Lecture Notes 3/4 – Temporal Reasoning

Patrick Doherty

February 7, 1998

1 Outline

1. Tasks requiring Temporal Reasoning in AI.
2. Approaches to Temporal Reasoning.
   (a) Classical FOPC
   (b) Method of Temporal Arguments.
   (c) Method of Reification.
3. BTK - Bacchus, Tennenberg, Kooman.
4. STL - Shoham.
5. Situation Calculus.
6. Comparing BTK, STL, SC.
7. Discussion.

2 Tasks Requiring Temporal Reasoning

Just about any type of intelligent behavior or reasoning task has a temporal aspect. Agents, both artificial and human, are embedded in a dynamically changing world. Reasoning takes time and involves a combination of sequential and parallel processes. To solve a problem, agents delegate various amounts of time to potential solutions, increasing the amount of time spent on partial solutions that show progress toward a goal, decreasing it for others. When reasoning about physical systems, interest lies in their dynamic behavior, and we try to predict how they will behave in the future based on observations in the past. Shoham and Goyal [**?**] identify several tasks in AI that require reasoning about time:

1. *Prediction* – Given a description of the world over some period of time, and the set of rules governing change, predict the world at some future time.
2. *Explanation* – Given a description of the world over some period of time and the rules governing change, produce a description of the world at some earlier time that accounts for the world being the way it is at the later time.
3. *Planning* – Given a description of some desired state of the world over some period of time, and the rules governing change, produce a sequence of actions that would result in a world fitting that description.

4. *Learning new rules* – Given a description of the world at different times, produce the rules governing change which account for the observed regularities in the world.

Assuming it is our task to represent knowledge and construct inference relations for reasoning about action and change, it is clear that if we are successful, then our artificial agents will be capable of envoking a number of tasks representative of intelligent behavior.

According to Davis' methodology, our first task is to define the microworld and analyze it in three respects. The first level of analysis is the ontology of our domain.

1. The definition of the microworld: its entities, its relations, and the rules that govern them. This definition is called the ontology of the domain.

From the tasks listed above, there are a number of fundamental entities that could be included in our ontology:

1. A class of temporal entities and relations between them. The tasks mention specific time points and durations.
2. A state or description of the world.
3. Facts about the world true in some states of the world, false in others.
4. Actions whose effects change the state of the world.

As far as "rules that govern them", there are "rules governing change" and "regularities in the world". It appears that we will need some forms of representation for both the static and dynamic aspects of the world; what changes when actions are performed and what remains unchanged.

The next level of the analysis is

1. The analysis of the type of information that the system must record and the types of inferences it must perform. The result of this analysis is generally a formal language, in which facts of the microworld can be expressed and axioms or rules of inference, which characterize inferences that are reasonable in the microworld.

So, we need to determine a suitable formal languange that is epistemologically adequate for our purposes and capable of modeling in a clear way the phenomena we want to study regarding action and change. There are a number of choices regarding formal languages that we will consider shortly, but note that our choice of ontology will influence our choice of formal language. What sort of inferences are reasonable in our microworld? Well, it is difficult to say as yet, with so little detail. What we can observe are a number of generic tasks that can be described in terms of inference. On the one hand we have both prediction and postdiction, predicting states of the world both forward and backward in time given only partial information about a world history. There is already a hint of a nonmonotonic character here. The explanation task may also be characterized in terms of abduction. Whatever the results, we can see already that we have chosen an extremely complex microworld. Before considering specific formal languages and rules of inference for reasoning about action and change, we will consider a few different approaches to reasoning about time, action and change.

3 Approaches to Temporal Reasoning

Our temporal representations must be capable of addressing a number of different issues. We list a few of them:

1. Representation of change.
2. Representation of the occurence of events.

3. Representation of constraints on possible states of the world.
4. Representation of causal laws.

In addition there are a number of issues related to the particular temporal structures we use:

1. Representation of quantitative relations among times and durations.
2. Distinguishing between past, present and future.
3. Reasoning about processes and continuous time.

Shoham and Goyal [?] make a distinction between two broad classes of formalisms for temporal reasoning, that of the *change-based approach* and the *time-based approach*. Within each class, there are a number of choices that can be made which help or hinder in addressing a number of the issues listed above. During the course, we will consider examples of both approaches and a number of the issues listed.

Change-based approaches are characterized by their event-driven nature. The world is in a particular state until an action, event, or execution of a program, for instance, changes the world's state. The notion of time is generally implicit in the formalism and one is unable to reason about time explicitly on the object level. Examples of change-based approaches are the situation calculus [?], Kowalski and Sergot's event calculus [?], and Pratt's dynamic logic [?]. The situation calculus has had a tremendous influence on research in the area, both positive and negative. We will have much more to say about it during the course.

Time-based approaches are characterized by their time-driven nature. Time progresses independent of any explicit event or activity. There is a time structure, facts are true at various points in the structure, and one can reason about time on the object level. Examples of time-based approaches are Allen's interval logic [?], McDermott's point-based logic [?], Shoham's temporal logic [?] and Sandewall's discrete temporal logic [?].

The first task we face is to choose a method for representing facts that are temporally limited.

3.1 Classical First-Order Logic

The first-order predicate calculus (FOPC) was initially used for reasoning about static phenomona. In particular, it dealt with mathematical truths or universal truths which hold regardless of the state of the world. Such truths were always true, are true now and will always be true. Consequently, there was initially no need to represent facts of a contingent nature, such as beliefs that hold in particular contexts, but not in others, or facts which are true for certain periods of time, false for others. The common denominator for such types of reasoning is that they are context dependent.

What makes FOPC so powerful is its context independence. Its conservatism in this regard provides an explanation for both its power and weakness. We can be assured that given a theory where each fact in the theory is true, any fact deduced from the theory preserves truth even when new facts consistent with the theory are added to it. To insure that this is the case, we can only deduce what is already implicit in our theory.

Very little of the reasoning we use in everyday decision making is context independent. We quite often make conjectures or jump to conclusions that are not necessarily truth preserving relative to particular theories. Trying to model such types of reasoning formally, has a tendency to introduce additional complexity into both our syntax and semantics. In general, where contexts are involved, the property of extensionality is violated in our logics. One turns to intensional or higher order logics, although there are a number of ways to get the best of both worlds by using a technique of reification with FOPC.

When we say that a fact is true at a point in time, there are a number of ways to represent this within the classical framework. There is already a hint of a modal or meta-distinction when we say that "a fact is true" at "a point in time". The point in time is the context which temporally limits the truth of the fact. Note that "a formula is true" in a "model" has a similar ring to it. Unfortunately, the latter is not easily represented within FOPC. This would involve importing meta-level concepts into our object language. There are a number of techniques for representing the temporal information that a fact f is true at time t, or in the situation S, or during the duration I. Two of them are used within the framework of FOPC, whereas the third involves an extension to FOPC.

