \leq t_1 + ts] \ \vee$
   $(t_2 = t_1 + ts \wedge \mathit{Expire}(t2, \mathit{offer}(x_a, y_a, x_o, ts)))$
   $]$

5. "Anybody who makes an offer is committed to the offer as long as the offer is effective."
   $\forall\ t_1, t_2 : T_{\text{point}}, x_a : A$
   $[\ \mathit{Effective}(t_1, t_2, \mathit{offer}(x_a, \_, \_, \_)) \rightarrow \mathit{Committed}(t_1, t_2, x_a, \mathit{offer}(x_a, \_, \_, \_))\ ]$

6. "Anybody who receives an offer is obliged to send a confirmation to the offerer within two days."
   $\forall\ t : T_{\text{point}}, x_a, y_a : A, x_o : O$
   $[\ \mathit{Receive}(t, y_a, x_a, x_o) \rightarrow \mathit{Obliged}(t, t + 2d, y_a, ???)\ ]$

The "???" in the last formula indicates that it is not clear how to express that y_a is obliged to "send a confirmation of x_o to x_a" since in standard FOL we cannot predicate or quantify over propositions[4]. Besides this example, there are few additional general statements whose formalization is interesting to consider:

[4] The reader might come up with the idea of turning temporal predicates into terms in order to be able to take them as proper predicate arguments. This is the idea of temp. reified logics that we discuss below.

1. Time axioms: "The ordering between instants is transitive":
   $\forall\, t_1, t_2, t_3 : T_{\text{point}}\ [\ t_1 \leq t_2 \wedge t_2 \leq t_3 \rightarrow t_1 \leq t_3\ ]$

2. Temporal Incidence axioms such as "Fluents hold homogeneously":
   $\forall\, t_1, t_2, t_3, t_4 : T_{\text{point}}, x_1 : S_1, \ldots, x_n : S_n,$
   $[\ P(t_1, t_2, x_1, \ldots, x_n) \wedge t_1 \leq t_3 \leq t_4 \leq t_2 \wedge t_1 \neq t_4 \rightarrow P(t_3, t_4, x_1, \ldots, x_n)\ ]$

   This an "axiom schema" that is a shorthand for a potentially large set of axioms, one for each fluent predicate P in the language.

The previous examples are instances of relations holding between temporal entities, which can be important in some applications. In common-sense reasoning and planning, for instance, it is important to specify the CAUSE relationship:

> "Whenever an offer is effective it *causes* the agent who made the offer to be committed to it as long as the offer is effective."

Again, it is not clear how to express this piece of knowledge in the method of temporal arguments since it requires the predicate CAUSES to take as argument the proposition $\mathit{Effective}(t_1, t_2, \mathit{offer}(x_a, y_a, x_o, ts))$ which is beyond standard many-sorted FOL. A similar problem arises we attempt to formalize the following properties:

- "Whenever a cause occurs its effects hold."
- "Causes precede their effects."

3.3 Theorem Proving

Defining a temporal logic as a standard many-sorted logic has the advantage that we can use the various deductive systems available for many-sorted logics [6, 24, 12, 5]. However, we must bear in mind that, while many-sorted logics often allow one to delete sortal axioms, such as "All offers are legal documents", the inclusion of a number of time sorts and predicate symbols with a specific meaning (as determined by the properties of the model of time adopted) requires one to add a potentially large number of axioms that capture the nature of the temporal incidence theory. Moreover, it may be impossible to define a set of axioms that completely capture our time structure. For instance, we have taken the "set of integers" as our duration subdomain. But it is well-known that there is no complete axiomatization of the integers in first-order logic if the language includes addition. Therefore, it is important to choose a temporal structure that can be characterized fully in first-order logic, such as *"unbounded linear orders"*, *"totally ordered fields"* or some of the theories discussed in chapter "Theories of Time and Temporal Incidence".

Having a complete axiomatics and therefore a complete proof theory, though, is merely the beginning of the story. Time axioms can be a heavy load for our theorem prover as they often lead to a significant increase in the size of the search space. This problem may lead to the unavoidable effort of developing an specialized temporal theorem prover.

3.4 Analysis

The method of temporal arguments has a number of advantages over other approaches to temporal qualification. First, the **ontology** that one is committed to is relatively straightforward. In addition to "normal" objects, one merely has to add time objects to one's ontology.

Compared to the ontologies that underlie the other approaches to temporal quantification, the ontology is both parsimonious and clean. Moreover, again in contrast with some of the approaches discussed in this chapter, the system does not make any ontological commitments itself, and one is therefore completely free to make the ontology as parsimonious as the application allows.

Second, despite it seeming simplicity, the **expressive power** of languages embodying the temporal arguments approach exceeds that of many other approaches to temporal quantification. The inclusion of additional temporal arguments in predicate and function symbols allows one both to express information about individuals and their properties at specific times and to quantify over times. Moreover, it is straightforward to include purely temporal axioms explicitly in one's theories. However, this is not to say that the method of temporal arguments gives one all the desired expressive power. For example, as we indicated in the previous section, since it stays within the expressive limitations of first-order logic, it is not possible to express temporal incidence properties for all temporal entities in class (fluents, events and so on) or any other property or relation about temporal entities such as "event e at time t causes fluent f to be true at time t".

Third, the **notation** is perhaps not as efficient as some of the alternatives, specifically modal logic. Many of the modal temporal operators are a notational shortcut for existential or universal quantification. For example, the modal operator F provides an existential quantification over future times. Since no such notational shortcuts exist in systems based on the method of temporal arguments, the expression of sentences becomes more tedious in such systems. This is true in particular of sentences that require embedded temporal quantification, such as "The contract will have been signed by then".

Fourth, as we already indicated in the previous section, the fact that the method of temporal arguments is based on a standard first-order logic means that one can use the tried and tested **theorem proving** methods for such systems, which is not the case of methods based on a temporal logic with a non-standard temporal semantics. Moreover, setting up the system as an instance of a multi-sorted logic allows one to take advantage of the more efficient theorem provers developed for such logic. However, it is important to mention that the fact that one is forced to include explicit axioms describing temporal structures in one's theories has detrimental effects on the performance of the actual theorem provers. Many of the additional axioms lead to an combinatorial explosion of the search space and therefore significantly increase the time required to find a proof. For example, some axioms, such as for every point in time, there is a later point in time, are recursive and, unless carefully controlled, lead to an infinite search space. We discuss some approaches that can be used to address some of these issues below.

