Default Reasoning about Spatial Occupancy

Murray Shanahan

Imperial College
Department of Computing,
180 Queen's Gate,
London SW7 2BZ.
England.

Tel: +44 71 589 5111 x 5076
Email: mps@doc.ic.ac.uk

March 1993
Revised July 1993
Second Revision July 1994

Abstract

This paper describes a default reasoning problem, analogous to the frame problem, that arises when an attempt is made to construct a logic-based calculus for reasoning about the movement of objects in a real-valued co-ordinate system. A number of potential solutions to this problem are examined. Particular attention is given to the interaction between the default reasoning required by these solutions and that required to overcome the frame problem, especially when the latter demands an "existence of situations" axiom.

Introduction

Much commonsense reasoning about the everyday world concerns the spatial properties of objects — their shapes and locations — and how these properties change over time. Accordingly, if we are to develop a formal theory of commonsense, we need a precisely defined language for talking about shape, spatial location and change. The theory will include axioms, expressed in that language, that capture domain-independent truths about shape, location and change, and will also incorporate a formal account of any non-deductive forms of commonsense inference that arise in reasoning about the spatial properties of objects and how they vary over time.

This paper combines the situation calculus of McCarthy and Hayes [1969] with a formal language for talking about shape similar to that of Davis [1987]. When this language is used to describe the effects of simple move actions, a problem arises which, like the frame problem, appears to demand some form of default reasoning. It is an obvious precondition of any action that changes an object's location that the object's destination must be unoccupied. With the Blocks World, in which similar preconditions are required, this is not a source of trouble. But space in the Blocks World is normally represented simply as a small set of locations. In the kind of more sophisticated language required for a proper treatment of shape, space has to be represented via a real-valued co-ordinate system.

With space represented in this way, it becomes difficult to describe a situation completely with respect to which regions of space are occupied and which are empty. With only an incomplete description of spatial occupancy, it is impossible to prove that the preconditions for moving an object hold in a situation, because those regions which are unoccupied are not explicitly specified. To overcome this, we would like to be able to capture the commonsense law that space is normally empty. This paper attempts to formalise such a commonsense law.

The incorporation of multiple laws of commonsense into a theory must be done with great care if they are to interact properly. In this case, we must ensure that the commonsense law that "space is normally empty" operates correctly in the presence of the commonsense law that "actions normally don't affect fluents". One of the most successful attempts to overcome the frame problem, in other words to formalise the second of these laws, is state-based minimisation [Baker, 1991]. This approach is adopted in this paper because it copes so well with domain constraints. But one of the features of state-based minimisation is its demand for a so-called "existence of situations" axiom. This axiom guarantees that a situation exists for every legitimate combination of fluents. It is the presence of this axiom which makes the formalisation of the first of these laws difficult.

1. Space and Shape

To see how the need arises for a commonsense law that space is normally empty, we first need a language for talking about space and shape. The problem of defining a formal language for talking about change has received plenty of attention in the knowledge representation literature, and a variety of choices for the ontology and predicates of such a language have been suggested, along with collections of relevant axioms (see, for example, [McCarthy & Hayes, 1969], [McDermott, 1982], [Allen, 1984], [Hayes, 1985], [Kowalski & Sergot, 1986]). The task of defining a formal, logic-based language for talking about shape and spatial location has been given less attention, notable exceptions being the work of Hayes [1985], Shoham [1985], Davis [1987], [1990, Chapter 6], Kaufman [1991], and Randell, *et al.* [1992].

The language of this paper is a language of many-sorted first-order predicate calculus with equality. I will begin by introducing space, regions and points. I will assume that space is two-dimensional and corresponds to the set $S = R \times R$, where R is the set of reals. A *region* is a subset of S, and a *point* is a member of S. Accordingly, the language includes sorts for regions and points, with variables r, r0, r1, r2, etc. and p, p0, p1, p2, etc. respectively. Note that regions do not have to be connected. An example of an object occupying a non-continuous region would be a mist comprising a number of separate particles. I will consider only interpretations in which points are interpreted as pairs of reals, in which regions are interpreted as sets of points, and in which the $\in$ predicate and the comparative predicates $<, \leq, >, \geq$ all have their usual meanings.

