Combining Narratives

John McCarthy
Computer Science Department
Stanford University
Stanford, CA 94305

Tom Costello
Computer Science Department
Stanford University
Stanford, CA 94305

Abstract

A theory is *elaboration tolerant* to the extent that new information can be incorporated with only simple changes. The simplest change is conjoining new information, and only conjunctive changes are considered in this paper. In general adding information to a theory should often change, rather than just enlarge, its consequences, and this requires that some of the reasoning be non-monotonic.

Our theories are narratives—accounts of sets of events, not necessarily given as sequences. A narrative is elaboration tolerant to the extent that new events, or more detail about existing events, can be added by just adding more sentences.

We propose a new version of the situation calculus which allows information to be added easily. In particular, events concurrent with already described events can be introduced without modifying the existing descriptions, and more detail of events can be added. A major benefit is that if two narratives do not interact, then they can be consistently conjoined.

1 OBJECTIVES OF SITUATION CALCULUS

The logical approach to AI ([McC59] and [McC89]) is to make a computer program that represents what it knows about the world in general, what it knows about the situation it is in, and also its goals, all as sentences in some mathematical logical language. The program then infers logically what action is appropriate for achieving its goal and does it. Since 1980 it has been widely known that non-monotonic inference must be included. The actions our program can perform include some that generate sentences by other means than logical inference, e.g. by observation of the world or by the use of special purpose non-logical problem solvers.

Simpler behaviors, e.g. actions controlled by servomechanisms or reflexes can be integrated with logic. The actions decided on by logic can include adjusting the parameters of ongoing reflexive actions. Thus a person can decide to walk faster when he reasons that otherwise he will be late, but this does not require that reason control each step of the walking.[1]

Situation calculus is an aspect of the logic approach to AI. A situation is a snapshot of the world at some instant. Situations are *rich*[2] objects in that it is not possible to completely describe a situation, only to say some things about it. From facts about situations and general laws about the effects of actions and other events, it is possible to infer something about future situations. Situation calculus was first discussed in [McC63], but [MH69] was the first widely read paper about it.

In this formalization of action in situation calculus, there are at least three kinds of problem—*narrative*, *planning* and *prediction*. Of these, narrative seems to be the simplest for humans. A narrative is an account of what happened. We treat it by giving some situations and some events and some facts about them and their relations. Situations in a narrative are partially ordered in time. The real situations are totally ordered[3], but the narrative may not include full information about this ordering. Thus the temporal relations between situations need only be described to the extent needed to describe their interactions. Situations occurring entirely in different places give the most obvious examples, but even actions by the same person in the same place may not interact as far as the inferences we draw. If

[1] Thus we protect our flank from the disciples of Rod Brooks.

[2] Though rich, situations are still *approximate, partial objects*. The idea will be developed elsewhere.

[3] Hypothetical situations need not be totally ordered; the situation where Oswald missed Kennedy is neither in the past nor the future.

we state that the traveler on certain flight reads a book and also drinks a Coca-cola, we humans don't need to know any temporal relations between the two events unless they interact.

In situation calculus as it was originally envisaged (and has been used,) events (mainly actions) in a situation produce next situations, e.g. $s' = Result(e, s)$. The original theory did not envisage more than one event occurring in a situation, and it did not envisage intermediate situations in which events occur. However, rarely did people write axioms that forbade[4] these possibilities; it's just that no-one took advantage of them.

Our present formalism doesn't really change the basic formalism of the situation calculus much; it just takes advantage of the fact that the original formalism allows treating concurrent events even though concurrent events were not originally supposed to be treatable in that formalism. Gelfond, Lifschitz and Rabinov [GLR91] treat concurrent events in a different way from what we propose here.

In a narrative, it is not necessary that what is said to hold in a situation be a logical consequence (even non-monotonically) of what was said to hold about a previous situation and known common sense facts about the effects of events. In the first place, in stories new facts about situations are often added, e.g. "When Benjamin Franklin arrived in London it was raining". In the second place, we can have an event like tossing a coin in which neither outcome has even a non-monotonic preference.[5]

In interpreting the following formalizations, we regard situations as rich objects and events as poor. In fact, we are inclined to take a deterministic view within any single narrative. In principle, every event that occurs in a situation and every fact about following situations is an inevitable consequence of the facts about the situation. Thus it is a fact about a situation that a coin is tossed and that it comes up tails. However, such facts are only occasionally consequences of the facts about the situation that we are aware of in the narrative.

Perhaps narrative seems easy, since it is not yet clear what facts must be included in a narrative and what assertions should be inferable from a narrative. We have however a basic model that handles some of the more basic features.

[4] Reiter [Rei93] did write such axioms.

[5] Nevertheless, some narratives are anomalous. If we record that Junior flew to Moscow, and, in the next situation mentioned, assert that he is in Peking, a reader will feel that something has been left out.

We want to introduce a concept of a *proper narrative*, this is a narrative without anomalies. The fluents holding in a new situation should be reasonable outcomes of the events that have been reported, except for those fluents which are newly asserted, e.g. that it was raining in London when Franklin arrived.

2 ELABORATION TOLERANT REASONING

A formalization of a phenomenon is *elaboration tolerant* to the extent that it permits elaborations of the description without requiring completely redoing the basis of the formalization. In particular, it would be unfortunate to have to change the predicate symbols. Ideally the elaboration is achieved by adding sentences, rather than by changing sentences. Often when we add sentences we need to use some form of non-monotonic reasoning. This is because we often want to add information that we would previously have assumed was false. Unless we use non-monotonicity we would get inconsistency. In this paper we concentrate on the easier case when there is no need for non-monotonicity.

Natural language descriptions of phenomena seem to be more elaboration tolerant than any existing formalizations. Here are the two major kinds of elaboration tolerance that we examine in this paper.

2.1 NON-INTERACTING EVENTS

Allowing the addition of a description of a second phenomenon that doesn't interact with the first. In this case the conclusions that can be drawn about the combined narrative are just the conjunction of the conclusions about the component narratives. To infer the obvious consequences of events we need to assume that some other events do not occur. In this paper, a major novelty is that we do not assume that no other events occur. We only state that there are no events that would cause an event in our narrative to fail. Thus a narrative about stacking blocks will state that the only block moving actions[6] are those mentioned. A block stacking narrative will not say that no traveling events occur. Nor will a narrative about traveling makes claims about what block stacking events happen. This allows non-interacting narratives to be consistently conjoined.

Previous proposals could not conjoin two narratives, as they either assumed that the events that happened were picked out by the result function, or they assumed that the only events that occurred were those mentioned.