Finally, since the arguments that are added to the predicate and function symbols denote time, the method of temporal arguments does not easily lend itself to the **modular** inclusion of other modalities, such as epistemic or deontic modalities.

The methods that we discuss below have been developed to overcome some of the shortcomings associated with the method of temporal arguments. One way of increasing the expressive power of the formalism without moving to a higher-order logic is through the addition of some vocabulary and a complication in the ontology. The *temporal token arguments* is one such approach.

4 Temporal Token Arguments

The temporal token argument method (*TTA*) was introduced in the early AI temporal databases such as the *Event Calculus* [11] and Dean's *Time Map Manager* [8] and later presented in [9] in deeper detail. It is based on the simple idea, common in the database community, of introducing a key to identify every tuple in a relation. Here, a tuple of a temporal relation represents an instance of that relation holding at a particular time or time span. Therefore, we introduce a key that identifies a temporal instance of the relation, namely a **temporal token**, which shall receive full ontological status.

4.1 Definition

For a given time structure $\langle \mathcal{T}_1, \ldots, \mathcal{T}_{n_t}, \mathcal{F}_T, \mathcal{R}_T \rangle$ and a given set of *temporal entities* $\{\mathcal{E}_1, \ldots, \mathcal{E}_{n_e}\}$, we define a standard many-sorted first order language with the following *sorts*: one time sort $T_1, \ldots, T_{n_t}$ for each set of time objects, a number of non-time sorts $U_1, \ldots, U_n$ and one token sort $E_1, \ldots, E_{n_e}$ for each temporal entity.

Syntax. It is very similar to the temporal arguments method but instead of having extra time arguments in our temporal predicates, the extra argument is a single *temporal token* term. In turn token terms are taken as arguments by (i) time functions, and (ii) temporal incidence predicates that we introduce below.

The vocabulary is extended accordingly:

- *Function symbols*: In addition to the function symbol sets in temporal arguments, we have a set of **time-token** functions (not to be confused with time functions) that map tokens to their relevant times, for example the function `interval` : $E_{\text{token}} \mapsto T_{\text{int}}$ that returns the time interval associated with a given token.

- *Predicate symbols*: *Temporal predicates* no longer have any time argument, but instead have a single token argument from the sort of the temporal entity denoted by the temporal predicate. Thus, *Effective*$(t_1, t_2,$ *offer*$(\_))$ becomes *Effective*$(tt_1,$ *offer*$(\_))$ where tt_1 is a constant symbol of the new E_{fluent} sort.

  Time and *Atemporal predicates* remain the same. We incorporate one new **Temporal Incidence Predicate (TIP)**[5] for each temporal entity $\mathcal{E}_i$ which takes as sole argument a term of the temporal sort E_i. Given our temporal ontology we shall have 2 TIPs: HOLDS(*fluent token*) expresses that the fluent token holds throughout the time interval associated to it that is denoted by the term `interval`(tt_1) and OCCURS(*event token*) for event occurrence.

Semantics. The standard many-sorted first-order semantics is preserved with both time domains, non-time domains and temporal token domains with the usual interpretation of function and predicate symbols. Time and temporal incidence theories are incorporated as a set of first order axioms.

[5] TIPs actually are atemporal if we look at the definition we have given to this term.

4.2 Formalizing the Example

We illustrate the approach by formalizing the example. We make the same assumptions as before and we will frequently refer to the formalization of this example in the temporal arguments method.

In addition to the **sorts** defined in the *TA* example we introduce sorts for tokens of each temporal entity: E_{event} for event tokens and E_{fluent} for fluent tokens. In turn, our vocabulary will include event token constants and fluent token constants. Besides the usual functions, we have the following **time-token functions**: $\texttt{time}(token)$ denotes the time of an instantaneous temporal token, $\texttt{begin}(token)$ denotes the initial instant of a temporal token, $\texttt{end}(token)$ denotes the ending instant and $\texttt{interval}(token)$ the time interval of the token.

In addition to the time and atemporal predicates from the previous formalization, the *temporal predicates* now are as follows:

- *Events*: $Send^{\langle E_{\text{event}},A,A,O\rangle}$ (where the last argument denotes the event token of this particular send event), $Receive^{\langle E_{\text{event}},A,A,O\rangle}$, and $Accept^{\langle E_{\text{event}},A,O\rangle}$.
- *Fluents*: $Effective^{\langle E_{\text{fluent}},O\rangle}$ (where the first argument denotes the fluent token of a particular period where the legal object O is effective), $Accepted^{\langle E_{\text{fluent}},O\rangle}$ and $Expired^{\langle E_{\text{fluent}},O\rangle}$.

As in the *TA* method, we have four classes of basic formula: atomic atemporal formula, atomic temporal formula, temporal constraints and temporal incidence formula.
The statements in the example can be formalized as follows:

1. "On 1/4/04, SmallCo sent an offer to BigCo for selling goods g for price p with a 2 weeks expiration interval."
   $Send(s_1, sco, bco, offer(sco, bco, sale(g,p), 2w)) \ \wedge\ \textsc{Occurs}(s_1) \wedge \texttt{time}(s_1, 1/4/04)$

2. "BigCo received the offer three days later and it has been effective since then."
   $Receive(r_1, bco, offer(sco, bco, sale(g,p), 2w)) \ \wedge \textsc{Occurs}(r_1)\ \wedge$
   $\texttt{time}(r_1) = 1/4/04 + 3d\ \wedge$
   $Effective(e_1, offer(sco, bco, sale(g,p), 2w)) \ \wedge\ \textsc{Holds}(e_1)\ \wedge$
   $\texttt{time}(r_1) = \texttt{begin}(e_1)\ \wedge\ \texttt{end}(e_1) = now$