A convenient way to represent a shape is as a region. I will assume that every shape has a conventional reference point, which is the origin $\langle 0,0 \rangle$. For regular shapes, it is convenient to make this reference point the shape's centre. For example, a circle of radius d units could be defined using the following sentence (Ex1).[1] The *distance* sort is introduced here, which has variables d, d0, d1, d2, etc, and which is interpreted by R. It is assumed for this example that the language includes the function Disc being defined, which maps a distance onto a region, and the function Distance, which is defined to map two points onto the distance between them. Neither of these functions is part of the basic language, though Distance would be useful for many other examples.[2]

$$p \in \text{Disc}(d) \leftrightarrow \text{Distance}(p, \langle 0,0 \rangle) \leq d \qquad \text{(Ex1)}$$

Much more complicated shapes than this can be described using this simple language, but this is not the concern of the present paper. Space is occupied by *objects*. Each object has a unique *shape*. I will assume that an object's shape is fixed over time, but it would be easy to extend the language to allow objects with changing shapes. The sort for objects has variables ob, ob0, ob1, ob2, etc. The shape of an object is a region. The function Shape maps an object onto its shape.

For example, if D1 is an object, then given (Ex1), the sentence (Ex2) represents the fact that D1's shape is a circle with radius five units. Similarly, if D2 is an object, the sentence (Ex3) represents that its shape is a circle of unknown radius.

$$\text{Shape}(D1) = \text{Disc}(5) \qquad \text{(Ex2)}$$

$$\exists d\ [\text{Shape}(D2) = \text{Disc}(d)] \qquad \text{(Ex3)}$$

Since the spatial properties we want to capture are *fluents*, that is to say they are subject to variation over time, they will be reified, so that they can appear as arguments to temporal predicates. A sort for fluents is introduced, with variables f, f0 f1, f2, and so

---

1 A disc is represented here by a closed set. It could equally well have been represented by an open set. The problem of defining when two objects can be said to touch is beyond the scope of this paper.

2 All variables are universally quantified unless otherwise shown. A suitable set of unqieness-of-names axioms will be assumed.

forth. A sort for situations is also introduced, with variables s, s0, s1, s2, and so forth. A *situation* is an instantaneous snapshot of the state of the world. The formula Holds(f,s) represents that fluent f is true in situation s.

Now we can introduce fluents for describing an object's location and the space it occupies. The fluent Occupies(ob,r) represents that ob uniquely occupies the region r. No object can occupy two regions at the same time. The meaning of this fluent is such that although an object occupies a region r, it does not occupy any subset of r. The region occupied by an object is, in this sense, maximal. We have the following axioms. Axiom (Sp1) says that an object can occupy at most one region, and Axiom (Sp2) says that no two objects can occupy overlapping regions. Axiom (Sp2) will have to be made more liberal for domains in which an object can be decomposed into parts which are themselves objects.

$$[\text{Holds}(\text{Occupies}(ob,r1),s) \wedge \text{Holds}(\text{Occupies}(ob,r2),s)] \rightarrow r1=r2 \tag{Sp1}$$

$$[\text{Holds}(\text{Occupies}(ob1,r1),s) \wedge \text{Holds}(\text{Occupies}(ob2,r2),s) \wedge ob1 \neq ob2] \rightarrow \neg \exists p\, [p \in r1 \wedge p \in r2] \tag{Sp2}$$

These axioms are examples of domain constraints. The presence of domain constraints like these will restrict our choice of solution to the frame problem. A further domain constraint is required to relate an object's location to the region it occupies. The fluent Location(ob,p) represents that ob occupies the region obtained by displacing the shape of ob by d1 units horizontally and d2 units vertically, where p=⟨d1,d2⟩. Recall that the conventional reference point of a shape is the point ⟨0,0⟩. So in effect, the fluent Location(ob,p) represents that ob is positioned with its centre at point p.