2.1.1 DETAIL OF EVENTS

We can add details of an event. On the airplane from Glasgow to London, Junior read a book and drank a Coca-Cola. If we make the assumption that other relevant events do

[6] More precisely, no other actions that would move the blocks mentioned in the narrative occur. Other blocks might be stacked in Baghdad, if our narrative is about New York. Perhaps a theory of context, that would interpret a statement about all blocks in our narrative, as a statement about all the blocks *in New York* could be used here.

not happen, we can elaborate by adding another event, so long as it is compatible with what we have said. However the notion of relevant must be formalized very carefully, as is apparent when we elaborate a particular event as a sequence of smaller events. "How did he buy the Kleenex? He took it off the shelf, put it on the counter, paid the clerk and took it home." A narrative that just mentions buying the Kleenex should not exclude this particular elaboration. Moreover, if we elaborate in this way, we don't want to exclude subsequent elaboration of component events, e.g. elaborating paying the clerk into offering a bill, taking the change, etc. Our formalism allows details of an event to be added by conjoining extra sentences.

3 MODIFYING THE SITUATION CALCULUS

Formalisms such as the situation calculus of McCarthy and Hayes [MH69], and the event calculus of Kowalski [KS97] have been used to represent and reason about a changing world c.f [Sha97]. Neither of these formalisms is exactly what is needed to represent the kind of narratives we wish to consider.

The situation calculus in its most limited version does not allow us to represent what events occur explicitly—rather every sequence of events is assumed to occur. We can specify that a particular sequence of events occurs by introducing a predicate, $actual$ true of just the sequence of situations that occur[7]. This is not ideal, as it forces us to decide what events happened earlier, before we name the events that happen later.

For this reason we use a modified situation calculus, adding a new predicate $Occurs(e, s)$, that states what events occur. Thus, rather than the function $Result(e, s)$ serving two purposes, stating that e occurred at s, and designating the resulting situation, we split these two functions. We keep $Result(e, s)$, but it now only denotes the result of doing e in s when e $Occurs$ at s. If e does not occur, then the value of this function is an arbitrary situation[8]. This adds an event calculus style of presentation to the underlying situation calculus formalism. In particular, it allows us to specify a sequence of events, without making any claims as to what other events may have happened in the meantime.

3.1 OUR ONTOLOGY OF SITUATIONS

Reiter has suggested that the situations in the situation calculus be defined axiomatically. He suggests the following four axioms[9]

$$\begin{array}{l}
\mathbf{S0} : \forall s.\neg s < S_0 \\
\forall a, s, s'.s < Result(a, s') \equiv (s = s' \vee s < s') \\
\mathbf{P} : \forall a, a', s, s'.Result(a, s) = Result(a', s') \rightarrow \\
\quad a = a' \wedge s = s' \\
\mathbf{Ind} : \forall \phi.(\phi(S_0) \wedge (\forall a.\phi(s) \rightarrow \phi(Result(a, s)))) \rightarrow \\
\quad \forall s.\phi(s).
\end{array}$$

which determine equality of situations, relative to equality of events or actions. These axioms are categorical, that is relative to an interpretation of equality of actions, there is a unique model of situations.

Rather than use these axioms, which state that no other situations exist between s and $Result(a, s)$, we choose to say that situations can be ordered by a $<$ predicate, which is a strict partial order, which we axiomatize as follows.

$$\begin{array}{l}
\forall a, s.s < Result(a, s), \\
\forall s, s', s''.s < s' \rightarrow \neg(s' < s), \\
\forall s, s, s', s''.s < s' \wedge s' < s'' \rightarrow s < s''
\end{array} \qquad (1)$$

The predicate $<$ is similar to the $future(s, s')$ predicate, introduced by [MH69], which is true when s' is in the future of s. We find it useful to write this in infix notation, and to use $s \leq s'$ as the non-strict version. It also is useful to write $s \leq s' \leq s''$ for $s \leq s' \wedge s' \leq s''$.

4 SPECIFYING THE EFFECTS OF EVENTS

In the situation calculus it is usual to specify the effects of actions by writing *effect* axioms, like[10],

$$\forall s.Holds(Loaded, s) \rightarrow Holds(Dead, Result(Shoot, s)).$$

If we move to a formalism that allows other events to occur between s and $Result(e, s)$, then this way of specifying change needs to be adjusted. It is possible that something might occur in the time between s and $Result(Shoot, s)$ that causes the event to have a different result. For this reason it seems natural to allow the preconditions, those things that hold on the left hand side, to mention properties of all times between s and $Result(Shoot, s)$.

In previous versions of the situation calculus the preconditions for an event were always modeled as a set of fluents, namely those fluents that had to hold at s, for the event to have an effect at $Result(a, s)$. If we allow other things to

[7] Pinto and Reiter [PR95] actually do this.

[8] We could choose instead to make $Result$ a partial function, but this introduces the difficulties of partial functions.

[9] Reiter's notation differ from ours, he uses $do(a, s)$, while we use $Result(a, s)$. We use $\leq s'$ as a shorthand for $s < s' \vee s = s'$. Reiter writes $<$ as $\sqsubset$.

[10] As is customary in Logical A.I. we write $Holds(Dead, s)$ without saying who is dead. We can suppose the events occur in a context and lifting rules exist to make this $Dead(Victim)$ in an outer context. The outer context may contain further preconditions, like that shooter is present.

happen during an event, we cannot just specify the preconditions that must hold at the beginning of the event.

Consider a plane journey from Glasgow to London. It is necessary that the plane be in working order for the entire flight. It is also necessary to be in Glasgow at the beginning of the flight, but clearly, there is no need for this precondition to persist for the entire flight. It is necessary to have a ticket, until the airline steward takes it from you. This is an example of a precondition, "having a ticket" that must hold neither just at the moment the event starts, nor for the entire duration.

Consider another example from the Yale Shooting Problem. In order to successfully shoot a person, the gun must be loaded when the trigger is pulled, but the target must remain in the cross-hairs until the bullet hits. We represent the fact the target is in the cross-hairs by $aimed$. Thus we write:

$$\begin{array}{l}\forall s. Occurs(Shoot, s) \wedge Holds(Loaded, s) \wedge \\ \left(\begin{array}{l}\forall s''. s \leq s'' \wedge s'' < Result(Shoot, s) \rightarrow \\ \quad Holds(Aimed, s'')\end{array}\right) \rightarrow \\ Holds(Dead, Result(Shoot, s)).\end{array}$$

A possible objection to this example is that if the target arrives in the cross-hairs at any time before impact and remains in the path of the bullet, then they will be killed. In this case we write,

$$\begin{array}{l}\forall s. Occurs(Shoot, s) \wedge Holds(Loaded, s) \wedge \\ \left(\begin{array}{l}\exists s_1. s_1 < Result(Shoot, s) \wedge \forall s''. s_1 \leq s'' \wedge \\ \quad s'' < Result(Shoot, s) \rightarrow Holds(Aimed, s'')\end{array}\right) \rightarrow \\ Holds(Dead, Result(Shoot, s)).\end{array}$$

However, both these examples show the need for preconditions to be richer than a statement of what properties hold just when the event occurs. What possible properties can occur as preconditions is important because we wish to know what kinds of axioms can occur as effect axioms.