3. "A properly formalized offer becomes effective when is received by the offered ..."
   $\forall\ tt_1 : E_{\text{event}}, ts : T_{\text{span}}, x_a, y_a : A,\ x_o : O$
   $[\ Correct\_form(offer(x_a, y_a, x_o, ts))\ \wedge$
   $Receive(tt_1, y_a, x_a, offer(x_a, y_a, x_o, ts))\ \wedge\ \textsc{Occurs}(tt_1)\ \rightarrow$
   $\exists\ tt_2 : E_{\text{fluent}}$
   $[\ Effective(tt_2, offer(x_a, y_a, x_o, ts))\ \wedge\ \textsc{Holds}(tt_2)\ \wedge\ tt_1\ \texttt{Meets}\ tt_2\ ]$
   $]$

4. "... (an effective offer) continues to be so until it is accepted by the offered or the offer expires (as indicated by its expiration interval)."
   $\forall\ tt_1 : E_{\text{fluent}}, x_a, y_a : A,\ x_o : O,\ ts : T_{\text{span}}$
   $[\ Effective(tt_1, offer(x_a, y_a, x_o, ts))\ \wedge\ \textsc{Holds}(tt_1)\ \rightarrow$
   $\exists tt_2 : E_{\text{event}}$
   $[Accept(tt_2, y_a, offer(x_a, y_a, x_o, ts))\ \wedge\ \textsc{Occurs}(tt_2) \wedge$
   $\texttt{begin}(tt_1) < \texttt{time}(tt_2) \leq \texttt{begin}(tt_1) + ts]$

$\vee$
$(\texttt{end}(tt_1) = \texttt{begin}(tt_1) + ts \ \wedge$
$\exists tt_2 : E_{\text{event}}$
$[\mathit{Expire}(tt_2, \mathit{offer}(x_a, y_a, x_o, ts)) \ \wedge \ \textsc{Occurs}(tt_2) \wedge$
$\texttt{end}(tt_1) = \texttt{time}(tt_2)])$
$]$

5. "Anybody who makes an offer is committed to the offer as long as the offer is effective."
$\forall\ tt_1 : E_{\text{fluent}}, x_a, y_a : A, x_o : O, ts : T_{\text{span}}$
$[\ \mathit{Effective}(tt_1, \mathit{offer}(x_a, y_a, x_o, ts)) \wedge \textsc{Holds}(tt_1) \ \rightarrow$
$\exists\ tt_2 : E_{\text{fluent}}$
$[\ \mathit{Committed}(tt_2, x_a, \mathit{offer}(x_a, y_a, x_o, ts)) \wedge \textsc{Holds}(tt_2) \ \wedge$
$\texttt{interval}(tt_1) = \texttt{interval}(tt_2)\ ]\ ]$

6. "Anybody who receives an offer is obliged to send a confirmation to the offerer within two days."
$\forall\ tt_1 : E_{\text{event}}, x_a, y_a : A, x_o : O, ts : T_{\text{span}}$
$[\ \mathit{Receive}(tt_1, y_a, \mathit{offer}(x_a, y_a, x_o, ts)) \wedge \textsc{Occurs}(tt_1) \rightarrow$
$\exists tt_2 : E_{\text{event}}$
$[\mathit{Obliged}(x_a, tt_2) \wedge \mathit{Send}(tt_2, y_a, x_a, \mathit{conf}(x_o)) \ \wedge$
$\texttt{time}(tt_1) \leq \texttt{time}(tt_2) \leq \texttt{time}(tt_1) + 2d\ ]\ ]$

Observe that we express that x_a is obliged to a temporal proposition by using a temporal token of that proposition. In general, the additional flexibility of temporal tokens allows us (i) to talk about temporal occurrences that may or may not happen, and (ii) to express that an agent is obliged to that event. This is not possible in the *TA* method.
Let's see how the more general statements are formalized:

- Time axioms are expressed as usual:
  $\forall\ t_1, t_2, t_3 : T_{\text{point}}[\ t_1 \leq t_2 \wedge t_2 \leq t_3 \rightarrow t_1 \leq t_3\ ]$

- Temporal Incidence axioms become more compact since we can quantify over all the instances of a given entity (e.g. all fluents) independently of their particular meaning. It is no longer necessary to have an "axiom schema". For instance the "homogeneity of fluent holding" is stated by:
  $\forall\ tt : E_{\text{fluent}}, I : T_{\text{int}}\ [\ \mathit{holds}(tt) \wedge I \subseteq \texttt{interval}(tt) \rightarrow \textsc{Holds}_{on}(tt, I)\ ]$

- "It is necessary for an offer to be properly written to be effective".
  $\forall\ tt : E_{\text{fluent}}, x_o : O\ [\mathit{Effective}(tt, x_o) \wedge \textsc{Holds}(tt) \rightarrow \mathit{Correct\_form}(x_o)]$

- "Whenever an offer is effective it *causes* the agent who made the offer to be committed to it for as long as the offer is effective."
  $\forall\ tt_1 : E_{\text{fluent}}, x_a, y_a : A, x_o : O, ts : T_{\text{span}}$
  $[\ \mathit{Effective}(tt_1, \mathit{offer}(x_a, y_a, x_o, ts)) \wedge \textsc{Holds}(tt_1) \rightarrow$
  $\exists tt_2 : E_{\text{fluent}}$
  $[\ \textsc{Cause}(tt_1, tt_2) \wedge \mathit{Committed}(tt_2, \mathit{offer}(x_a, y_a, x_o, ts)) \wedge \texttt{interval}(tt_1, tt_2)\ ]\ ]$

- "Whenever a cause occurs its effects hold."
  $\forall\ tt_1 : E_{\text{event}}, tt_2 : E_{\text{fluent}}$
  $[\textsc{Occurs}(tt_1) \wedge \textsc{Cause}(tt_1, tt_2) \rightarrow \textsc{Holds}(tt_2)]$

- "Causes precede their effects."
  $\forall\ tt_1 : E_{\text{event}}, tt_2 : E_{\text{fluent}}$
  $[\ \textsc{Cause}(tt_1, tt_2) \rightarrow (\textsc{Occurs}(tt_1) \rightarrow \textsc{Holds}(tt_2) \wedge \mathtt{time}(tt_1) \leq \mathtt{begin}(tt_2)\ ]$

4.2.1 Token Incidence Theory

The specific semantics of temporal tokens may yield some additional temporal incidence axioms. An example is the so-called "maximality of fluent tokens". For practical reasons, one is interested in adopting the following convention:

> "A fluent token denotes a *maximal* piece of time where that fluent is true."