Note that there is a difference between space which is actually occupied, and space which is simply used to describe a shape. Although there can be many shapes whose reference point is the origin ⟨0,0⟩, only one object can actually occupy a region including the origin. The relationship between Location and Occupies is given by the following axiom. The function Displace(r,p) denotes the region obtained by displacing the region r by d1 units horizontally and d2 units vertically, where p=⟨d1,d2⟩.

$$\text{Holds}(\text{Occupies}(ob,r1),s) \leftarrow \text{Shape}(ob)=r2 \wedge \text{Holds}(\text{Location}(ob,p),s) \wedge r1=\text{Displace}(r2,p) \tag{Sp3}$$

Notice that the fluent Location only permits the description of translations of an object's shape with respect to the origin. It does not enable us to describe rotations. However, because (Sp3) is an implication rather than a biconditional, it would be straightforward to extend the language to include other fluents for describing rotations in combination with translations.

The basic vocabulary of a versatile language, which I will call L1, for describing the shape and spatial location of objects has now been presented. The features of L1 are summarised in Appendix A. The language is similar to that developed by Davis [1987]. But before we could use it to represent and reason about complicated shapes, we would first have to formalise the commonsense law about spatial occupancy which is the main concern of this paper. In the next section, a simple move action is formalised, whose precondition that an object's destination must be empty is an example of how the need for this law arises. In the succeeding sections, we will see how attempts to formalise this law interact with the existence of situations axiom required by Baker's approach to the frame problem.

2. Movement

A full-scale theory of commonsense would have to incorporate a theory of continuous motion. Our everyday world is full of it — the movement of people, cars, animals, clouds, and so forth. The formalisation of discrete motion is only useful when studying abstractions like the Blocks World. However, the Blocks World is valuable as a distillation of certain problems which arise in any formalisation of change, such as the frame problem and the qualification problem. When we begin to scale-up from the Blocks World, we encounter new problems. But these problems too can be most easily studied by looking at the smallest possible scaling up in which they are still manifest.

The language of the preceding section moves beyond the usual representations of the Blocks World by considering space as a real-valued co-ordinate system, rather than a finite number of locations.[3] The next step in a full-scale theory would be to formalise continuous motion. In [Shanahan, 1994], a variant of state-based minimisation is applied to the representation of continuous change. The work reported there, in combination with elements of the language L1, could form the basis of such a formalisation. But this is not the concern of the present paper. Here we are interested in certain issues in default reasoning which arise when we consider spatial occupancy even in the discrete case. This section formalises discrete motion in the context of the language of Section 1, and draws attention to this spatial occupancy problem with an example.

The familiar notation of the situation calculus will be adopted, and is assumed to be contained in the language L1. This includes a sort for actions, with variables a, a0, a1, a2, and so forth. The term Result(a,s) denotes the situation which results from performing action a in situation s. The effects of an action are described by a number of axioms of motion. The domain we will study comprises the single action Move(ob,p), which, if successful, moves the reference point of the object to point p. That is to say, if the action is successful, the object will occupy the region obtained by displacing the object's shape by d1 units horizontally and d2 units vertically, where p=⟨d1,d2⟩.

I will not consider the possibility of two Move actions taking place concurrently, although the work of Lin and Shoham [1992] would be helpful in this respect. The Move action has a single precondition. The action Move(ob,p) will be successful if and only if the region around p to which ob is to be moved is empty. In practice, a more complex precondition than this would be required, one which insisted on a clear path to the

[3] In fact, we do not have to move to a real-valued co-ordinate system for the issues under discussion in this paper to arise. Space does not even have to be dense, but simply has to comprise a very large number of possible locations.

object's destination. But my concern here is only to illustrate the need to minimise spatial occupancy. We have the following axiom of motion.

$$\text{Holds}(\text{Location}(ob,p),\text{Result}(\text{Move}(ob,p),s)) \leftarrow \text{Possible}(\text{Move}(ob,p),s) \qquad \text{(Do1)}$$

$$\begin{aligned}\text{Possible}(\text{Move}(ob1,p1),s) \leftrightarrow\ & \text{Shape}(ob1)=r1 \wedge r2=\text{Displace}(r1,p1) \wedge \\ & \neg \exists\, ob2, r3, p2\ [\text{Holds}(\text{Occupies}(ob2,r3),s) \wedge \\ & \quad ob1 \neq ob2 \wedge p2 \in r2 \wedge p2 \in r3] \end{aligned} \qquad \text{(Do2)}$$