1. The method of temporal arguments.
2. The method of reification.
3. Modalities.

We will have little to say about modalities in this course. This is not because such approaches are unimportant. In fact, they are the predominant approaches for studying and representing temporal information in philosophy, linguistics and conventional computer science. This course is essentially concerned with temporal reasoning techniques and tools used in mainstream AI. Although a number of approaches do use modalities, the majority of them have not. One has instead relied on the reification technique. There is a wide body of excellent work in computer science which use modal approaches in the area of programming language semantics, specification and analysis of hybrid systems, concurrency, etc. This area deserves a course of its own and an essential area of future research is the integration of insights in both disciplines. Recent work, in fact, is moving in this direction.

We will consider the various approaches and techniques in the following way. In the initial discussion, we will briefly describe the method of temporal arguments and method of reification. We will use a simple time structure, that of situations. We will then study three different logics which use the various approaches and techniques. To simplify the discussion, we first introduce the notion of a *situation* or *state*.

Definition 3.1 (Situation) *A* situation*, or* state *is a snapshot of the world at a given point in time. At different points in time, the world can be in different states.*

Situations will be considered as first-class entities. Consequently they can be used as arguments to first-order predicates and terms. Later on, we will use more explicit and complex temporal entities such as timepoint structures and interval structures, but for the time being situations will suffice.

3.2 The Method of Temporal Arguments

The temporal argument approach involves a straightforward use of FOPC. Every ordinary predicate and function is supplemented by one or more special arguments for time. Here we simply represent our fact f as a predicate constant P, introduce temporal variables and constants as terms in our language, and use the temporal terms as arguments for our predicates. So, fact f is true in the situation $S1$ is represented within the classical framework as $P(S1)$. More concretely, the location of block1 in situation $S1$ is the table would be represented as

$$Location(S1, Block1, Table).$$

Another alternative using a functional term instead of a predicate would result in the following representation

$$location(S1, Block1) = Table.$$

In addition, we would of course have to add an axiomatization of the particular temporal structure we are interested in using to the theory.

In this approach, there is really no distinction between the temporal and atemporal aspects of the formalism in either the syntax or the semantics. For example, unless we introduce sorts into our language, such formulas as $P(red(S1))$ and $P(red(house1))$ are legal formulas. Such problems are not insurmountable, but there is an additional syntactic complexity involved in using this approach. For example, when using functions, it is generally necessary to repeat the temporal argument, $P(S1, red(S1, house1))$. The major advantage of this approach is that it is developed completely within the classical framework. Consequently, both classical theorem provers and results may be used for these formalisms.

3.3 The Method of Reification

The reified arguments approach has been used predominantly in AI. One can probably trace its source back to Montague [?], where he considered using predicates instead of modal operators as a technique for studying various aspects of the semantics of natural language, and McCarthy's reference to the article when presenting the situation calculus [?]. Essential to an understanding of this technique is the notion of a *fluent*. A fluent is a function of situations. A fluent whose range is the boolean truth values is called a boolean or truth-valued fluent. A fluent whose range is an arbitrary object domain is called a fluent.

In the method of reification, time-varying relations, such as *Location*, denote truth-valued or boolean fluents,

$$Location(b, l) : \langle situation \mapsto \{true, false\}.$$

Syntactically, we can use the following representation:

$$Holds(location(b, l), s)$$

where $Holds$ is a predicate constant, $location(b, l)$ denotes a boolean fluent, and s denotes a situation. An alternative representation is

$$value(location(b, l), s) = true$$

where $value$ is a function constant taking a fluent and situation as arguments. Both representations have their uses.

In addition, time-varying terms such as the color of a block, *color*, denote fluents of arbitrary object domains,

$$color(b) : \langle situation \mapsto \{Red, Green, Blue\}.$$

Syntactically, we can use the following representation:

$$value(color(b), s) = Blue.$$

Note that what were previously represented as formulas and terms with their usual denotations in the method of temporal arguments, are now represented as terms with a quite different denotation in the method of reification. Such terms denote fluents, functions defined on situations. Consequently, fluents are first-class entities in our ontology and their syntactic equivalents, the refied terms, can be used as arguments to functions that return fluents themselves, and can even be quantified over. Such possibilities provide a great deal of expressibility, lacking in the method of temporal arguments approach. At the same time, using reification techniques can be problematic and make the language more complex.

The ability to quantify over fluents can be quite useful. For example, one can easily express the fact that all changes in values of fluents are caused by actions

$$\forall f, s, s'.s < s' \wedge value(f, s) \neq value(f, s') \rightarrow \exists a.Causes(a, f, value(f, s')).$$

In the next three sections, we will consider three different logics for temporal representation which have been proposed and which include most of the techniques currently used in AI for reasoning about temporally limited information.

1. BTK is an example of the time-based approach and uses the method of temporal arguments. It is the most classical in nature and is formulated completely withing the confines of classical FOPC with sorts.
2. STL is an instance of the time-based approach and uses a technique similar to reification but does not quite go all the way. It is a sort of half-animal with a refied and modal flavor.
3. The situation claculus (SC) is an instance of the change-based approach and uses the method of reification. It was the first temporal representation used in AI (1963) and is still quite popular, although its usage and limitations are now a topic of current debate.

It is hoped that mastery of these three formalisms will provide enough insight into these techniques so that you may be able to penetrate the majority of articles written in this area, where the logical formalisms used are simply variations on the three discussed.

4 BTK – Bacchus, Tenenberg and Kooman

In this section, we give an example of a temporal logic using the temporal arguments method. The logic is two-sorted. For a discussion of many-sorted logics, see Davis [**?**], pp. 40-43.

4.1 Syntax

In the 2-sorted case, the vocabulary $\Lambda = \langle C, F, R, d \rangle$, contains a mapping d defined on $C \cup F \cup R$. For each constant c, $d(c)$ is either t, or u, representing the sort (t temporal, u atemporal); For each $r \in R$, $d(r)$ is a tuple $\langle k_1, \ldots, k_n, k_{n+1}, \ldots, k_m \rangle$, where $d(k_1) =, \ldots, = d(k_n) = t$ and $d(k_{n+1}) =, \ldots, = d(k_m) = u$, where the k_i's specify the type of argument to r; For each $f \in F$, $d(f)$ is a structure $\langle k_1, \ldots, k_n, k_{n+1}, \ldots, k_m \rightarrow k \rangle$, where again $d(k_1) =, \ldots, = d(k_n) = t$ and $d(k_{n+1}) =, \ldots, = d(k_m) = u$, and $d(k) = t$ or u. If $d(k) = t$ then it is called a *temporal function* and if $d(k) = u$ then it is called an *atemporal function*. So for both relations and functions, the first n arguments are temporal and the last m arguments are atemporal. The list of variables is:

$$\xi_1^t, \xi_2^t, \xi_3^t, \ldots, \xi_1^u, \xi_2^u, \xi_3^u, \ldots$$

In each case the superscript shows the sort of variable. We can now define terms as follows:

Definition 4.1 *Let $\Lambda = \langle C, F, R, d \rangle$ be a vocabulary of the 2-sorted logic. Then*

- *For $c \in C$, c is a term of sort $d(c)$.*
- *The variable ξ_i^j is a term of sort j.*
- *If g is a function symbol of Λ with $d(g) = \langle t_1, \ldots, t_n, u_{n+1}, \ldots, u_m \rightarrow k \rangle$, and $\mu_1, \ldots, \mu_n$ and $\mu_{n+1}, \ldots \mu_m$ are terms of Λ of sorts t and u, respectively, then*

$$g(\mu_1, \ldots, \mu_n, \mu_{n+1}, \ldots, \mu_m)$$

is a term of Λ of sort $d(k)$.