A consequence of this is the following property "Any two intervals associated with the same fluent are either identical or disjoint." Thus, in practice in can be interesting to define some additional incidence predicates such as $\textsc{Holds}_{at}{}^2$ and $\textsc{Holds}_{on}{}^2$ which are shorthands for

$$\textsc{Holds}_{at}(\mathit{fluent}, t) \equiv \exists f : E_{\text{fluent}}\ (\mathit{fluent}(f) \wedge \textsc{Holds}(f) \wedge i \in \mathtt{interval}(f))$$
$$\textsc{Holds}_{on}(\mathit{fluent}, I) \equiv \exists f : E_{\text{fluent}}\ (\mathit{fluent}(f) \wedge \textsc{Holds}(f) \wedge I \subseteq \mathtt{interval}(f))$$

respectively, where f is a variable of the *fluent token* sort E_{fluent} and $\mathit{fluent}(f)$ denotes the atomic proposition fluent with the extra temporal token argument f.

4.3 Analysis

The temporal token arguments method has several advantages. The extra objects (the temporal tokens) introduced in the language gives the **notation** an increased flexibility that helps overcome some of the expressiveness problems that we identified in the temporal arguments method: First, as temporal tokens are used as argument of other predicates it is useful to express nested temporal references as shown by the example. Second, different levels of time are supported by diversifying the time-token functions. For instance, we may have $\mathtt{begin}_{\mathrm{V}}(tt_1)$ to refer to *valid time* and $\mathtt{begin}_{\mathrm{t}}(tt_1)$ to refer to *transaction time*. Third, at the implementation level, a different temporal constraint network instance is maintained for each time level. Every temporal term will be mapped to a node in its corresponding constraint network.

However, the increased notation flexibility causes the notation to be more baroque and sometimes awkward (compare the formalization of our example here with the formalizations obtained by other methods). To improve notational *conciseness* we can define some syntactic sugar that allows the omission of token symbols whenever they are not strictly necessary.

Another advantage of this approach is its **modularity**. A clear separation is made between the temporal and other information as a atomic temporal formulas are linked to time through time-token functions like `begin` and `end`. However, token symbols can also be used as the link to other modalities as the deontic modalities of commitment and obligation illustrated by the example.

5 Temporal Reification

Temporal reification (*TR*) was motivated by the desire to extend the expressive power of the temporal arguments approach while remaining within the limits of first order logic. It is achieved by: (i) complicating the underlying ontology and (ii) representing temporal propositions as terms in order to be able to predicate and quantify over them.

In essence, in reified temporal logic, both time objects and temporal entities receive full ontological status and one introduces in the language terms referring to them. The *Temporal Incidence Predicates* are used to associate a temporal entity with its time of occurrence and allow a direct and natural axiomatization of the given temporal incidence properties, as illustrated in the example below.

Syntax. Reified temporal logics are in fact relatively straightforward to construct from a standard first order language. First, it is useful to move to a sorted logic in which we make a distinction between temporal entities, normal individuals and temporal units. Second, for each n-place function symbol in the first order language, one introduces a corresponding n-place function symbol in the reified language. Its sortal signature is that it maps n normal individuals into a normal individual. For each n-place predicate in the original language, one also introduces a n-place function symbol in the language. However, its sortal signature is different. It takes as input n normal individuals and maps them into a temporal entity.

Semantics. Interestingly, not many authors worried about providing a clear model-theoretic semantics for their formalism, either because they were not interested in doing so, or because they believed that reified temporal logic would simply inherit its semantics from first order predicate calculus. It was not until [19] that the semantics of reified temporal logics became an issue. Shoham observed that reified temporal logic are very similar to formalizations of the model theory for modal temporal logic in a first order logic and proposed to formulate the semantics for reified temporal logic in these terms. It is not clear that the actual framework proposed by Shoham actually achieved this. For example, [23] argue that Shoham's formalism is more appropriately regarded as being a hybrid between a modal temporal logic and a system in the tradition of the temporal arguments method. As a matter of fact, Shoham's is subsumed by the *TA* method[3]. Nevertheless, Shoham's insight was the inspiration for Reichgelt [16] who indeed formulated a reified temporal logic.

5.1 Formalizing the Example

Let's see how this method is used to formalize our example. We make the same assumptions and we shall be continuously referring to the formalization of this example made with the temporal arguments method.

Besides the **sorts** T_i for time instants, T_I for time intervals, A for agents, etc. we now have additional sorts, one for each temporal entity: E_{event} for events and E_{fluent} for fluents. Our **vocabulary** is composed of:

- For each sort, a set of constant symbols, including event constants and fluent constants.
- We have time, temporal and atemporal function symbols as in the temporal arguments approach except that the set of temporal functions (where we have functions like $\mathit{offer}^{\langle A,A,O,T \mapsto O \rangle}$) is extended with new temporal functions produced by temporal reification, one for each temporal relation (which in the temporal arguments is represented by a temporal predicate):
  - $E_{\text{event}} = \{\mathit{Send}^{\langle A,A,O \mapsto E_{\text{event}} \rangle}, \mathit{Receive}^{\langle A,O \mapsto E_{\text{event}} \rangle}, \mathit{Accept}^{\langle A,O \mapsto E_{\text{event}} \rangle}, \mathit{Expire}^{\langle O \mapsto E_{\text{event}} \rangle}\}$
  - $E_{\text{fluent}} = \{\mathit{Effective}^{\langle O \mapsto E_{\text{fluent}} \rangle}, \ldots\}$

- the following sets of predicates:
  - $\mathbf{P}_{time} = \leq^{\langle T,T \rangle}, =^{\langle T,T \rangle}, \ldots$
  - $\mathbf{P}_{\infty} = \leq_{\$}^{\langle P,P \rangle}$ (that denotes the $\leq$ relation between prices).
- and a set of variable symbols for each sort.