Many techniques have been developed to address the frame problem, especially since Hanks and McDermott [1987] introduced the Yale shooting scenario as a benchmark (see [Lifschitz, 1987] and [Shoham, 1988], for example). In this paper, the following general purpose frame axiom (Ch1) will be used in combination with the circumscription policy devised by Baker [1991]. Baker's attempt to overcome the frame problem is one of the most successful, and is certainly the most appropriate here, since as well as dealing well with the Hanks-McDermott problem, it can cope with domain constraints (or actions with ramifications).

$$[\text{Holds}(f,s) \leftrightarrow \text{Holds}(f,\text{Result}(a,s))] \leftarrow \neg\, \text{Ab}(a,f,s) \qquad \text{(Ch1)}$$

In addition to the frame axiom (Ch1), Baker's approach to the frame problem employs an axiom which guarantees the existence of a situation for every possible combination of fluents in the domain. Axiom (St1) below fulfils exactly the same role as Baker's existence of situations axiom, but works in a slightly different way. Instead of employing Baker's generalised fluents (compound fluents formed with the functions And and Neg), I will only consider interpretations in which the domain of the situation sort is the power set of the set of all fluents. Obviously, any such interpretation will include a set for every possible combination of fluents, containing exactly those fluents. The only extra axiom that is required then is the following.

$$[\text{Holds}(f,s) \leftrightarrow f \in s] \leftarrow \neg\, \text{AbState}(s) \qquad \text{(St1)}$$

The $\in$ predicate is now being used for both sets of fluents and sets of points. Axiom (St1) is made into a default by the AbState condition, as in Baker's axiom. This makes it consistent for domain constraints to rule out certain combinations of fluents. The circumscription policy to overcome the frame problem, representing the commonsense law of inertia, is to minimise Ab and AbState, with the minimisation of AbState taking a higher priority than that of Ab, allowing the Result function to vary. Letting the Result function vary means that two models can still be compared although they interpret the Result function differently.

Now let's consider an example. Suppose that we are interested in a world of discs, each of radius five units. There are two discs in the initial situation D1 and D2, whose reference points are located respectively at $\langle 0,0 \rangle$ and $\langle 10,10 \rangle$. What is the result of moving D1 to $\langle 20,20 \rangle$? The initial situation, which is denoted by S0, is described by the following sentences.

$$\text{Shape}(D1) = \text{Disc}(5) \quad \text{(Ex4)}$$

$$\text{Shape}(D2) = \text{Disc}(5) \quad \text{(Ex5)}$$

$$\text{Holds}(\text{Location}(D1,\langle 0,0\rangle),S0) \quad \text{(Ex6)}$$

$$\text{Holds}(\text{Location}(D2,\langle 10,10\rangle),S0) \quad \text{(Ex7)}$$

Let S1=Result(Move(D1,⟨20,20⟩),S0). Intuitively, from the axioms given so far, we might expect to be able to prove Holds(Location(D1,⟨20,20⟩),S1). But it is easy to see that we cannot. In fact, we can't prove anything useful about D1's location in Result(Move(D1,⟨20,20⟩),S0), because we can neither prove nor disprove the precondition Possible(Move(D1,⟨20,20⟩),S0). And the reason we can neither prove nor disprove this precondition is that the sentences describing the initial condition (Ex4) to (Ex7) do not exclude the possibility that there are other objects besides D1 and D2, which could occupy regions overlapping with D1's destination. Some models will exist which include such extra objects, and others will exist which do not. In the first kind of model, D1 will stay put because the frame axiom (Ch1) will apply, and in the second kind D1 will move, because Axiom (Do1) will apply.

3. The Spatial Occupancy Problem

The obvious solution to the spatial occupancy problem described at the end of the last section is to state explicitly which regions of space are not occupied in S0. For the example here, this could be done with the following sentence, which says that the only objects occupying any space in S0 are D1 and D2.

$$\neg \exists ob,r\ [\text{Holds}(\text{Occupies}(ob,r),S0) \wedge ob{\neq}D1 \wedge ob{\neq}D2] \quad \text{(Ex8)}$$

From (Ex8) and (Sp1) it is straightforward to prove that the five unit circle of space around ⟨20,20⟩ is empty in S0. However, there are several reasons why we cannot expect always to be able to write a sentence like (Ex8). First, we have to worry about every situation, not just the initial one. But, as I will show later, an appropriate treatment of the problem for the initial situation may suffice for all situations.