Before we consider how to represent preconditions, we recall how we represented all possible preconditions earlier. If we wish to introduce a predicate that can parameterize all effect axioms in the old-fashioned situation calculus can write[11],

$$\begin{array}{l}\forall a, f, g. Changes(a, f, g) \stackrel{def}{=} \forall s. Holds(g, s) \rightarrow \\ \neg(Holds(f, s) \equiv Holds(f, Result(a, s))\end{array}$$

following [Cos97]. g is a predicate on fluents that encodes the preconditions. Parameterizing all effect axioms allows us to minimize effect axioms easily.

However, as we want to have preconditions that can extend over the duration of the event, we need more than one set of fluents. For this reason we allow as preconditions, two sets of fluents. The first set need only hold at the start of an event, while the others must persist for the entire event.

[11] We will slightly abuse notation and write $Holds(g, s)$ for $(\forall f'. g(f') \rightarrow Holds(f', s))$, when g is a predicate on fluents.

It might seem that this does not model preconditions that need hold for only part of the duration of the event. However, we can model these by using other defined fluents. Thus we can write that "It is necessary to have a ticket, until the airline steward takes it from you." using a new fluent F_1 defined by

$$\begin{array}{l}\forall s. Holds(F_1, s) \equiv Holds(Has(Ticket), s) \vee \\ \quad Holds(Takenby(Steward, Ticket), s).\end{array}$$

This fluent should hold for the entire[12] duration of the flight. $HoldsD(f, s, e)$ is a shorthand for $\forall s'. s \leq s' < Result(e, s) \rightarrow Holds(f, s')$.

We can write "It is necessary that the plane be in working order for the entire flight" using the fluent $WorkingOrder$,

$$\begin{array}{l}\forall s, l, l', t. HoldsD(WorkingOrder, s, Fly(l, l', t)) \wedge \\ HoldsD(F_1, s, Fly(l, l', t))) \rightarrow \\ \quad Holds(at(l'), Result(Fly(l, l', t), s))\end{array}$$

We can add " It is necessary to be in Glasgow[13] at the beginning of the flight"

$$\begin{array}{l}\forall s, l, l', t. Holds(at(l), s) \wedge \\ HoldsD(WorkingOrder, s, Fly(l, l', t)) \wedge \\ \quad HoldsD(F_1, s, Fly(l, l', t))) \rightarrow \\ \qquad Holds(at(l'), Result(Fly(l, l', t), s))\end{array}$$

Now that our preconditions are represented as two sets[14], rather than one, we redefine $Changes$ as follows:

$$\begin{array}{l}\forall e, f, g_1, g_2. Changes(e, f, g_1, g_2) \stackrel{def}{=} \\ \forall s. (\forall f_1. g_2(f_1) \rightarrow Holds(f_1, s)) \wedge \\ \quad (\forall f_1. g_1(f_1) \rightarrow HoldsD(f_1, s, e)) \rightarrow \\ \neg(Holds(f, s) \equiv Holds(f, Result(e, s)))\end{array}$$

We find it useful to introduce a predicate $Succeeds(e, f, s)$ defined as,

$$\begin{array}{l}\exists g_1, g_2. Changes(e, f, g_1, g_2) \wedge \\ Holds(g_2, s) \wedge HoldsD(g_1, s, e).\end{array} \tag{2}$$

Frame Axioms

We usually would write a frame axiom for a fluent, say $On(A, Top(B))$, block A is on the top of block B, and an action, in this case $Shoot$ as,

$$\begin{array}{l}\forall s. Holds(On(A, Top(B)), s) \equiv \\ Holds(On(A, Top(B)), Result(Shoot, s)).\end{array}$$

[12] The entire duration is taken to be up to, but not including the endpoint. It is sometimes natural that the endpoint should not be needed as a precondition.

[13] We write the general formula with a variable l for Glasgow.

[14] Here we assume that we have a fluent function $\neg$, such that $\forall s, f. Holds(f, s) \equiv \neg Holds(\neg f, s)$. In the absence of the fluent function $\neg$, we would need four sets, two for positive fluents and two for negative fluents.

However, since $Result$ no longer encodes what events occurred, we need to say something like, "if no event that could change the fluent $On(A, Top(B))$ occurred in the interval between s and s' and $On(A, Top(B))$ held at s then $On(A, Top(B))$ will hold at s'." It is notable that this needs the notion of $Changes$ that we introduced above. Thus we write,

$$\begin{array}{l} \mathbf{Frame\ Axiom}: \\ \forall s, s', f. s \leq s' \wedge \\ \left(\begin{array}{l} \forall s'', a. s < Result(a, s'') \leq s' \rightarrow \\ \quad \neg(Occurs(a, s'') \wedge \\ \quad Succeeds(a, f, s'')) \end{array} \right) \rightarrow \\ \quad (Holds(f, s) \equiv Holds(f, s')) \end{array} \tag{3}$$

This will generate all of our frame axioms if we minimize $Changes$, varying $Succeeds$ and $Holds$, and allowing the domain to vary as in [Cos98b, Cos98a]. We do not consider using non-monotonic reasoning to minimize $Changes$ here, as we wish to stress other issues. Thus we explicitly axiomatize the result of the minimization, much in the same way as Reiter[Rei91] uses an explanation closure axiom and an explicit statement of what events can change what fluents.

4.1 WHAT EVENTS OCCUR?

Before we consider combining narratives, we address a problem that arises in our new formalism that was not present in the earlier versions of the situation calculus.

Even frame axioms like this are not enough to allow us to carry out the simple reasoning we could carry out in previous versions of the situation calculus. We also need to know that certain events do not occur.

Consider the following example of moving a block. We have the action that moves a block, but to move a block successfully it must be clear. For instance if someone else puts another block on top of the block we are moving to, then our action will fail.

Thus our only effect axiom is the following one, which states moving block a onto block c succeeds, if a and c are clear for the entire duration, and a is not equal to c.

$$\begin{array}{l} \forall s, a, c, e, l. \\ \left(\begin{array}{l} a \neq c \wedge e = Move(a, Top(c)) \wedge \\ l \neq top(c) \wedge HoldsD(Clear(Top(a)), s, e) \wedge \\ \quad Holds(On(a, l), s) \wedge \\ \quad HoldsD(Clear(Top(c)), s, e) \end{array} \right) \rightarrow \\ \left(\begin{array}{l} Holds(On(a, Top(c)), Result(e, s)) \wedge \\ (l \neq Table \rightarrow \neg Holds(On(a, l), Result(e, s))) \end{array} \right) \end{array}$$

For this example we need some other facts about the world, these are given in an Appendix. We are also told that the only block that is not on the table is A, which is on B, and that the action of moving A to the top of C occurred at the situation $S0$.

$$\begin{array}{l} \forall a, l. Holds(On(a, l), S0) \equiv \\ (a = A \wedge l = Top(B)) \vee (a \neq A \wedge l = Table), \\ Occurs(Move(A, Top(C)), S0). \end{array}$$

We can now write our frame axiom, which in this case is,

$$\begin{array}{l} \forall s, s', a, c. s \leq s' \wedge \\ \left(\begin{array}{l} \forall s'', a', e. e = Move(a, Top(a')) \wedge \\ s < Result(e, s'') \leq s' \wedge a \neq c \rightarrow \\ \quad \neg \left(\begin{array}{l} Occurs(e, s'') \wedge \\ Succeeds(e, On(a, Top(a')), s'') \end{array} \right) \end{array} \right) \\ \quad \rightarrow \left(\begin{array}{l} Holds(On(a, Top(c)), s) \equiv \\ \quad Holds(On(a, Top(c)), s') \end{array} \right) \end{array}$$

This states that a block a is on a block c in a situation s if and only if a is on c in s', so long as there is no event e, of moving moving a to $Top(a')$, which occurs, and is successful.