The statements in the example may be formalized as follows:

1. "On 1/4/04, SmallCo sent an offer to BigCo for selling goods g for price p with a 2 weeks expiration interval."
   $\textsc{Occurs}(1/4/04, \mathit{Send}(\mathit{sco}, \mathit{bco}, \mathit{offer}(\mathit{sco}, \mathit{bco}, \mathit{sale}(g, p), 2w)))$

2. "BigCo received the offer three days later and it has been effective since then."
   $\textsc{Occurs}(1/4/04 + 3d, \mathit{Receive}(\mathit{bco}, \mathit{offer}(\mathit{sco}, \mathit{bco}, \mathit{sale}(g, p), 2w))) \wedge$
   $\textsc{Holds}(1/4/04 + 3d, \mathit{now}, \mathit{Effective}(\mathit{offer}(\mathit{sco}, \mathit{bco}, \mathit{sale}(g, p), 2w)))$

3. "A properly formalized offer becomes effective when is received by the offered ..."
   $\forall\ t_1 : T_{\text{point}}, x_a, y_a : A,$
   $[\ \mathit{Correct\_form}(\mathit{offer}(x_a, y_a, \_, \_))\ \wedge \textsc{Occurs}(t_1, \mathit{Receive}(y_a, \mathit{offer}(x_a, y_a, \_, \_)))\ \rightarrow$
   $\exists t_2 : T_{\text{point}}\ [\ \textsc{Holds}(t_1, t_2, \mathit{Effective}(\mathit{offer}(x_a, y_a, \_, \_)))]$
   $]$

4. "...(an effective offer) continues to be so until it is accepted by the offered or the offer expires (as indicated by its expiration interval)."
   $\forall\ t_1, t_2 : T_{\text{point}}, x_a, y_a : A, x_o : O, ts : T_{\text{span}}$
   $[\ \textsc{Holds}(t_1, t_2, \mathit{Effective}(\mathit{offer}(x_a, y_a, x_o, ts)))\ \rightarrow$
   $\exists t_3 : T_{\text{point}}\ [t_1 < t_3 \leq t_1 + ts \wedge \textsc{Occurs}(t_3, \mathit{Accept}(y_a, x_o))]\ \vee$
   $(t_2 = t_1 + ts \wedge \textsc{Occurs}(t_2, \mathit{Expire}(offer(x_a, y_a, x_o, ts))))$
   $]$

5. "Anybody who makes an offer is committed to the offer as long as the offer is effective."
   $\forall\ t_1, t_2 : T_{\text{point}}, x_a : A, x_o : O$
   $[\ \textsc{Holds}(t_1, t_2, \mathit{Effective}(\mathit{offer}(x_a, \_, \_, \_))) \rightarrow$
   $\textsc{Holds}(t_1, t_2, \mathit{Committed}(x_a, \mathit{offer}(x_a, \_, \_, \_)))\ ]$

6. "Anybody who receives an offer is obliged to send a confirmation to the offerer within two days."
   $\forall\ t : T_{\text{point}}, x_a, y_a : A, x_o : O,$
   $[\textsc{Occurs}(t, \mathit{Receive}(y_a, \mathit{offer}(x_a, y_a, x_o, \_))) \rightarrow$
   $\textsc{Holds}(t, t + 2d, \mathit{Obliged}(y_a, send(y_a, x_a, \mathit{conf}(\mathit{offer}(x_a, y_a, x_o, \_)))))]$

Now last formula is legal but still the formalization we have is not that clear. It expresses that the obligation holds between t and $t+2d$ but indeed it is "sending the confirmation" that must be between t and $t+2d$. Next, let's see how the more general statements are formalized:

- Time axioms are expressed as usual:
  $\forall\ t_1, t_2, t_3 : T_{\text{point}}\ [\ t_1 \leq t_2 \wedge t_2 \leq t_3 \rightarrow t_1 \leq t_3\ ]$

- Temporal incidence axioms become more compact since we can quantify over all the instances of a given entity (e.g. all fluents) independently of their particular meaning (and it is no longer necessary to have an "axiom schema"). For instance the "homogeneity of fluent holding" is stated as:
  $\forall\ t_1, t_2, t_3, t_4 : T_{\text{point}}, f : E_{\text{fluent}}$
  $[\ \textsc{Holds}(t_1, t_2, f) \wedge t_1 \leq t_3 \leq t_4 \leq t_2 \wedge t_1 \neq t_4 \rightarrow \textsc{Holds}(t_3, t_4, f)\ ]$

- "It is necessary for an offer to be properly written to be effective".
  $\forall\ t, t' : T_{\text{point}}, x_o : O\ \ [\textsc{Holds}(\mathit{Effective}(t, t', x_o)) \rightarrow \mathit{Correct\_form}(x_o)]$

- "Whenever an offer is effective it *causes* the agent who made the offer to be committed to it for as long as the offer is effective."
  $\forall\ t_1, t_2 : T_{\text{point}}, x_a, y_a : A, x_o : O, ts : T_{\text{span}}$
  $[\ \textsc{Cause}(\mathit{Effective}(t_1, t_2, \mathit{offer}(x_a, y_a, x_o, ts)), \mathit{Committed}(t_1, t_2, x_a, \mathit{offer}(x_a, y_a, x_o, ts)))\ ]$

- "Whenever a cause occurs its effects hold."
  $\forall\ e : E_{\text{event}}, f : E_{\text{fluent}}\ [\ \textsc{Occurs}(e) \wedge \textsc{Cause}(e, f) \rightarrow \textsc{Holds}(f)\ ]$

- "Causes precede their effects."
  $\forall\ e : E_{\text{event}}, f : E_{\text{fluent}}$
  $[\ \textsc{Cause}(e, f) \rightarrow (\textsc{Occurs}(e) \rightarrow \textsc{Holds}(f) \wedge \texttt{time}(e) < \texttt{begin}(f))\ ]$

5.2 Full Temporal Reified Logic

Implicitly in the previous section we have restricted ourselves to reification of atomic propositions, but we can push it further and reify also non-atomic propositions (as first discussed in [14, 2]). This can be motivated by statements like the following:

1. "The offer ...has been sent but not effective from t_1 to t_2". $\textsc{Holds}(t_1, t_2, \mathit{Sent}(o_1) \wedge \neg \mathit{Effective}(o_1)$

2. "From t_1 to t_2 all offers offered by agent a_1 have been frozen." $\textsc{Holds}(t_1, t_2, \forall x_a : A, x_o : O, ts : T_{\text{span}}\ [frozen(\mathit{offer}(a_1, y_a, x_o, ts))]$