Definition 4.2 *An* atomic formula *of* Λ *is an expression of* Λ *of the form:*

$$r(\mu_1, \ldots, \mu_n, \mu_{n+1}, \ldots, \mu_m)$$

where r *is a relation symbol of* Λ, $d(r) = \langle t_1, \ldots, t_n, u_{n+1}, \ldots u_m \rangle$, *and* $\mu_1, \ldots, \mu_n, \mu_{n+1}, \ldots, \mu_m$ *are terms of* Λ *of sorts* t *and* u, *respectively.*

The definition of *formula* is exactly the same as for the 1-sorted case.

4.2 Semantics

An *interpretation* I of the 2-sorted vocabularly Λ consists of:

1. *Domains* D_t and D_u each of which is a non-empty set.
2. For each constant symbol c of Λ, an element $c_I \in D_{d(c)}$.
3. For each function symbol g of Λ with

$$d(g) = \langle t_1, \ldots, t_n, u_{n+1}, \ldots, u_m \rightarrow k \rangle$$

   a mapping g_I of $D_t^1, \times, \ldots D_t^n \times D_u^{n+1}, \times, \ldots D_u^m$ into D_k.
4. For each relation symbol r of Λ with

$$d(r) = \langle t_1, \ldots, t_n, u_{n+1}, \ldots u_m \rangle$$

   a mapping r_I of $D_t^1, \times, \ldots D_t^n \times D_u^{n+1}, \times, \ldots D_u^m$ into $\{1, 0\}$.

A *valuation* v of an interpretation I is the sequence

$$v_j^t, v_j^u; \; j = 1, 2, 3, \ldots$$

where each $v_j^t \in D_t$ and each $v_j^u \in D_u$. We say that v_j^k is the value of the variable ξ_j^k with respect to the valuation v. If v and w are both valuations of I, we write $v \equiv_{k,j} w$ to mean that $v_i^k = w_i^k$ for all $i \neq j$.

Now, given these preliminaries, we can modify the 1-sorted version of $VALUE$ as follows:

2. If v is the variable xi_j^k then $VALUE(v, I, v) = v_j^k$.
6. If $\gamma = (\exists \xi_j^k)\alpha$ then $VALUE(\gamma, I, v) = 1$ if there is a valuation w of I such that $w \equiv_{k,j} v$ and $VALUE(\alpha, I, w) = 1$; otherwise $VALUE(\gamma, I, v) = 0$.
7. If $\gamma = (\forall \xi_j^k)\alpha$ then $VALUE(\gamma, I, v) = 0$ if there is a valuation w of I such that $w \equiv_{k,j} v$ and $VALUE(\alpha, I, w) = 0$; otherwise $VALUE(\gamma, I, v) = 1$.

4.3 Temporal Ontology

Given the 2-sorted logic, BTK, we would like to give the temporal subdomain D_t some structure. For example, the temporal subdomain could be the integers, rationals, or the reals, dependent on whether we wanted to work with discrete, dense, or continuous time. We must be careful though. So far, we have placed no structure on the domains of BTK. Consequently, there is a complete proof theory for BTK. In fact, BTK can be relativized to a standard 1-sorted logic without loss of power.[1] On the other hand, if we choose certain structures for time, it may not be the case that there is a complete axiomatization for the particular structures. For example, choosing the integers with multiplication and addition as the temporal domain will not provide us with a

[1] See Davis [?], pp. 40-43, or Bacchus *et. al.* [?],p.90.

complete axiomatization. There isn't any. Although the integers can not be characterized by a set of first-order axioms, there are a number of other structures that can be, such as linear-ordered structures, Peano arithmetic and totally ordered fields. In fact, we can even choose intervals as our temporal objects and we can provide a complete axiomatization for such a temporal domain.

Given our choice of temporal domain, how do we use it with BTK? We simply add the axioms which characterize our temporal structure to the axiomatization of first-order logic. The first-order rules of inference will, in union with our extended axiom set, provide a complete proof theory. Let's look at an example. Below we list the first-order axioms which characterize the linearly-ordered temporal structures:

$$\forall x . x \leq x$$
$$\forall xy . x \leq y \wedge y \leq x \rightarrow y = x$$
$$\forall xyz . x \leq y \wedge y \leq z \rightarrow x \leq z$$
$$\forall xy . x \leq y \vee y \leq x.$$

In conclusion, we have described a temporal logic based on the method of temporal arguments. It has the advantage that it is constructed completely within the classical framework and therefore we can use standard proof theoretical methods to reason with our temporal scenarios. This is of course under the assumption that monotonic logic suffices for modeling the phenomena we are interested in, which of course, it doesn't. In any event, extending a well understood logic is probably a better idea then constructing a nonstandard logic which is not well understood at all. But, this claim remains to be verified. It just may be the case that one can capitalize on properties of nonstandard formalisms when constructing inference engines.

4.4 Example1

The action scenarios we will consider take place in the context of a blocks world where blocks can be moved and painted.

We will use the following sorts: timepoints (t), individuals (u).

Our vocabulary, $\Lambda = \langle C, F, R, d \rangle$, contains the following:

1. $C = TC \cup C'$ where

$$TC = \{1, 2, \ldots\} \text{ and } \forall x \in TC . d(x) = t,$$
$$C' = \{Block_1, \ldots Block_n, Table, Red, White, Blue\} \text{ and } \forall x \in C' . d(x) = u,$$

2. $F = TF \cup F'$ where

$$TF = \{+\} \text{ and } d(+) = \langle t \times t \rightarrow t \rangle.$$
$$F' = \{top\} \text{ and } d(top) = \langle t \times u \rightarrow u \rangle.$$

3. $R = \{At, Color, Clear, Move, Paint, Location, Block, Trivial, Ab_x, \leq\}$ where

$$d(At) = d(Color) = d(Move) = d(Paint) = \langle t \times u \times u \rangle,$$
$$d(Clear) = d(Trivial) = d(Ab_7) = \langle t \times u \rangle,$$
$$d(Location) = d(Block) = \langle u \rangle,$$
$$d(Ab_n) = \langle t \times u \times u \rangle \text{ for n } = 1 \ldots 4,$$
$$d(\leq) = \langle t \times t \rangle,$$

The list of variables for sorts is:

$$\xi^t_1, \xi^t_2, \xi^t_3, \ldots, \xi^u_1, \xi^u_2, \xi^u_3, \ldots$$

To make the axioms more readable we will use the sort type (possibly subscripted) for variables:

$$t, t_1, t_2, t_3, \ldots, u, u_1, u_2, u_3, \ldots \text{etc.}$$

We will use the temporal structure $\langle \mathcal{N}, 0, S, <, + \rangle$, assuming that the proper axioms are added to any of our scenarios.