Some writers like to think that if an event that might change a fluent f occurs, but fails, then the fluent f should be undetermined. We can weaken our frame axiom, so that even if the event of moving a block a' to a fails, then our block a might not be clear. We write this as,

$$\begin{array}{l} \forall s, s', a, c, a', c', b'. (s \leq s' \wedge \\ \left(\begin{array}{l} \forall s'', a'. s \leq Result(a, s'') \leq s' \rightarrow \\ \quad \neg \left(\begin{array}{l} Occurs(Move(a, a'), s'') \wedge \exists g_1, g_2. \\ \quad Changes \left(\begin{array}{l} Move(a, a'), \\ On(a, Top(c)), g_1, g_2 \end{array} \right) \end{array} \right) \end{array} \right) \\ \rightarrow \left(\begin{array}{l} Holds(On(a, Top(c)), s) \equiv \\ \quad Holds(On(a, Top(c)), s'). \end{array} \right) \end{array}$$

In general we shall prefer the stronger frame axiom.

We wish to prove that

$$Holds(On(A, Top(C)), Result(Move(A, Top(C)), S0)),$$

however, we can only prove the weaker,

$$\begin{array}{l} \forall e. e = Move(A, Top(C)) \wedge \\ HoldsD(Clear(A), s, e) \wedge HoldsD(Clear(C), s, e) \rightarrow \\ \quad Holds(On(A, Top(C)), Result(e, S0)) \end{array}$$

Thus, we need to prove that A and C remain clear during the move action in order to show that the action is successful. To show that they remain clear we need to use our frame axiom. But, all we can prove is that the A block remains clear if there is no successful move action $Move(b_1, Top(A))$ that occurs, and whose result is before $Result(Move(A, C), S0)$. As we allow situations before $S0$, we can imagine that there was a move in progress that placed a block on C just after $S0$.

Thus we explicitly state that no action that might put something on A or C occurred in the interval[15], save of course

[15] We need to state that no event whose result lies in the interval

the action of putting A on the top of C.

$$\forall s, b_1, e.e = Move(b_1, Top(A)) \rightarrow \neg\left(\begin{array}{l} Occurs(e,s) \wedge S0 \leq Result(e,s) \\ \leq Result(Move(A, Top(C)), S0) \end{array}\right)$$

$$\forall s, b_1, e.e = Move(b_1, Top(C)) \wedge (s \neq S0 \vee b \neq b_1) \rightarrow \neg\left(\begin{array}{l} Occurs(e,s) \wedge \quad S0 \leq Result(e,s) \\ \leq Result(Move(A, Top(C)), S0) \end{array}\right)$$

We can now prove that

$$Holds(On(A, Top(C)), Result(Move(A, Top(C)), S0)),$$

as the above principal allows us to prove that A and C are clear for the entire interval.

We cannot prove that C remains on the Table however, as there may be events that put C on top of other blocks. These events will not make C unclear, so they do not block the action of putting A on C. To prove that C remains on the table we would need,

$$\forall s, x. \neg\left(\begin{array}{l} Occurs(Move(C,x), s) \wedge \\ \quad S0 \leq Result(Move(C,x), s) \\ \qquad \leq Result(Move(A, C), S0). \end{array}\right)$$

It is notable that there was no need to state that no other actions occurred. It sufficed[16] to say that no other events occurred that might cause a precondition of an event in our narrative to fail. The notion that we need only state that certain events did not occur becomes very important if we wish to axiomatize domains in a way that will later allow them to be conjoined. In fact, the motivating property for developing this new axiomatization was to allow separate axiomatizations, that do not interfere with each other, to be conjoined. This is not possible in the old-style situation calculus, as we explicitly list the sequence of actions that occurs. It is also not possible if we state that the only events that occur are those mentioned, as is sometimes done in narrative reasoning.

The reasoning that we did was not significantly more difficult than the usual reasoning in the situation calculus. We needed to check a few more conditions, namely that blocks remained clear, but strategies, such as goal regression[Rei91] continue to be effective.

or at the endpoints, thus the use of $\leq$. Sometimes, especially when we are checking preconditions of events, we will only need to show that nothing had an effect strictly before the end, and this we will only need to show $<$. When we try to use inertia we will need to show the $\leq$.

[16] In this paper we state that the other events do not happen monotonically. These statements can be inferred nonmonotonically from sentences that tell which occurrences and what fluents are explicitly *stated* to occur and hold in our narrative, and the axiomatization of $Changes$. A fluent is relevant if it is a precondition or an effect of a stated event that occurs, or if the fluent's value is stated in the narrative. This gives us a notion of what the relevant fluents and events are in terms of what fluents and events are explicitly given in the narrative. We then state that no other events occur that would change the effects of the relevant fluents. We avoid explaining this reasoning, as the machinery we currently use is quite complex.

4.2 EVENTS WITH MULTIPLE EFFECTS

In general an event may have more than one effect. The preconditions for each effect may differ, thus preconditions may be parameterized by the effect. Furthermore, each effect may occur at a different time.

4.2.1 Extending $Result$

If there is more than one effect of an event e, we write $Result(e, s)$ for the time of the main effect, and $Result(e, f, s)$ for the time of the effect of changing f. For instance, flying from Glasgow to London has as its resulting situation the situation where you arrive in London. However, another effect of this event is to no longer have your ticket, as the air-hostess takes it from you. The situation where she takes the ticket off you is picked out by $Result(Fly(Glasgow, LHR, T_1), takenby(Steward, ticket), s)$. At this situation, the fluent $Has(ticket)$ is also made false. We use other situations like the time the airline-steward takes your ticket, rather than explicit times, as explicit times, like all numerical values are less natural—the numbers are hard to get. The statement that you no longer have your ticket after the air-hostess takes it is very intuitive, while the statement that your no longer have your ticket after n minutes, for some definition of n is not.

4.2.2 Implied events

We can deal with events having multiple effects at different times by stating that certain events trigger other events. Thus we might write,

$$\forall s, l, l'.occurs(Fly(l, l', Ticket), s) \rightarrow occurs(take(Steward, Ticket), s).$$

This is an alternate way to model the notion that the airline steward takes your ticket during the flight.

We now consider narratives in two domains. One concerns stacking blocks, the other a plane journey. We show that we can axiomatize these two narratives separately, but in such a way that their conjunction is consistent.