3. "As of 1/may/04, when an offer is sent the offerer will have to pay a tax within the next 3 days." $\textsc{Holds}(1/may/04, +\infty, \forall x_a, y_a : A, x_o : O, ts : T_{\text{span}}\ [\mathit{send}(x_a, y_a, \mathit{offer}(x_a, y_a, x_o, ts)) \rightarrow \mathit{Obliga}$

Reichgelt's reified temporal logic takes as its starting point modal temporal logic, and the observation that the semantics for such logics can itself relatively straightforward be formalized in a first-order language. This language, however, becomes rather baroque as it needs to include terms to refer both to semantic entities and terms to refer to the expressions in the modal temporal logic. Thus, a full reified logic would need to codify such statements as "Fp(a) is true at time t if and only if there is a time t' later than t at which the individual denoted by a is an element of the set denoted by P" and this requires the full reified logic to have expressions to refer to times ("t, t' "), expressions to refer to individuals ("the individual denoted by a") and denotations of predicates ("the set denoted by P"), as well as expressions to refer to expressions in the modal temporal logic that is used as its starting point ("the expression a"). The semantics for a full reified logic becomes correspondingly complex, as it needs to include normal individuals and points in time, as well as entities corresponding to the linguistic entities that make up the underlying modal temporal logic. Reichgelt's logic is therefore more of academic interest, rather than of any practical use. However, the system shows that one can indeed use Shoham's proposal to regard reified temporal logics as a formalization of the semantics of modal temporal logic in a complicated, sorted but classical first-order logic.

5.3 Advantages and Shortcomings of Temporal Reified method

As illustrated by the example, the temporal reification method provides a fairly natural and efficient notation and an expressive power clearly superior than temporal arguments as it can talk and quantify over temporal relations satisfactorily.

However temporal reified approaches have been criticized on two different direction. On the one hand, because the **ontologies** they commit one to. In the example $\text{OCCURS}(1/4/04+3d, \textit{Receive}(\textit{bco}, \textit{offer}(\textit{sco}, \textit{bco}, \textit{sale}(g,p), 2w))) \wedge$ $\text{HOLDS}(1/4/04 + 3d, now, \textit{Effective}(\textit{offer}(\textit{sco}, \textit{bco}, \textit{sale}(g,p), 2w)))$ we observe that, in both cases, the non-time arguments to the temporal incidence predicate stand for a type of event or fluent, respectively. There are two objections against the introduction of event and state types. The first is ontological. Thus, taking his lead from [7], and following a long tradition in ontology, A. Galton [9] argues that a logic which forces one to reify event *tokens* instead of event *types*, would be preferable on ontological grounds. Using Occam's razor, Galton argues that one should not multiply the entities in one's ontology without need, and that, unless one is a die-hard Platonist, one would prefer an ontology based on particulars rather than universals. A second argument against the introduction of types is that the resulting logic may have expressiveness shortcomings. Haugh [10] talks about the "individuation and counting of the events of a particular type". One cannot, for instance, refer to the set of multiple effects originated by a single event causing them. Also, one cannot quantify over causes and the related set of the effects each produces in order to assert general constraints between them.

On the other hand, temporal reification has been criticized as an unnecessary technical complication, specially in the case that it is not defined as a standard many-sorted logic and we have to develop a new model theory and a proof theory that is complete for it. Some researchers look at the *temporal token arguments* method a the ideal alternative since it avoid both criticisms and seem to retain the expressiveness adavantages, in particular in quatifying over predicates as shown in the *TTA* section.

6 Temporal Token Reification

This method is motivated by the attempt of achieving the expressiveness advantages of temporal reification (while staying in standard FOL) and the ontological and technical advantages of temporal tokens shown by the temporal token arguments approach which avoids having to reify temporal types.

The primary intuition behind *Temporal Token Reification* (*TTR*) is that one reifies temporal tokens rather than temporal types. However rather than making names for event tokens an additional argument to a predicate (like in the temporal token arguments approach), it proposes to introduce "meaningful" names for temporal tokens. This allows one to talk and quantify about "parts of a token" as well as over all tokens and thus express express general temporal properties.

6.1 Definition

The logical language of *TTR* is a many-sorted FOL with same sorts that *TTA* : $T_1, \ldots, T_{n_t}$, one for each time set, a number of non-time sorts $U_1, \ldots, U_n$ and one token sort $E_1, \ldots, E_{n_e}$ for each temporal entity.

Syntax. The vocabulary is is defined accordingly to the ideas above.

- *Function symbols:* In addition to the time and atemporal function symbols of *TTA* , we have On the one hand, we have a $m+n$-place function symbol for each n-place temporal relation, with its first m arguments being of a time sort and its last n arguments being of some non-time or token sort and its output being either of type E_i.

  On the other hand, we have the usual *time-token function* symbols, whose input argument is of sort E_i and whose output argument is of sort T_j. For instance, `begin` denotes the starting point of a temporal token and their definition is straightforward. Thus

  $$\texttt{begin}(f(\ldots, t, t')) = t'$$

  where $f(\ldots, t, t')$ is a term referring to a temporal token.

  Finally, the language contains the 1-place function symbol TYPE. It takes as argument the name of a temporal token and returns a function from pairs of points in time into the set of event or state tokens respectively. Hence,

  $$\textsc{Type}(f(\ldots, t, t'))$$

  is basically syntactic sugar for

  $$\lambda x \lambda y f(\ldots, x, y)$$

- *Predicate symbols:* As *TTA* , *TTR* makes *TIPs* 1-place. It contains one of them for each E_i with its only argument being the name for an temporal entity. For instance, the predicates HOLDS or OCCURS simply state that a fluent token indeed holds, or that an event token indeed occurs.

Semantics. The semantics of the *TTR* is relatively straightforward as well and *TTR* function and predicate symbols are mapped onto the appropriate functions and relations respecting the signature of the symbol. Additionally, there are a few conditions that we need to impose on our models in order to reflect intuitions about the nature of time and the nature of the temporal entities.