4.4.1 The Axioms

The example is from McCarthy [?]. For the moment, disregard the Ab_x predicates. Shoham introduces an extension to his temporal logic involving modalities which replace the use of abnormality predicates. But our purpose with the example is to show how one could represent facts, not how to solve various problems associated with reasoning about action and change.

Initial Situation

All blocks are on the table and are white.

$$\forall u. Block(u) \rightarrow At(1, u, Table). \tag{1}$$

$$\forall u. Block(u) \rightarrow Color(1, u, White). \tag{2}$$

Laws of Motion

Blocks change their location when moved and their color when painted.

$$\begin{aligned}\forall t, u, u'. Move(t, u, u') \wedge \neg Ab_3(t, u, u') \\ \rightarrow At(t+1, u, u').\end{aligned} \tag{3}$$

$$\begin{aligned}\forall t, u, u'. Paint(t, u, u') \wedge \neg Ab_4(t, u, u') \\ \rightarrow Color(t+1, u, u').\end{aligned} \tag{4}$$

Frame Axioms

Blocks change their location and color only if there is a reason to change.

$$\begin{aligned}\forall t, u, u'. At(t, u, u') \wedge \neg Ab_1(t, u, u') \\ \rightarrow At(t+1, u, u').\end{aligned} \tag{5}$$

$$\begin{aligned}\forall t, u, u'. Color(t, u, u') \wedge \neg Ab_2(t, u, u') \\ \rightarrow Color(t+1, u, u').\end{aligned} \tag{6}$$

$$\forall t, u, u'. Move(t, u, u') \rightarrow Ab_1(t, u, u'). \tag{7}$$

$$\forall t, u, u'.Paint(t, u, u') \rightarrow Ab_2(t, u, u'). \tag{8}$$

Qualification Axioms

Known conditions which prevent a block from being moved.

$$\forall t, u, u'.\neg(Block(u) \wedge Clear(t, top(t, u))) \vee \neg Clear(t, u') \\ \vee u' = top(t, u) \rightarrow Ab_3(t, u, u'). \tag{9}$$

State Constraints

Since we do not use types for the blocks worlds, we must identify our objects.

$$Block(Block_1) \wedge \ldots \wedge Block(Block_n). \tag{10}$$

$$Location(Table)). \tag{11}$$

$$\forall u.Block(u) \rightarrow Location(top(u)). \tag{12}$$

$$Color(Red) \wedge Color(White) \wedge Color(Blue). \tag{13}$$

A location is clear if there are no trivial objects on it.

$$\forall t, u.Clear(t, u) \equiv \neg \exists u'.\neg Trivial(t, u') \wedge At(t, u', u). \tag{14}$$

$$\forall t, u.\neg Ab_7(t, u) \rightarrow Trivial(t, u). \tag{15}$$

4.4.2 Querying the System

We can use the axiomatization for various purposes. For example, to answer the question "is Block1 White after it is moved on top of Block2?", we would first add

$$Move(1, Block1, top(Block2))$$

to our theory and then try to prove

$$Color(2, Block1, White).$$

Here is another query involving explanation. "Given that Block1 is Red at time 5, can this be explained by the performance of a paint action?". We would first add

$$Color(5, Block1, Red))$$

to our theory and then try to prove

$$\exists t.1 \leq t < 5 \wedge Paint(t, Block1, Red)).$$

4.4.3 Comments

A number of comments are in order as regards the example. In order to provide similarity between the representations to ease comparison between different temporal formalisms, we have used a particular form of theory called "abnormality theories", proposed by John McCarthy. The use of *Ab* predicates provide *hooks* into the representation that will ease discussion related to the frame and qualification problems. It is not at all clear that this is the best way to represent our scenarios for each temporal formalism. But for the moment this will suffice as a means of presenting the syntactic characteristics of each formalism. In later lectures, we will attack the frame and qualification problems using a number of different representations and techniques.

We will also not be too concerned with the type of temporal structure chosen for the same reasons. We know the capabilities and limitations of the BTK as regards choice of temporal construction from previous discussions.

5 STL – Shoham

Another class of temporal formalisms in AI are based on the notion of reification but the actual constructions can not really be classified as pure reified logics. They have more of an intensional flavor. As an example, we will consider Shoham's temporal logic [**?**]. Shoham made an attempt to formally define the reification techniques used in the logics of McDermott and Allen in a precise manner. He also made a number of claims as to the inadequacy of the temporal argument approach previously considered.

Shoham's starting point is "how does one represent temporal information?". As an example, suppose we would like to represent that the color of HOUSE17 is red during the interval $\langle 1,4\rangle$. We have already considered the technique of using temporal arguments. In this case we would use the following formula:

$$Color(1,4,HOUSE17,Red).$$

Shoham is very much against such representations. He states that

> This option is not acceptable from our standpoint, although there is nothing technically wrong with it. The problem is that if time is represented as an argument (or several arguments) to predicates, then there is nothing general you can say about the temporal aspects of assertions. For example, you cannot say that "effects cannot precede their causes"; at most you can say that about specific causes and effects. Indeed, this first option accords no special status to time – neither conceptual nor notational – which goes against the very spirit of our enterprise.([**?**], p. 90.)

Before considering his claims, we will present his approach to temporal reasoning. The first step is to separate the atemporal component "HOUSE17 is red" from the temporal component $\langle 1,4\rangle$. We will call the atemporal component a *propositional type* in agreement with current AI terminology. The next step is to associate the propositonal type with the interval. Shoham introduces a strange animal $TRUE$ and calls it a global predicate. It is strange, because it is not quite a predicate in the classical sense, nor is it a modal operator, but it's behavior is closer to that of a modality. The propositional type and the interval will appear as arguments to $TRUE$. In our example we would have $TRUE(1,4,COLOR(HOUSE17,RED))$. The immediate questions that arise are

1. What is the logical status of $COLOR(HOUSE17,RED)$? Is it a term or somthing else?
2. What sorts of entities are allowed as propositional types? Do we allow variables or quantifiers? If so, what is a well-formed formula?
3. What is the meaning of such formulas?

There are two directions one can take at this juncture. Propositions can be reified, so that the propositional type $COLOR(HOUSE17, RED)$ has the logical status of a term in the language. This was the approach suggested by McCarthy and used in the situation calculus, a formalism we will also consider in a later section. The other approach, the approach taken by Shoham, is to retain the propositional nature of the propositional types and at the same time the classical nature of the logic as a whole. This is somewhat inconsistent with the approach Shoham actually takes because he introduces a nonstandard semantics for his temporal formalism.