5 GLASGOW, LONDON, MOSCOW AND NEW YORK

The object of this section is to give narratives illustrating the treatment of concurrent events in two cases. The first is when two sub-narratives do not interact, and the second is when they do. The first sub-narrative is ordinary block stacking (as discussed in many situation calculus papers), and we suppose the stacking to be done by a person called Daddy in New York.

In the second sub-narrative, the actor is named Junior, and he wants to fly from Glasgow to Moscow via London. The story is taken from earlier web-published but widely circulated manuscripts [McC92] discussing how circumscription could be used to treat an unexpected obstacle to a plan, and [McC95] how narratives should be represented. This story is also used by Shanahan in [Sha97] in Chapter 10 as an example to motivate a use of context.

These two sub-narratives do not interact, and thus give an example of our first goal, a treatment of non-interacting narratives that can be conjoined consistently.

Because we want to treat interacting events, we make life more complicated for Junior. If he loses his ticket, he must wire Daddy in New York for money. Daddy, who normally indulges Junior, has to interrupt his block stacking and sell a block in order to get the money to send Junior. In this part of the narrative we have an example of adding details to an event. We state the event of Junior getting money occurs, we also give a sequence of events, Daddy stacking blocks until block3 is clear, then selling block 3, receiving money and sending it to Junior. The sequence *realizes* the single event of getting money. We show that both statements are consistent with each other, and the explanation can be consistently conjoined onto the narrative that mentions only the first event.

The following uses the axiomatizations of traveling and commerce and blocks-world in the Appendix. In the text we only give those axioms that are particular to the story.

We give axiomatizations of both the narratives where Junior loses his ticket, and contacts Daddy who sends him money, that Daddy raises by selling a block (In New York, blocks are made of Gold). Naturally Daddy has to clear the block before selling it, so the narratives interact in a non-trivial way.

Narrative 1

In this narrative Junior doesn't lose his tickets, T_1 and T_2 and gets to Moscow without asking for help. Daddy stacks blocks in New York. There is no interaction, and nothing is said about the time relations between the two sub-narratives.

$$\begin{array}{l} Holds(At(J, Glasgow), S0) \\ Occurs(Fly(Glasgow, LHR, T_1), S0) \\ Dest(T_1) = LHR \wedge Source(T_1) = Glasgow \\ Holds(Has(J, T_1), S0) \\ Dest(T_2) = Moscow \wedge Source(T_2) = LHR \\ Holds(Has(J, T_2), S0) \\ Result(Fly(Glasgow, LHR, T_1), S0) < S1 \end{array} \tag{4}$$

We should be able to infer:

$$Holds(At(J, LHR), S1)$$

To infer this we need to know that,

$$\begin{array}{l} \forall s', e. S0 \leq Result(e, s') < \\ Result(Does(J, Fly(Glasgow, LHR, T_1)), S0) \rightarrow \\ \neg(Occurs(e, s') \wedge Succeeds(e, Has(J, T_1), s')) \end{array}$$

That is, no event occurs that would cause Junior to lose his ticket before he has to give it to the air-hostess[17]. We actually state the following stronger fact[18], that no events that would cause Junior to no longer have a ticket occur, save of course flying from Glasgow.

$$\begin{array}{l} \forall s', e. (t = T_1 \vee t = T_2) \wedge \\ \quad (e \neq Fly(Glasgow, LHR, T_1) \vee s' \neq S0) \wedge \\ \quad S0 \leq Result(e, s') < S1 \rightarrow \\ \neg(Occurs(e, s') \wedge Succeeds(e, Has(J, t), s')) \end{array} \tag{5}$$

When Junior is in London, inertia, and the instance of the above axiom with $t = T_2$, gets us that Junior still has the ticket to Moscow. As for the ticket to London, we would infer that he does not have it as we brought up the fact that a ticket is used up when one takes the flight the ticket is for. That is certainly a part of the knowledge of anyone who travels using tickets. Thus someone who had traveled by bus would infer it about airplane travel. Indeed it could be inferred from more general principles about commerce, e.g. that a seller doesn't want to allow the buyer to get an arbitrary number of what he has a paid for one of. However, anyone who travels has the more specific information and doesn't need to infer it from general principles about commerce. Indeed he may never have formulated any general principles about commerce.

$$\begin{array}{l} Occurs(Fly(LHR, Moscow, T_2), S1) \\ Result(Fly(LHR, Moscow, T_2), S1) < S2 \end{array} \tag{6}$$

We wish to infer,

$$Holds(At(Junior, Moscow), S2)$$

Again we need to know that no bad events occur, that is, Junior doesn't lose any tickets.

$$\begin{array}{l} \forall s', e. (e \neq Fly(LHR, Moscow, T_2) \vee s' \neq S1) \wedge \\ \quad S1 \leq Result(e, s') < S2 \rightarrow \\ \neg(Occurs(e, s') \wedge Succeeds(e, Has(J, T_2), s')) \end{array} \tag{7}$$

We call these sentences $Nar1J$, that is the sentences from 4 to 7. Now we begin Daddy's life as a block stacker.

[17] If we wished that the air-hostess took Junior's ticket at another time, we might use our three argument version of result and write,

$$\begin{array}{l} \forall s', e. S0 \leq Result(e, s') < \\ Result\left(\begin{array}{c} Does(J, Fly(Glasgow, LHR, T_1)), \\ Has(J, T_1), S0 \end{array}\right) \rightarrow \\ \neg(Occurs(e, s') \wedge Succeeds(e, Has(J, T_1), s')). \end{array}$$

[18] Whether or not the stronger fact is warranted depends on whether we wish to state that no event that *might* cause Junior to lose his ticket happens, or no event that *does* cause Junior to lose his ticket happens.

We have no $\leq$ relation between the situations $S0$ and $S0'$ and know nothing of their temporal relations. If we asserted $S0 < S0' < S1$, then we could conclude that Junior still had the tickets in $S0'$. Also asserting $S0' = S0$ would do no harm to the conclusions drawn about either sub-narrative.

$$\begin{array}{l} Holds(At(D, NY), S0') \\ Holds(Has(D, A_1), S0') \\ Holds(Has(D, A_2), S0') \\ Holds(Has(D, A_3), S0') \\ Holds(On(A_3, Top\ A_1), S0') \\ \forall b.Holds(Clear(b), S0') \equiv b \neq A_1 \\ Holds(On(A_1, Table), S0') \\ Holds(On(A_2, Table), S0') \\ Occurs(Does(D, Move(A_3, Table)), S0') \\ Result(Does(D, Move(A_3, Table)), S0') < S1' \\ Occurs(Does(D, Move(A_2, Top\ A_1)), S1') \\ Result(Does(D, Move(A_2, Top\ A_1)), S1') < S2' \\ Occurs(Does(D, Move(A_3, Top\ A_2)), S2') \\ Result(Does(D, Move(A_3, Top\ A_2)), S2') < S3' \end{array} \tag{8}$$