6.2 Formalizing the Example

The **sorts** and the vocabulary is as in the temporal reification with the following additions:

- $\mathbf{F}_{time} = \{\texttt{end}^{\langle E_{\text{fluent}}, T_{\text{point}}, T_{\text{point}} \mapsto T_{\text{point}} \rangle}, \texttt{begin}^{\langle E_{\text{fluent}}, T_{\text{point}}, T_{\text{point}} \mapsto T_{\text{point}} \rangle}\}$ where $f(\ldots, t, t')$ is a term referring to a fluent-token.

- $\mathbf{F}_t = \mathbf{F}_{event} \cup \mathbf{F}_{fluent}$
  - $\mathbf{F}_{event} = \{Send^{\langle T_{\text{point}}, A, A, O \mapsto E_{\text{event}} \rangle}, Receive^{\langle T_{\text{point}}, A, O \mapsto E_{\text{event}} \rangle}, Accept^{\langle T_{\text{point}}, A, O \mapsto E_{\text{event}} \rangle}, Expire^{\langle T_{\text{point}}, O \mapsto E_{\text{event}} \rangle}\}$
  - $\mathbf{F}_{fluent} = \{Effective^{\langle O, T_{\text{point}}, T_{\text{point}} \mapsto E_{\text{fluent}} \rangle}\}$
- $\mathbf{P}_t = \emptyset$
- $\mathbf{P}_\infty = \{\leq_{\$}^{\langle P,P \rangle}\}$ (that denotes the $\leq$ relation between prices).
- and a set of variable symbols for each sort.

The statements in the example can be formalized as follows:

1. "On 1/4/04, SmallCo sent an offer for selling goods g to BigCo for price p with a 2 weeks expiration interval." $\textsc{Occurs}(Send(1/4/04, sco, bco, offer(sco, bco, sale(g,p), 2w)))$
2. "BigCo received the offer three days later and it has been effective since then."
   $\textsc{Occurs}(Receive(1/4/04 + 3d, bco, offer(sco, bco, sale(g,p), 2w))) \;\wedge$
   $\textsc{Holds}(Effective(1/4/04 + 3d, \texttt{now}, offer(sco, bco, sale(g,p), 2w)))$
3. "A properly formalized offer becomes effective when is received by the offered..."
   $\forall\, t_1 : T_{\text{point}}, x_a, y_a : A, x_o : O, ts : T_{\text{span}}$
   $[\; Correct\_form(offer(x_a, y_a, x_o, ts)) \wedge$
   $\textsc{Occurs}(Receive(t_1, y_a, offer(x_a, y_a, x_o, ts))) \rightarrow$
   $\exists t_2\; [\textsc{Holds}(Effective(t_1, t_2, offer(x_a, y_a, x_o, ts))) \wedge t_1 \leq t_2] \;]$
4. "... (an effective offer) continues to be so until it is accepted by the offered or the offer expires (as indicated by its expiration interval)."
   $\forall\, t_1, t_2 : T_{\text{point}}, x_a, y_a : A, x_o : O, ts : T_{\text{span}}$
   $[\; \textsc{Holds}(Effective(t_1, t_2, offer(x_a, y_a, x_o, ts))) \wedge t_1 \leq t_2 \rightarrow$
   $\exists t_3 : T_{\text{point}} [Accept(t_3, y_a, offer(x_a, y_a, x_o, ts)) \wedge t_1 < t_3 \leq t_1 + ts] \;\vee$
   $(t_2 = t_1 + ts \wedge \textsc{Occurs}(Expire(t2, offer(x_a, y_a, x_o, ts)))) \;]$
5. "Anybody who makes an offer is committed to the offer as long as the offer is effective."
   $\forall\, t_1, t_2 : T_{\text{point}}, x_a : A$
   $[\textsc{Holds}(Effective(t_1, t_2, offer(x_a, \_, \_, \_))) \rightarrow$
   $\textsc{Occurs}(Committed(t_1, t_2, x_a, offer(x_a, \_, \_, \_)))]$
6. "Anybody who receives an offer is obliged to send a confirmation to the offerer within two days."
   $\forall\, t_1 : T_{\text{point}}, x_a : A, x_o : O,$
   $[\textsc{Occurs}(Receive(t_1, y_a, offer(x_a, y_a, x_o, \_))) \rightarrow$
   $\textsc{Holds}(Obliged(t, t + 2d, y_a, send(y_a, conf(offer(x_a, y_a, x_o, \_)))))$
   $]$

Let's look at the additional statements:

- Time axioms: "The ordering between instants is transitive":
  $\forall\, t_1, t_2, t_3 : T_{\text{point}} \;[\; t_1 \leq t_2 \wedge t_2 \leq t_3 \rightarrow t_1 \leq t_3 \;]$

- Temporal Incidence axioms such as "Fluents hold homogeneously":
  $\forall f : E_{\text{fluent}}, t_1, t_2, t_3, t_4 : T_{\text{point}}$
  $[\ \textsc{Holds}(\textsc{Type}(f)(< t_1, t_2 >)) \wedge t_1 \leq t_3 \leq t_4 \leq t_2 \wedge t_1 \neq t_4 \rightarrow$
  $\textsc{Holds}(\textsc{Type}(f)(< t_3, t_4 >))\ ]$

- "It is necessary for an offer to be properly written to be effective".
  $\forall\ t_1, t_2 : T_{\text{point}}, x_o : O$
  $[\textsc{Holds}(\mathit{Effective}(t_1, t_2, x_o)) \rightarrow \mathit{Correct\_form}(x_o)]$

- "Whenever an offer is effective it *causes* the agent who made the offer to be committed to it for as long as the offer is effective."
  $\forall\ t_1, t_2 : T_{\text{point}}, x_a, y_a : A, x_o : O, ts : T_{\text{span}}$
  $[\textsc{Cause}(\mathit{Effective}(t_1, t_2, \mathit{offer}(x_a, y_a, x_o, ts)), \mathit{Committed}(t_1, t_2, x_a, \mathit{offer}(x_a, y_a, x_o, ts)))]$

- "Whenever a cause occurs its effects hold."
  $\forall\ e : E_{\text{event}}, f : E_{\text{fluent}}$
  $[\ \textsc{Occurs}(e) \wedge \textsc{Cause}(e, f) \rightarrow \textsc{Holds}(f)\ ]$

- "Causes precede their effects."
  $\forall\ e : E_{\text{event}}, f : E_{\text{fluent}}$
  $[\ \textsc{Cause}(e, f) \rightarrow (\textsc{Occurs}(e) \rightarrow \textsc{Holds}(f) \wedge \mathtt{time}(e) \leq \mathtt{begin}(f))\ ]$

7 Conclusion

In this chapter we have identified the relevant issues around the temporal qualification method which is central in the definition of a temporal reasoning system in AI. We have described the most relevant temporal qualification methods, illustrated them with a rich example and analysed advantages and shortcomings with respect to a number of representational and reasoning efficiency criteria. The various methods are schematically presented in figure 1.