The idea is as follows. Each primitive formula is an interval/assertion pair, $\langle\langle t_1, t_2\rangle, p\rangle$, where t_1 and t_2 are timepoints and p is a proposition $r(trm_1, \ldots, trm_n)$ where r is a relation symbol and the terms used as arguments are nontemporal. Shoham uses $TRUE(t_1, t_2, p)$ as a visual aid for the primitve formula $\langle\langle t_1, t_2\rangle, p\rangle$. Given this informal introduction, we now describe the formal syntax and semantics of Shoham's temporal logic.

5.1 Syntax

In the 2-sorted case, the vocabulary $\Lambda = \langle C, F, R, d\rangle$, contains a mapping d defined on $C \cup F \cup R$. Let $C = TC \cup C'$ where $d(c) = t$ for $c \in TC$ and $d(c) = u$ for $c \in C'$. For each constant c, $d(c)$ is either t or u, representing the sort (t temporal, u atemporal). TC and C' are disjoint.

Let $F = TF \cup F'$ where for $f \in TF$, $d(f) = \langle t_1, \ldots, t_n \rightarrow t\rangle$, and for $f \in F'$, $d(f) = \langle u_1, \ldots, u_n \rightarrow u\rangle$. The functions in TF are called *temporal functions* and those in F' are called *atemporal functions*. TF and F' are disjoint.

Let $R = \{TRUE\} \cup R'$ where for $r \in R'$, $d(r) = \langle u_1, \ldots, u_n\rangle$, and for $r = TRUE$, $d(r) = \langle t_1, t_2, \langle u_1, \ldots, u_n\rangle\rangle$.

The list of variables is:

$$\xi_1^t, \xi_2^t, \xi_3^t, \ldots, \xi_1^u, \xi_2^u, \xi_3^u, \ldots$$

where $TV = \{\xi_1^t, \xi_2^t, \xi_3^t, \ldots\}$ and $V = \{\xi_1^u, \xi_2^u, \xi_3^u, \ldots\}$. In each case the superscript shows the sort of variable. We can now define terms as follows:

Definition 5.1 (Temporal Terms) *Let $\Lambda = \langle C, F, R, d\rangle$ be a vocabulary of the 2-sorted logic. Then*

- *For $c \in TC$, c is a term of sort $d(c) = t$.*
- *The variable ξ_i^t is a term of sort t.*
- *If $g \in TF$ is a function symbol of Λ with $d(g) = \langle t_1, \ldots, t_n \rightarrow t\rangle$, and $\mu_1, \ldots, \mu_n$ are terms of Λ of sort t,then*

  $$g(\mu_1, \ldots, \mu_n)$$

  is a term of Λ of sort $d(g) = t$.

Definition 5.2 (Atemporal Terms) *Let $\Lambda = \langle C, F, R, d\rangle$ be a vocabulary of the 2-sorted logic. Then*

- *For $c \in C'$, c is a term of sort $d(c) = u$.*
- *The variable ξ_i^u is a term of sort u.*
- *If $g \in F'$ is a function symbol of Λ with $d(g) = \langle u_1, \ldots, u_n \rightarrow u\rangle$, and $\mu_1, \ldots, \mu_n$ are terms of Λ of sorts u,then*

  $$g(\mu_1, \ldots, \mu_n)$$

  is a term of Λ of sort $d(g) = u$.

Definition 5.3 (Terms) *The terms of the language are the union of the temporal and atemporal terms.*

Definition 5.4 (Atomic Formula) *An* atomic formula *of* Λ *is an expression of* Λ *of the form:*[2]

$$TRUE(t_1, t_2, r(\mu_1, \ldots, \mu_n))$$

where t_1 *and* t_2 *are temporal terms,* $r \in R'$ *is a relation symbol of* Λ*,* $d(r) = \langle u_1, \ldots, u_n \rangle$*, and* $\mu_1, \ldots, \mu_n$ *are terms of* Λ *of sort* u*.*

Definition 5.5 (Formulas) *The set of* formulas *of* Λ *is similar to the the 1-sorted case with some modification and is defined inductively as follows:*

1. *The atomic formulas of* Λ *are formulas of* Λ*.*
2. *If* t_1 *and* t_2 *are temporal terms, then* $t_1 = t_2$ *and* $t_1 \preceq t_2$ *are formulas of* Λ*.*
3. *Clauses 2-4 in Davis, p. 33.*

5.2 Semantics

We now proceed to define the semantics. Note that it is nonstandard.

An *interpretation* I of the 2-sorted vocabulary Λ is a tuple $I = \langle D_t, \leq, D_u, TFN, FN, RL, M \rangle$ where:

1. The domain D_t is a non-empty set of timepoints.
2. $\leq$ is a binary mapping $D_t \times D_t \mapsto \{0, 1\}$.
3. The domain D_u is a non-empty set of individuals disjoint from D_t.
4. TFN is a set of total functions $\bigcup_i \{D_t^1 \times, \ldots, D_t^i \mapsto D_t\}$.
5. FN is a set of total functions $\bigcup_i \{D_u^1 \times, \ldots, D_u^i \mapsto D_u\}$.
6. RL is a set of total functions $\bigcup_i \{D_u^1 \times, \ldots, D_u^i \mapsto \{1, 0\}\}$.
7. M is a meaning function $\langle M_1, M_2, M_3, M_4, M_5 \rangle$ where
   (a) M_1 is a mapping from $TC \mapsto D_t$.
   (b) M_2 is a mapping from $C' \mapsto D_u$.
   (c) M_3 is a mapping from $TF \mapsto TFN$.
   (d) M_4 is a mapping from $D_t \times D_t \times F' \mapsto FN$. The first two arguments of M_4 are commutative.
   (e) M_5 is a mapping from $D_t \times D_t \times R' \mapsto RL$. The first two arguments of M_5 are commutative.

A *valuation* v of an interpretation I is the sequence

$$v_j^t, v_j^u;\ j = 1, 2, 3, \ldots$$

where each $v_j^t \in D_t$ and each $v_j^u \in D_u$. We say that v_j^k is the value of the variable ξ_j^k with respect to the valuation v. If v and w are both valuations of I, we write $v \equiv_{k,j} w$ to mean that $v_i^k = w_i^k$ for all $i \neq j$.

If v is a temporal term of Λ, I is an interpretation of Λ with domains D_t and D_u, and v is a valuation of I, we define $VALUE(v, I, v) \in D_t$ as follows:

- Temporal Terms:

[2] Recall that $TRUE(t_1, t_2, r(\mu_1, \ldots, \mu_n))$ is syntactic sugar for $\langle\langle t_1, t_2 \rangle r(\mu_1, \ldots, \mu_n)\rangle$. "$TRUE$" is not a predicate or modal operator and "$r(\mu_1, \ldots, \mu_n)$" is not a term.