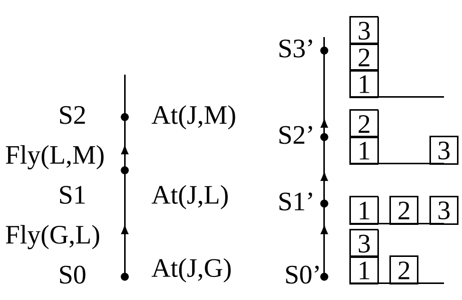
We also need to know that no other actions that would interrupt the block stacking[19] occur.

$$\begin{array}{l} \forall s', e, a, b.S0' \leq Result(e, s') \leq S3' \wedge \\ \quad (s' \neq S0' \vee e = Move(A_3, Table)) \wedge \\ \quad (s' \neq S1' \vee e = Move(A_2, Top\ A_1)) \wedge \\ \quad (s' \neq S2' \vee e = Move(A_3, Top\ A_2)) \rightarrow \\ \neg(Occurs(e, s') \wedge Succeeds(e, On(a, Top(b)), s')) \end{array} \tag{9}$$

We call the sentences from 8 to 9 $Nar1D$ We now notice that if B is the axiomatization of blocksworld in the Appendix, and T is the axiomatization of traveling, then

$$\begin{array}{l} B \wedge T \wedge Nar1D \wedge Nar1J \models \\ \quad Holds(On(A_3, Top\ A_2), S3') \wedge \\ \quad Holds(On(A_3, Table), S1') \wedge \\ \quad Holds(At(J, LHR), S1) \wedge \\ \quad Holds(At(J, Moscow), S2) \end{array} \tag{10}$$

Thus we can derive the obvious conclusions of our narrative. We further note that the two narratives are consistent.



Narrative 2

In this narrative Junior loses the ticket and sends a telegram to Daddy asking for money. Daddy, who normally indulges Junior, sells a block and sends Junior the money, who buys a London-Moscow ticket and goes on to Moscow.

We chose a telegram rather than a telephone call, because we would not want to tell about a telephone call as a sequence of statements by Junior and Daddy but rather to regard its result as a joint action, e.g. an agreement that Junior and Daddy would do certain actions.

Note also we haven't treated what Daddy now knows as the result of the telegram. It seems that treating knowledge and treating agreement are similar in their requirement for treating intentional entities. The intentional state that Junior has requested that Daddy send him the money is not merely that Daddy knows that Junior wants Daddy to send him the money. Also the agreement is likely to have something like a bit of narrative as an argument, e.g. a set of actions that Junior and Daddy will do with only partial time relations between the actions.

Here we include sentences 4 and 5. Up to here, narrative 2 is the same as narrative 1. We will also need the sentences 8 and 9.

$$Occurs(Loses(J, T_2), S1) \tag{11}$$

This contradicts 7, which stated that no event that lost the ticket happened before $S2$. We want to regard losing the ticket as something that happens to Junior rather than as something he does. That's why we don't write $does(J, lose\ ticket(LHR, Moscow))$. The bad consequences of doing the latter would arise when we get around to writing laws that quantify over voluntary actions. We will use some of the same names now for situations that are different than in narrative 1.

$$\begin{array}{l} Result(Loses(J, T_2), S1) < S2 \\ \neg Holds(Has(J, Cash), S2) \end{array} \tag{12}$$

$$Occurs(Does(J, Telegraph(D, Request\ Send\ Cash)), S2) \tag{13}$$

Here we intend to have two explanations for what happens next. One is the simple observation that Daddy does sell a block, and send the money to Junior. This is a simple sequence of events, like we detailed earlier.

However we also know that as a dutiful father, Daddy will get money to Junior. Thus we can predict the event of Daddy getting money to Junior. Here we are treating Daddy as if his actions were determined by our inputs. Sometimes it is useful to describe people in that way. In more elaborate narratives we would need to reason about Daddy mental processes, but for this case we can treat him as an automaton.

The following axiom characterizes what Daddy does when

[19] If we wish to restrict this to block stacking in New York we would add a conjunct $Holds(In(a, New\ York), s') \wedge Holds(In(b, New\ York), s')$ to the left hand side of the implication.

he receives a request from Junior.

$$\forall s.Holds\left(\begin{array}{l}Happens(Gets(J, Cash)),\\ Result\left(\begin{array}{c}Receives(D, Telegram\text{-}from\\ (J, Request\ Send\ Cash)), s\end{array}\right)\end{array}\right)$$

$$S3.5 = Result\left(\begin{array}{c}Receives(D, Telegram\text{-}from\\ (J, Request\ Send\ Cash)), S2\end{array}\right) \quad (14)$$

This is an example of a triggered action, as we have the defining rule for $Holds(Happens(e), s)$,

$$\forall e, s.Holds(Happens(e), s) \equiv Occurs(e, s).$$

We now state that the money arrives before $S3$, when Junior buys the ticket.

$$Result(Gets(J, Cash), S3.5) < S3$$

We now give the other facts about occurrences.

$$\begin{array}{l}S3' =\\ Result(Does(J,\ Telegraph(D, Request\ Send\ Cash)), S2)\\ \neg Holds(Has(D, Cash), S3')\\ Occurs(Does(D, Sell\ A_3), S3')\\ Result(Does(D, Sell\ A_3), S3') < S4'\\ Occurs(Does(D, Send(J, Cash)), S4')\\ Result(Does(D, Send(J, Cash)), S4') < S3\\ Occurs(Does(J, Buy\ ticket(T_2)), S3)\\ Result(Does(J, Buy\ ticket(T_2)), S3) < S4\\ Occurs(Does(J, Fly(LHR, Moscow)), S4)\\ Result(Does(J, Fly(LHR, Moscow)), S4) < S5\end{array}$$

We also need to know that no events occur that would divert the money in the meantimes between these events and the result of the previous events.

$$\begin{array}{l}\forall s', e.(e \neq Does(D, Send(J, Cash)) \vee s' \neq S4') \wedge\\ Result(Does(D, SellA_3), S3') \leq\\ Result(e, s') \leq S4' \rightarrow\\ \neg\left(\begin{array}{l}Occurs(e, s') \wedge\\ Succeeds(e, Has(D, Cash), s')\end{array}\right)\end{array} \quad (15)$$

$$\begin{array}{l}\forall s', e.(e \neq Does(J, Buy\ ticket(T_2)) \vee s' \neq S3) \wedge\\ Result(Does(D, Send(J, Cash)), S4') \leq\\ Result(e, s') \leq S3 \rightarrow\\ \neg\left(\begin{array}{l}Occurs(e, s') \wedge\\ Succeeds(e, Has(J, Cash), s')\end{array}\right)\end{array}$$

$$\begin{array}{l}\forall s', e.(e \neq Fly(LHR, Moscow, T_2) \vee s' \neq S4) \wedge\\ Result(Does(J, buy\ ticket(T_2)), S3) \leq\\ Result(e, s') \leq S4 \rightarrow\\ \neg\left(\begin{array}{l}Occurs(e, s') \wedge\\ Succeeds(e, Has(J, T_2), s')\end{array}\right)\end{array}$$

We now consider the consequences of narrative two. Let $Nar2$ be those sentences directly above and the sentences from 4 to 5 and 8 and 9 and 11 to 15.

$$\begin{array}{l}Nar2 \models\\ \quad \neg Holds(Has(J, T_2), S2) \wedge\\ \quad Holds(Has(D, Cash), S4') \wedge\\ \quad Holds(Has(J, Cash), S3) \wedge\\ \quad Holds(Has(J), T_2, S4) \wedge\\ \quad Holds(At(J, Moscow), S5)\end{array} \quad (16)$$

Most interestingly we can derive the occurrence of a triggered action:

$$Nar2 \wedge Dep \wedge U \models \quad (17)$$

$$Occurs\left(\ gets(J, Cash), S3.5\ \right)$$

We have two explanations for Junior receiving the money, the $gets$ event, and the $send$ event. We cannot tell which happens first, or if they happen simultaneously. Thus our formalism allows us to add detail of an event without contradiction.