The second reason is that it may not be a straightforward matter to work out what objects are present in the initial situation. In the example here, they are given explicitly, but this may not be the case. Suppose we have compound objects. Given that a compound object is present in the initial situation, we might want to be able to deduce that all its parts are also present. Conversely, given that all the parts of a compound object are present, we might want to be able to deduce that the whole object is there. Such examples are not considered in this paper, but whenever there is a complex logical relationship between the various objects that exist in the initial situation, it will in general not be possible to write a sentence like (Ex8) without first of all working out the logical consequences of the other sentences describing the initial situation.

A third reason is simply that any such sentence may turn out to be false. We don't really want to pretend that we know all the objects that are present in the initial situation when we don't, but rather to be able to assume by default that the objects we do know about are all there are. This admits the possibility that later information may lead us to reject our assumption. What we seek, in other words, is an "elaboration tolerant" solution,[4] one which does not demand the reconstruction of our knowledge when new information arrives, but instead is able to gracefully absorb revisions to our old assumptions.

Finally, facts about what space is empty seem so mundane that we feel that we shouldn't have to write them out explicitly. They are a matter of commonsense. In constructing a set of axioms to describe a situation, why should we be forced to concern ourselves with the obvious? In a domain of any complexity, the task will be hard enough as it is.

In short, we want to use some form of default reasoning to assume that space is unoccupied by default, in much the same way and for much the same reasons that we needed to employ default reasoning to overcome the frame problem by assuming that actions do not affect fluents by default.[5]

The first option is to use some form of domain closure to minimise the objects in the domain. However, not every object in the domain of discourse is necessarily present (in the sense of being spatially located) in every situation. In principle, objects can come into being and can cease to exist. For example, the outcome of cutting a loaf of bread into slices is that the loaf ceases to exist and the individual slices come into being (see [Davis, 1993]). Appendix B suggests how the language of this paper can be used to formalise such examples. Domain closure is too weak to be useful here, because it would not prohibit models in which the loaf reappears as a separate entity to block a later Move action.

The minimisation required must be relativised to each situation. This is why domain closure is too coarse to be effective. In effect, we need to pick a fluent to minimise in each situation. This can be done using circumscription in the usual way through an abnormality predicate. The question is which fluent to pick. The two possibilities are Occupies and Location.

If we choose Location, then we need to introduce an extra axiom which insists that if an object occupies a region, then it must also have a location. Otherwise models will be permitted in which a phantom object, which occupies space but has no location, blocks a Move action. That is to say, we could have an object A such that Holds(Occupies(A,r),s), for some r and s, but where there is no p such that Holds(Location(A,p),s). With the introduction of an axiom setting up a one-one mapping between Occupies and Location, it is easy to prove that minimising Location and minimising Occupies would be equivalent.

So let's choose to minimise Occupies. Note that this is not an attempt to select models with fewer objects in their domains. Rather, the circumscription, along with all the circumscriptions in the rest of the paper, is an attempt to prefer models in which fewer objects are located in space. The following axiom, plus a circumscription policy that minimises AbSpace, seems as if it should do the job.

$$\text{AbSpace(r,s)} \leftarrow \exists \text{ob Holds(Occupies(ob,r),s)} \qquad \text{(Oc1)}$$

---

[4] This term is due to McCarthy.

[5] I suspect that the spatial occupancy problem described here, like the frame problem, will arise with any formalism for representing change. There is no reason to suppose that it is a consequence of choosing the situation calculus.

However, in the presence of Axiom (St1), since a situation exists for every possible combination of fluents, minimising AbSpace in fact has no effect whatsoever. A circumscription policy which minimises AbSpace prefers models with situations in which there is less spatial occupancy. But every model includes all possible situations anyway, with every possible degree of spatial occupancy, so to prefer models with a certain kind of situation is simply to have no preference.