1. If $v \in TC$ then $VALUE(v, I, v) = M_1(v)$.
2. If $v \in TV$ is the variable ξ^t_j then $VALUE(v, I, v) = v^t_j$.
3. If $v \in TF$ is the function $v = g(\mu_1, \ldots, \mu_n)$, and if

$$VALUE(\mu_i, I, v) = d_i \in D_t, \text{ for } i = 1, 2, \ldots, n.$$

then we set

$$VALUE(v, I, v) = M_3(g)(d_1, d_2, \ldots d_n).$$

If v is an atemporal term of Λ, I is an interpretation of Λ with domains D_t and D_u, and v is a valuation of I, we define $MVA(d_t, d'_t, v) \in D_u$ as follows:

- Atemporal Terms:

1. If $v \in C'$ then $\forall d_t, d'_t \in D_t.MVA(d_t, d'_t, v) = M_2(v)$.
2. If $v \in V'$ is the variable ξ^u_j, then $\forall d_t, d'_t \in D_t.MVA(d_t, d'_t, v) = v^u_j$.
3. If $v \in F'$ is the function $v = g(\mu_1, \ldots, \mu_n)$, then for all $d_t, d'_t \in D_t$, if

$$MVA(d_t, d'_t, \mu_i) = d_i \in D_u, \text{ for } i = 1, 2, \ldots, n$$

then

$$M_4(d_t, d'_t, g)(d_1, d_2, \ldots d_n).$$

If γ is a formula of Λ, and I, d, and v are as above, we define the $VALUE(\gamma, I, v) \in \{0, 1\}$ as follows:

0. If $\gamma = TRUE(t_1, t_2, r(\mu_1, \ldots, \mu_n))$ is an atomic formula and if

$$VALUE(t_1, I, v) = d_t \in D_t \text{ and } VALUE(t_2, I, v) = d'_t \in D_t$$

and

$$MVA(d_t, d'_t, \mu_i) = d_i \in D_u, \text{ for } i = 1, 2, \ldots, n$$

then

$$VALUE(\gamma, I, v) = M_5(d_t, d'_t, r)(d_1, d_2, \ldots d_n).$$

1-5. Same as in Davis, pp36-37.

6. If $\gamma = (\exists \xi^k_j)\alpha$ then $VALUE(\gamma, I, v) = 1$ if there is a valuation w of I such that $w \equiv_{k,j} v$ and $VALUE(\alpha, I, w) = 1$; otherwise $VALUE(\gamma, I, v) = 0$.
7. If $\gamma = (\forall \xi^k_j)\alpha$ then $VALUE(\gamma, I, v) = 0$ if there is a valuation w of I such that $w \equiv_{k,j} v$ and $VALUE(\alpha, I, w) = 0$; otherwise $VALUE(\gamma, I, v) = 1$.
8. If $\gamma = (t_1 = t_2)$ then $VALUE(\gamma, I, v) = 1$ if $VALUE(t_1, I, v) = VALUE(t_2, I, v)$; otherwise $VALUE(\gamma, I, v) = 0$.
9. If $\gamma = (t_1 \preceq t_2)$ then $VALUE(\gamma, I, v) = 1$ if $VALUE(t_1, I, v) \leq VALUE(t_2, I, v)$; otherwise $VALUE(\gamma, I, v) = 0$.

5.3 Temporal Ontology

Shoham's base logic permits reasoning with *temporal propositions*, propositions which associate a *propositional type* with an interval. On the other hand, when reasoning about temporal phenomena, a number of ontological distinctions can, and should be made between the entities we reason about. Just to name a few:

- *Properties* such as an object's color, or speed.
- *Actions* such as moving a block, or shooting a gun.
- *Processes* such as keeping a bucket filled with water, or running around a building.

Rather than fixing an ontology and hard-wiring specific propositional types into the logic as as done by Allen with his properties/events/processes ontology, or McDermott with his facts/events ontology, Shoham's logic permits the construction of ontology via axiomatization. In other words, the axiom writer, or knowledge representer can add new propositional types to the formalism simply be writing axioms for the type. Different kinds of propositional type will be distinguished by specifying how the truth of a proposition over one interval is related to its truth over other intervals.

Shoham provides a number of examples. "The robot travels less than two miles" is a type of activity where if it occurs during an interval, it also occurs during all subintervals of the interval. Such a propositional type has the property of being *downward hereditary.*

Definition 5.6 *A proposition type x is considered* downward-hereditary *if whenever it holds over an interval it holds over all of its subintervals, possibly excluding the two end points.*

This property can be characterized in Shoham's logic using the following axiom:

$$\begin{aligned}\forall t_1, t_2, t_3, t_4.\ t_1 \leq t_3 \wedge t_4 \leq t_2 \wedge t_1 \neq t_4 \wedge t_3 \neq t_2 \wedge \\ TRUE(t_1, t_2, x) \rightarrow TRUE(t_3, t_4, x).\end{aligned} \tag{16}$$

The "robot travelled at a speed of two miles per hour" is a type of activity where if it occurs in all subintervals of an interval, it also occurs in that interval. Such a propositional type has the property of being *upward hereditary.*

Definition 5.7 *A proposition type x is considered* upward-hereditary *if whenever it holds for all proper subintervals of some nonpoint interval (except possibly at its endpoints) it also holds over the nonpoint interval itself.*

This property can be characterized in Shoham's logic using the following axiom:

$$\begin{aligned}\forall t_1, t_2. t_1 < t_2 \wedge (\forall t_3, t_4. t_1 \leq t_3 \leq t_4 \leq t_2 \wedge \\ (t_1 \neq t_3 \vee t_2 \neq t_4) \wedge t_1 \neq t_4 \wedge t_3 \neq t_2 \rightarrow TRUE(t_3, t_4, x)) \\ \rightarrow TRUE(t_1, t_2, x).\end{aligned} \tag{17}$$

This is a fascinating tool which permits a great flexibility in setting up ontologies. Although the idea is associated with Shoham's particular logic, it does not rule out its use in other formalisms. We will observe during the course that there are a number of different propositional types that are useful when modeling temporal phenomena. We should also keep in mind this particular method for formalizing propositional types.

5.4 Example1

The action scenarios we will consider take place in the context of a blocks world where blocks can be moved and painted.

We will use the following sorts: timepoints (t), individuals (u).

Our vocabulary, $\Lambda = \langle C, F, R, d \rangle$, contains the following:

1. $C = TC \cup C'$ where

$$TC = \{1, 2, \ldots\} \text{ and } \forall x \in TC.d(x) = t,$$

$$C' = \{Block_1, \ldots Block_n, Table, Red, White, Blue\} \text{ and } \forall x \in C'.d(x) = u,$$

2. $F = TF \cup F'$ where

$$TF = \{+\} \text{ and } d(+) = \langle t \times t \rightarrow t \rangle.$$

$$F' = \{top, move\} \text{ and } d(top) = \langle u \rightarrow u \rangle.$$

$$d(move) = \langle u \times u \rightarrow u \rangle$$

3. $R = \{At, Color, Clear, Move, Paint, Location, Block, Trivial, Ab_x\}$ where

$$d(At) = d(Color) = d(Move) = d(Paint) = d(Eq) = \langle u \times u \rangle,$$

$$d(Clear) = d(Trivial) = d(Ab_7) = d(Location) = d(Block) = \langle u \rangle,$$

$$d(Ab_n) = \langle u \times u \rangle \text{ for n } = 1 \ldots 4,$$

The list of variables for sorts is:

$$\xi_1^t, \xi_2^t, \xi_3^t, \ldots, \xi_1^u, \xi_2^u, \xi_3^u, \ldots$$

To make the axioms more readable we will use the sort type (possibly subscripted) for variables:

$$t, t_1, t_2, t_3, \ldots, u, u_1, u_2, u_3, \ldots \text{etc.}$$

We will also use the simplifying assumption that the time structure is that of the (positive and negative)integers (discrete, linear, unbounded on both sides). In other words, we will only assume use of models with this particular structure.