5.1 ELABORATIONS

Interpolating unconnected situations and events into a narrative does not harm the conclusions. For example, we could put situations $S0.5$ and $S0.7$ between $S0$ and $S1$, and suppose that Junior reads a book on the airplane during the inner interval. The previous statements about what holds when Junior arrives in London should still seem ok. Indeed we have that when we add

$$\begin{array}{l}Occurs(Read(J, Book), S1.5)\\ \wedge S1 < S1.5 < Result(Read(J, Book), S1.5) < S2\\ \forall s.Holds(Intelligent(P), Result(Read(P, Book), s))\end{array}$$

we can conclude all our previous sentences, plus some more about Junior's intelligence, something we earlier did not have an opinion about.

In our second narrative, after $S2$ we have two possible explanations of how Junior gets the cash to buy his ticket. One explanation is that Daddy always gets cash to Junior. We also have the more detailed explanation that Daddy sells A_3 and sends the proceeds to Junior. The more detailed explanation is an elaboration of how Daddy got them cash to Junior. If is worth noting that both explanations can co-exist in our narrative without inconsistencies.

5.2 ELABORATION OF NARRATIVES

Suppose we are asked, "How did Junior fly from Glasgow to London?" and want to respond with facts about taking a taxi to the airport, presenting his ticket at the check-in counter, going to the gate, getting on the airplane, taking his assigned seat, etc. We can add this additional narrative with its intermediate situations, and we can throw in reading the book if we like. There is no reason to discard $Occurs(Does(J, Fly(Glasgow, LHR, T_1)), S0)$.

We merely have a redundant way of reaching the same conclusion. This is allowed in our formalism, and this property is demonstrated by the two ways in which we describe how Daddy gets the money for Junior.

However, we would like a sentence relating the more detailed narrative to the less detailed narrative, i.e. of asserting that one *realizes* the other. For this we will at least need narratives as objects, and this has not yet been introduced.

Note that the relation $Elaborates(N2, N1)$, when we get around to introducing it, will not be like the relation between a subroutine call and the subroutine. $N1$ will not in any sense be the definition of $N2$. $N2$ could be realized in a number of ways, only one of which corresponds to $N1$.

5.3 PLANNING AND PREDICTION

We would like to treat the circumstances of the previous narrative from the point of view of planning. In that case we need to be explicit about the consequences of actions and other events. The difference between planning and narrative is that in narrative we know that events and actions will succeed. This allows us to make assumptions we otherwise could not make. We can also assume that all the important effects of actions are mentioned. When we plan we need to show that we have taken into account all the important effects.

Let us consider the purposes of Junior and Daddy and predict what actions they will take and what the outcome will be. Of course, Junior losing the ticket will be an unpredicted event. We just throw it in, but then we should be able to predict what Junior and Daddy will subsequently do. This seems more difficult than either planning or predication.

5.4 PHILOSOPHICAL CONSIDERATIONS

Reality may be regarded as the deterministic limit of nondeterminist approximations. In what a human or robot can know about the world many events are not inevitable. In any human account, it did not have to be raining when Benjamin Franklin first arrived in London. Indeed, maybe it wasn't. Even if the world is deterministic, any achievable description of it is nondeterministic. Elaborations of particular narratives sometime remove some of the nondeterminism by accounting for the causes of particular events and for fluents holding in the results of these events.

Therefore, it may be worthwhile to regard the world as determinist and suppose that every event has causes whether we know them or not. Thus any particular nondeterminism is potentially eliminable.

It might be supposed that quantum mechanics vitiates these considerations, but we don't think it requires modifications on the common sense level. Free will in a determinist world is discussed in [MH69].

5.4.1 REMARKS

We have always felt that the careful classification of the ways in which events can overlap is unnecessary for almost all common sense reasoning. We think this article shows it. Moreover, it is also usually unnecessary to combine concurrent events into compound events as do Gelfond, Lifschitz and Rabinov [GLR91].

Acknowledgments

This research was partly funded by DARPA (ONR) grant # N00014-94-0775 under ARPI and by the Air Force Office of Scientific Research Grant # F49620-97-1-0207 under the DARPA HPKB project and New World Vistas.

References

[Cos97] Tom Costello. *Non-monotonicity and Change*. PhD thesis, Stanford University, 1997.

[Cos98a] Tom Costello. Domain Formula Circumscription. *Journal of Logic Language and Information*, 1998. to appear.

[Cos98b] Tom Costello. Minimizing the Effects of Actions. In *Proceedings of the Fifth International Symposium on Commonsense Reasoning*, 1998.

[GLR91] Michael Gelfond, Vladimir Lifschitz, and Arkady Rabinov. What are the limitations of the situation calculus? In Robert Boyer, editor, *Automated Reasoning: Essays in Honor of Woody Bledsoe*, pages 167–179. Kluwer Academic, Dordrecht, 1991.

[KS97] R. Kowalski and F. Sadri. Reconciling the Situation Calculus and Event Calculus. *Journal of Logic Programming*, 31:39–58, 1997.

[McC59] John McCarthy. Programs with Common Sense[20]. In *Mechanisation of Thought Processes, Proceedings of the Symposium of the National Physics Laboratory*, pages 77–84, London, U.K., 1959. Her Majesty's Stationery Office. Reprinted in McC90.

[McC63] J. McCarthy. Situations, Actions and Causal Laws. Technical Report Memo 2, Stanford University Artificial Intelligence Project, Stanford, CA, 1963.

[20] http://www-formal.stanford.edu/jmc/mcc59.html

[McC89] John McCarthy. Artificial intelligence, logic and formalizing common sense. In Richmond Thomason, editor, *Philosophical Logic and Artificial Intelligence*. Klüver Academic, 1989.

[McC92] John McCarthy. Overcoming an Unexpected Obstacle. Available as http://www-formal.stanford.edu/jmc/elaboration.html, 1992.

[McC95] John McCarthy. Situation Calculus with Concurrent Events and Narrative[21]. 1995. Contents subject to change. URL will remain.

[MH69] J. McCarthy and P. Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In D. Michie, editor, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, Edinburgh, UK, 1969.