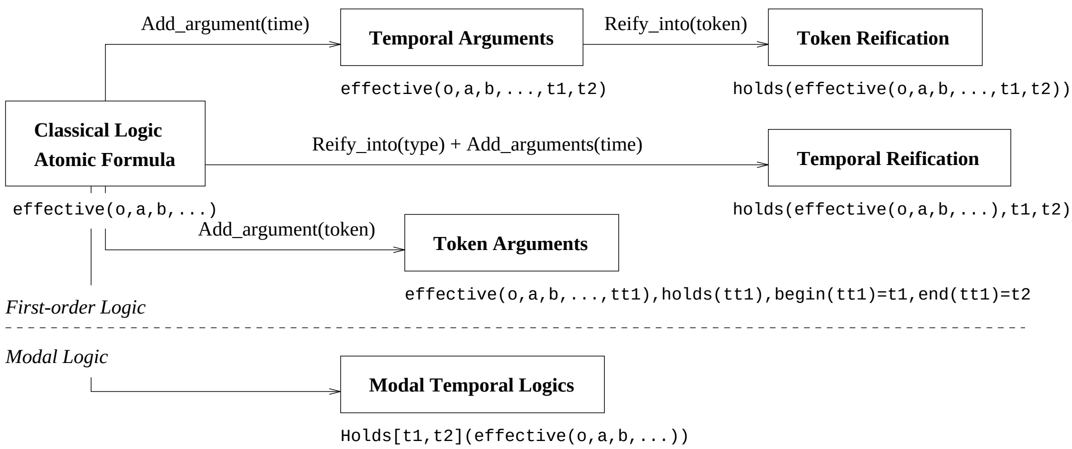

Figure 1: Temporal qualification methods in AI.

Temporal arguments is the classical, easy to define method that turns out to be more expressive that what has traditionally been recognized. It is enough for many applications except for those where one needs to represent nested temporal references or we needs to quantify over temporal propositions. In fact, the subsequent methods are a response to this limitation in a more or less sophisticated manner. *Temporal Token Arguments* allows a "sort of" quantification while the language is a mere temporal arguments approach by moving to a token-based ontology and introducing names for temporal token in the language which provides a good deal of represenation flexibility. The other two approaches are based on reification which is a natural way to talk and quantify over temporal propositions. The increased expressiveness allows one to express statements like "receiving an offer causes to be obliged to send a confirmation" or " causes never preced their effects" which is not possible in the temporal argumentn method.

Technically the temporal reification methods are not that complex. It can be complex if one preferes to define a non-standard semantics for that, but it is not strictly necessary: some temporal reified logics can be defined as a many-sorted logic with the appropriate time and temporal incidence axiomatizations. In such case, it is important to be aware that these axioms can be a source of high inefficiency for the theorem prover. Things can get very complicated, though, if we aim at supporting the reification of non-atomic formula as shown in [15]. It is not clear what is the practical interest of this and it is not known what is the relationship between such logic and temporal modal logics defined over a first order language.

References

[1] I. Ahn. Towards an implementation of database management systems with temporal support. In *Proc. of International Conference on Data Engineering*, pages 374–381, 1989.

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

[3] F. Bacchus, J. Tenenberg, and J. Koomen. A non-reified temporal logic. *Artificial Intelligence*, 52:87–108, 1991.

[4] J. Chomicki. Temporal query languages: A survey. *Proc. fo the Intl. Conference on Temporal Logic*, Lecture Notes in AI(827):68–81, 1994.

[5] A. Cimatti, F. Giunchiglia, and R. W. Weyhrauch. A many-sorted natural deduction. *Computational Intelligence*, 14:134–149, 1998.

[6] A. Cohn. A more expressive formulation of many sorted logics. *Journal of Automated Reasoning*, 3:113–200, 1987.

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

[8] T. Dean and D. McDermott. Temporal data base management. *Artificial Intelligence*, 32:1–55, 1987.

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

[10] B. A. Haugh. Non-standard semantics for the method of temporal arguments. In *Proc. IJCAI'87*, pages 449–454. Morgan Kaufmann, 1987.

[11] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 3, 1986.

[12] M. Manzano. *Many-sorted logic and its Applications in Computer Science*, chapter Introduction to Many-sorted logic (pp. 1-86). John Wiley&Sons. Chichester. (UK), tucker & meinke edition, 1993.

[13] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of AI. *Machine Intelligence*, 4, 1969.

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

[15] H. Reichgelt. Semantics for reified temporal logic. Technical Report D.A.I. Research Paper No. 299, Department of Artificial Intelligence University of Edinburgh, 1986.

[16] H. Reichgelt. A comparison of first-order and modal logics of time. In P. Jackson, H. Reichgelt, and F. van Harmelen, editors, *In Logic-based Knowledge Representation*, pages 143–176. The MIT press, 1989.

[17] R. Reiter. *Knowledge in Action.* MIT press, 2001.

[18] M. Shanahan. *The Frame Problem.* MIT press, 2000.

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

[20] R. Snodgrass and I. Ahn. Temporal databases. *IEEE Computer*, 19(9), 1986.

[21] A. Tansel. A generalized relational framework for modelling temporal data. In T. et al., editor, *Temporal Databases*, pages 183–201. Benjamin/Cummings Publishing, 1993.

[22] A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass. *Temporal Databases.* Benjamin/Cummings Publishing, 1993.

[23] L. Vila and H. Reichgelt. The token reification approach to temporal reasoning. *Artificial Intelligence*, 83(1):59–74, May 1996.

[24] C. Walther. *A Many-Sorted Calculus Based on Resolution and Paramolulation.* Research Notes in Artificial Intelligence. Pitman, 1987.