5.4.1 The Axioms

The example is from McCarthy [?]. For the moment, disregard the Ab_x predicates. Shoham introduces an extension to his temporal logic involving modalities which replace the use of abnormality predicates. But our purpose with the example is to show how one could represent facts, not how to solve various problems associated with reasoning about action and change.

Initial Situation

All blocks are on the table and are white.

$$\forall u.TRUE(1, 1, Block(u)) \rightarrow TRUE(1, 1, At(u, Table)). \tag{18}$$

$$\forall u.TRUE(1, 1, Block(u)) \rightarrow TRUE(1, 1, Color(u, White)). \tag{19}$$

Laws of Motion

Blocks change their location when moved and their color when painted.

$$\forall t,t',u,u'.t=t' \wedge TRUE(t,t',Move(u,u')) \wedge \neg TRUE(t,t',Ab_3(u,u')) \rightarrow TRUE(t+1,t'+1,At(u,u')). \quad (20)$$

$$\forall t,t',u,u'.t=t' \wedge TRUE(t,t',Paint(u,u')) \wedge \neg TRUE(t,t',Ab_4(u,u')) \rightarrow TRUE(t+1,t'+1,Color(u,u')). \quad (21)$$

Frame Axioms

Blocks change their location and color only if there is a reason to change.

$$\forall t,t',u,u'.t=t' \wedge TRUE(t,t',At(u,u')) \wedge \neg TRUE(t,t',Ab_1(u,u')) \rightarrow TRUE(t+1,t'+1,At(u,u')). \quad (22)$$

$$\forall t,t',u,u'.t=t' \wedge TRUE(t,t',Color(u,u')) \wedge \neg TRUE(t,t',Ab_2(u,u')) \rightarrow TRUE(t+1,t'+1,Color(u,u')). \quad (23)$$

$$\forall t,t',u,u'.t=t' \wedge TRUE(t,t',Move(u,u')) \rightarrow TRUE(t,t',Ab_1(u,u')). \quad (24)$$

$$\forall t,t',u,u'.t=t' \wedge TRUE(t,t',Paint(u,u')) \rightarrow TRUE(t,t',Ab_2(u,u')). \quad (25)$$

Qualification Axioms

Known conditions which prevent a block from being moved.

$$\forall t,t',u,u'.\neg TRUE(t,t',Block(u) \wedge Clear(top(u))) \vee \neg TRUE(t,t',Clear(u')) \vee TRUE(t,t',Eq(u',top(u))) \rightarrow TRUE(t,t',Ab_3(u,move(u,u'))). \quad (26)$$

State Constraints

Since we do not use types, we must identify our objects.

$$\forall t,t'.TRUE(t,t',Block(Block_1) \wedge \ldots \wedge Block(Block_n)). \quad (27)$$

$$\forall t,t',u.TRUE(t,t',Location(Table)). \quad (28)$$

$$\forall t,t',u.TRUE(t,t',Block(u)) \rightarrow TRUE(t,t',Location(top(u))). \quad (29)$$

$$\forall t,t'.TRUE(t,t',Color(Red) \wedge Color(White) \wedge Color(Blue)). \quad (30)$$

A location is clear if there are no trivial objects on it.

$$\forall t,t',u.TRUE(t,t',Clear(u)) \equiv \neg \exists u'.\neg TRUE(t,t',Trivial(u')) \wedge TRUE(t,t',At(u',u)). \quad (31)$$

$$\forall t,t',u.\neg TRUE(t,t',Ab_7(u) \rightarrow \neg TRUE(t,t',Trivial(u)) \quad (32)$$

5.4.2 Querying the System

We can use the axiomatization for various purposes. For example, to answer the question "is Block1 White after it is moved on top of Block2?", we would first add

$$TRUE(1, 1, Move(Block1, top(Block2)))$$

to our theory and then try to prove

$$TRUE(2, 2, Color(Block1, White)).$$

Here is another query involving explanation. "Given that Block1 is Red at time 5, can this be explained by the performance of a paint action?". We would first add

$$TRUE(5, 5, Color(Block1, Red))$$

to our theory and then try to prove

$$\exists t.1 \preceq t \prec 5 \wedge TRUE(t, t, Paint(Block1, Red)).$$

6 The Situation Calculus – McCarthy and Hayes

The *situation calculus* was proposed by McCarthy and Hayes [?] for formalizing reasoning about action and change. The situation calculus can be classified as a member of the change-based approach. There is no explicit notion of time modeled other than the progression of *situations* defined as "complete states of the universe at instances in time". The world persists in one state or situation until an action is performed which changes the state to a new state. Temporal reasoning is done by reasoning about facts true in situations as a result of actions performed in sequence. In the situation calculus, only one event occurs at a time, time durations are irrelevent, and the state of the world during the performance of an action is irrelevent. Actions are considered instantanteous.

The formalism has dominated research in the area and continues to do so, although there are now a number of alternatives which in the long run will probably make the situation calculus obsolete. This is a point of current debate in the research community. The situation calculus has a number of nice features, the most important being that it is defined within the classical framework as a sorted first-order predicate calculus. The sorts used vary relative to the object domain being modeled and to the particualr methods used for dealing with a number of generic problems such as the frame and qualification problems.

We have already considered an example of a 2-sorted logic where our sorts were temporal objects and individuals. It is straightforward to imagine how an n-sorted logic could be defined. For a general definition, see Davis [?], pp. 40-43. The situation calculus is a reified logic. There will always be sorts for facts, actions and situations. Facts have previously been called propositional types and in this framework are normally called *fluents*. Recall that a reified logic is generally defined as a "logic where propositions are made into objects (reified) by providing special propositional terms to refer to them."[?]. In addition to propositional terms (fluents), the situation calculus also reifies actions such as "move" and situations. Because of this, we are able to represent general axioms about facts, actions and situations simply by quantifying over their sorts. Before proceeding with the description of the situation calculus, we will first describe the essentials in an informal manner:

- A *Situation* is the complete state of the universe at an instant in time. Object variables for situations will be written as $s, s_0, s_1, \ldots$. We will also use constants written as $S0, S1, \ldots$.
- A *fluent* is a function defined on situations. Object variables for fluents will be written as $f, f_0, f_1, \ldots$.

  For example, the location of a block x at situation s is denoted $location(x, s)$.

- An *action* is a function or constant.Object variables for actions will be written as $a, a_0, a_1, \ldots$.

  For example, the action of moving block x to location l is denoted by $move(x, l)$

- If s is a situation and a is an action then the situation resulting from performing action a in s is represented by the function $result$, where $result(a, s)$ refers to the resulting situation.