[PR95] J. Pinto and R. Reiter. Reasoning about Time in the Situation Calculus. *Annals of Mathematics and Artificial Intelligence*, 14(2-4):251–268, September 1995.

[Rei91] R. Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation*, pages 359–380. Academic Press, 1991.

[Rei93] R. Reiter. Proving properties of states in the situation calculus. *Artificial Intelligence*, 64(2):337–351, December 1993.

[Sha97] Murray Shanahan. *Solving the Frame Problem, a mathematical investigation of the common sense law of inertia*. M.I.T. Press, 1997.

APPENDIX

BLOCKSWORLD

Our blocks world has 4 sorts, situations s, blocks b, locations l and actions a. These are all disjoint.

We have a situation constant $S0$, other situation constant Sn and Sn' for various n's, a set of block constants $A_1, \ldots, A_n, \ldots$, where $n \in \omega$ and one block location constant $Table$. We also have constants $A = A_1$, $B = A_2$ and $C = A_3$.

All blocks are unique, but we do not postulate domain closure.

$$A_i \neq A_j | i \neq j$$

[21] http://www-formal.stanford.edu/jmc/narrative.html

We have block locations[22], which are the Top of a block, or are the $Table$.

$$\forall l. \exists b. Top(b) = l \vee l = Table$$

All distinct block location terms denote distinct locations.

$$\forall b, b'. Top(b) = Top(b') \rightarrow b = b'$$

$$\forall b. Top(b) \neq Table$$

We have a function from actions and situations to situations, $Result(a, s)$, and a function from blocks and locations to actions, $Move(b, l)$, which gives the action where block b is moved to location l.

All distinct action terms are distinct.

$$\forall b, b', l, l'. Move(b, l) = Move(b', l') \rightarrow b = b' \wedge l = l'$$

We have the foundational axioms for situation calculus we considered earlier.

We have fluents, $Holds(On(b, l), s)$ which states that b is on location l in situation s, and $Holds(Clear(l), s)$. $Holds(Clear(l), s)$ is fully defined in terms of On.

$$\begin{array}{l} \forall l, s. Holds(Clear(l), s) \equiv \\ (\exists b. \forall b'. l = Top(b') \wedge \neg Holds(On(b', Top(b)), s)) \vee \\ \quad l = Table \end{array}$$

We now add the obvious definition of $Changes$ for $Move(b, l)$ actions. That is, there is a change in $On(b, l')$ and $On(b, l)$ when g_2 contains $On(b, l')$ for an l' not equal to l, and $l \neq Top(b)$, and g_1 contains $Clear(l)$ and $Clear(top(b))$.

$$\begin{array}{l} \forall b, b', l, l', g_1, g_2. \\ Changes(Move(b, l), On(b', l'), g_1, g_2) \equiv \\ l \neq l' \wedge l \neq Top(b) \wedge b = b' \wedge \\ \left(\begin{array}{l} (b = b' \wedge g(\neg On(b', l'))) \vee \\ (b \neq b' \wedge g(On(b', l'))) \end{array} \right) \wedge \\ G(Clear(l)) \wedge G(Clear(l')) \end{array}$$

This concludes the axiomatization of blocksworld, we could add domain constraints, but this is not necessary for the reasoning we do in this paper. We now present an axiomatization of traveling, followed by an axiomatization of buying selling and sending and receiving.

TRAVELING AND COMMERCE

Our events are flying, doing actions, getting, receiving and losing. Our actions are selling, sending, telegraphing. Distinct event terms are distinct. We have two people constants, Junior and Daddy. We have among our objects Cash,

[22] These are not the same as geographical locations like New York or London. We use l to range over both, which is unfortunate.

a message ($Request\ Send\ Cash$), cities, including London, Glasgow and Moscow. We also have tickets, and functions which yield destination $Dest$ and source $Source$ of a flight, given a ticket for that flight. We have unique names for all fluent terms, and thus for all the terms that can appear in fluents. Our fluent forming functions are At which takes a person and a place, Has which takes a persona and a thing, $Happens$, which takes an event, and the earlier fluents of On and $Clear$. Our sorts are disjoint, and the sorts of variables are to be inferred by their use.

$$\begin{array}{l}\forall s, P, l, t. Holds(At(P, l), s) \wedge Source(t) = l \wedge \\ Dest(t) = l' \wedge \\ (\forall s'. s \leq s' < Result(Fly(l, l', t), s) \rightarrow \\ Holds(Has(P, t), s')) \rightarrow \\ Holds(At(P, l'), Result(Fly(l, l', t), s)) \wedge \\ \neg Holds(Has(P, t), Result(Fly(l, l', t), s))\end{array}$$

This is equivalent to

$$\begin{array}{l}\forall l, l', t, P. \\ Changes(Fly(l, l', t), Has(P, t), g_1, g_2) \leftarrow \\ \quad Source(t) = l \wedge Dest(t) = l' \wedge \\ \quad g_2(At(P, l)) \wedge g_1(Has(P, t))\end{array}$$

As this is the only effect axiom for flying, we can change this to the equivalence.

$$\begin{array}{l}\forall l, l', t, P, g_1, g_2. \\ Changes(Fly(l, l', t), Has(P, t), g_1, g_2) \equiv \\ \quad Source(t) = l \wedge Dest(t) = l' \wedge \\ \quad g_2(At(P, l)) \wedge g_1(Has(P, t))\end{array}$$

$$\forall s, P, o. \neg Holds(has(P, o), Result(lose(P, o), s))$$

This immediately gives,

$$\forall s, P, g_1, g_2, o. Changes(Lose(P, o), Has(P, o), g_1, g_2)$$

as any set of preconditions is sufficient. We add the obvious axioms that describe $Changes$ for the following effect axioms using the same method.

$$\begin{array}{l}\forall r, P, s. Holds(Happens(Receives(P', \\ \qquad\qquad Telegram\text{-}from(P, r))), \\ \qquad Result(Does(P, Telegraph(P', r)), s))\end{array}$$

$$\forall s. Holds\left(\begin{array}{l}Happens(Gets(J, Cash)), \\ Result\left(\begin{array}{c}Receives(D, Telegram\text{-}from(J, \\ Request\ Send\ Cash)), s\end{array}\right)\end{array}\right)$$

$$\begin{array}{l}\forall s, P, o. Holds(Has(P, o), s) \wedge Holds(Clear(o), s) \rightarrow \\ \quad Holds(Has(P, Cash), Result(Does(P, Sell(o)), s)) \\ \forall s, P, o. Holds(Has(P, o), s) \wedge Holds(Clear(o), s) \rightarrow \\ \quad \neg Holds(Has(P, o), Result(Does(P, Sell(o)), s)) \\ \forall s, P, P', o. Holds(Has(P, o), s) \rightarrow \\ \quad Holds(Has(P', o), Result(Does(P, Send(P', o)), s)) \\ \forall s, P. Holds(Has(P, Cash), Result(Gets(P, Cash), s))\end{array}$$