  For example $location(A, result(move(A, Table), S0)) = Table$ states that the location of A after performing the move to table operation in situation $S0$ is $Table$ and $result(move(A, Table), S0)$ refers to the situation in which this is the case.

- In order to assert that fluents have particular values in particular states, we use the predicate $Holds$, where if f is a fluent and s is a situation then $Holds(f, s)$ denotes that the fluent f is true in situation s.

  For example, $Holds(dead(fred), result(shoot(fred), S0))$ states that fred is dead in the situation resulting from performing the action of shooting fred in situation $S0$.

Any vocabulary, $\Lambda = \langle C, F, R, d\rangle$, will generally contain the following:

1. There are at least three sorts, situations (s), actions (a), and fluents (f).
2. For each $x \in \{S0, S1, \ldots, Sn\}$, $d(x) = s$ and $x \in C$.
3. $d(result) = \langle a \times s \rightarrow s\rangle$ and $result \in F$.
4. $d(Holds) = \langle f \times s\rangle$ and $Holds \in R$.

The domain D will contain at least three non-empty subdomains for each of the sorts: $D = D_s \cup D_a \cup D_f \cup \ldots$ where each subdomain is non-empty and disjoint from the others.

Rather than starting from scratch, we will assume an n-sorted logic and describe the representation of two action scenarios.

6.1 Example1

The action scenarios we will consider take place in the context of a blocks world where blocks can be moved and painted.

We will use the following sorts: actions (a), situations (s), truth-valued fluents (f), blocks (b), locations (l), colors (c).

Our vocabulary, $\Lambda = \langle C, F, R, d\rangle$, contains the following:

1. $C = C_s \cup C_b \cup C_l \cup C_c$ where

$$C_s = \{S0\} \text{ and } \forall x \in C_s.d(x) = s,$$

$$C_b = \{Block_1, \ldots Block_n\} \text{ and } \forall x \in C_b.d(x) = b,$$

$$C_l = \{Table\} \text{ and } \forall x \in C_l.d(x) = l,$$

$$C_c = \{Red, White, Blue\} \text{ and } \forall x \in C_c.d(x) = c.$$

2. $F = \{top, at, color, clear, move, paint, result, trivial\}$ where

$$d(top) = \langle b \rightarrow l\rangle,$$

$$d(at) = \langle b \times l \rightarrow f\rangle,$$

$$d(color) = \langle b \times c \rightarrow f\rangle,$$

$$d(clear) = \langle l \rightarrow f \rangle,$$
$$d(move) = \langle b \times l \rightarrow a \rangle,$$
$$d(paint) = \langle b \times c \rightarrow a \rangle,$$
$$d(eql) = \langle l \times (b \rightarrow l) \rightarrow f \rangle,$$
$$d(result) = \langle a \times s \rightarrow s \rangle.$$
$$d(trivial) = \langle b \rightarrow f \rangle,$$

3. $R = \{Holds\}$ where

$$d(Holds) = \langle f \times s \rangle.$$

$$d(Ab_n) = \langle b \times a \times s \rangle \text{ for n} = 1 \ldots 4,$$
$$d(Ab_7) = \langle b \times s \rangle.$$

The list of variables for sorts is:

$$\xi_1^a, \xi_2^a, \xi_3^a, \ldots, \xi_1^s, \xi_2^s, \xi_3^s, \ldots \text{etc.}$$

To make the axioms more readable we will use the sort type (possibly subscripted) for variables:

$$a, a_1, a_2, a_3, \ldots, s, s_1, s_2, s_3, \ldots \text{etc.}$$

6.1.1 The Axioms

The following axioms are slightly modified and taken from an example used by McCarthy [?]. For the moment, disregard the Ab_x predicates. We will discuss their purpose after introducing abnormality theories in a future lecture.

Initial Situation

$$\forall b.Holds(at(b, Table), S0) \tag{33}$$

$$\forall b.Holds(color(b, White), S0) \tag{34}$$

Laws of Motion

Blocks change their location when moved and their color when painted.

$$\forall b, a, s.\neg Ab_3(b, a, s) \rightarrow \\ Holds(at(b, l), result(move(b, l), s)) \tag{35}$$

$$\forall b, a, s.\neg Ab_4(b, a, s) \rightarrow \\ Holds(color(b, c), result(paint(b, c), s)) \tag{36}$$

Frame Axioms

Blocks change their location and color only if there is a reason to change.

$$\forall b, a, s, l. Holds(at(b,l),s) \wedge \neg Ab_1(b,a,s) \rightarrow \\ Holds(at(b,l), result(a,s)) \tag{37}$$

$$\forall b, a, s, c. Holds(color(b,c),s) \wedge \neg Ab_2(b,a,s) \rightarrow \\ Holds(color(b,c), result(a,s)) \tag{38}$$

$$\forall b, l, s. Ab_1(b, move(b,l), s) \tag{39}$$

$$\forall b, c, s. Ab_2(b, paint(b,c), s) \tag{40}$$

<u>Qualification Axioms</u>

Known conditions which prevent a block from being moved.[3]

$$\forall b, l, s. \neg Holds(clear(top(b)),s) \vee \neg Holds(clear(l),s) \\ \vee Holds(eql(l, top(b)),s) \rightarrow Ab_3(b, move(b,l), s) \tag{41}$$

<u>State Constraints</u>

A location is clear if there are no trivial objects on it.[4]

$$\forall l, s. Holds(clear(l),s) \equiv \\ \neg \exists b. \neg Holds(trivial(b),s) \wedge Holds(at(b,l),s) \tag{42}$$

$$\forall b, s. \neg Ab_7(b,s) \rightarrow \neg Holds(trivial(b),s) \tag{43}$$

6.1.2 Querying the System

We can use the axiomatization for various purposes. For example, to answer the question "is Block1 White after it is moved on top of Block2?", we would try to prove

$$Holds(color(Block1, White), result(move(Block1, top(Block2), S0))).$$

Suppose we would like a plan for painting Block1 Red, placing Block2 on Block3, and Block1 on Block2. We would try to prove the following constructively:

$$\exists s. Holds(color(Block1, Red), s) \wedge Holds(at(Block1, top(Block2), s) \\ \wedge Holds(at(Block2, top(Block3), s) \tag{44}$$

A reasonable answer might be:

$$result(move(Block1, top(Block2)), s2)$$

where

$$s1 = result(paint(Block1, Red), S0),$$
$$s2 = result(move(Block2, top(Block3)), s1).$$

[3]Note that *eql* is a very specific type of equality fluent, applying only to locations. It should be made more general.

[4]Trivial objects would be such things as specks of dust, etc. Unfortunately, we only have the sort blocks for the moment, all of which are non-trivial objects.

7 Comparison between BTK, STL, SitCalc

1. Syntactic Overhead.
2. Semantic Overhead.
3. Status of temporal structure, flexibility.
4. Status of actions, fluents, flexibility.
5. Possibility of Implementation.
6. Expressivity.