To see this, consider (Ex4) to (Ex7) again, and suppose that we also have (Oc1), and that we circumscribe minimising AbSpace. Axioms (Ex6) and (Ex7) tell us that in situation S0, there are two discs. There is nothing to say that they are the only discs, but the intention was that the circumscription should rule out models with extra objects. However, we must not forget Axiom (St1), which has the same effect as Baker's existence of situations axiom, and which is there to ensure that we can overcome the frame problem.

Axiom (St1) guarantees that every model includes a situation for every legitimate combination of fluents (illegitimate combinations being ruled out by domain constraints). This means that every model includes situations with one disc, with two discs, with three discs, and so on, in every possible location. In general, every model includes a situation for every possible arrangement of shapes and objects. The effect of Axioms (Ex4) to (Ex7) is simply to ensure that in each model, S0 denotes a situation in which there are two discs in the appropriate locations.

Now, the circumscription policy prefers models in which there is less spatial occupancy. In other words, it would prefer a model which included only situations with discs D1 and D2, to a model which included those situations plus some situations with an extra, phantom disc. But the first kind of model does not exist. Even though only a fraction of them play any role, every model has to include every possible situation. So the circumscription policy doesn't achieve anything at all.

One way out of this would seem to be to minimise AbSpace with a higher priority than AbState. Then the circumscription would indeed prefer models with situations with less spatial occupancy. Unfortunately though, this would be at the expense of our solution to the frame problem.

To see this, consider the same example again, but suppose we are minimising AbSpace, AbState and Ab prioritised in that order. Now models which include situations with phantom extra discs are ruled out, because they will never be minimal with respect to AbSpace, which has the highest priority in the circumscription. But because AbSpace has the highest priority, the circumscription now prefers models in which spontaneous acts of destruction take place. For example, in models in which a Move action destroys all discs apart from the one it is moving, the extension of Ab will be larger than in some other models, but this is set against a reduction in the extension of AbSpace, which takes priority.

4. Default Reasoning and the Existence of Situations

A better approach to the spatial occupancy problem than that offered by Axiom (Oc1) is to distinguish certain situations as the "important" ones, without eliminating the rest from any model, and to minimise spatial occupancy with respect to the important situations only. To see how this might work, consider the following axiom which substitutes for (Oc1).

$$\text{AbSpace}(r,s) \leftarrow \text{Important}(s) \wedge \exists ob\ \text{Holds}(\text{Occupies}(ob,r),s) \quad \text{(Oc2)}$$

Of course, if the same combinations of fluents were the important ones in every model, this manoeuvre wouldn't succeed. But it does succeed, because the name of a situation can be used to identify it as an important one, and the same names can refer to different combinations of fluents in different models. However, the result of applying this strategy isn't always obvious. For example, to identify as important only those situations which are the result of applying a sequence of actions to the initial situation, it seems that we should be able to use the following sentences.

$$\text{Important}(s1) \leftrightarrow \text{Initial}(s1) \vee [s1=\text{Result}(a,s2) \wedge \text{Important}(s2)] \quad \text{(Oc3)}$$

$$\text{Initial}(s1) \wedge \text{Initial}(s2) \rightarrow s1=s2 \quad \text{(In1)}$$

Axiom (In1) ensures that there is a unique initial situation. Axiom (Oc3) picks out a subset of the possible combinations of fluents, namely those that are accessible from the initial situation. But contrary to what we might expect, the minimisation of AbSpace in the presence of (Oc2), (Oc3) and (In1) has no effect on any situation except the initial one. Intuitively, this is because the minimisation of Ab to solve the frame problem already takes care of minimising spatial occupancy, so long as spatial occupancy is minimised in the initial situation. If there is no unnecessary occupation of space in the initial situation, then any solution to the frame problem should ensure that no objects are spontaneously created (or destroyed) by any action, since that would constitute an unnecessary abnormality. This leads us to a more elegant solution of the spatial occupancy problem, which is simply to minimise spatial occupancy in the initial situation, using the following axiom instead of (Oc3).

$$\text{AbSpace}(r) \leftarrow \exists ob\ \text{Holds}(\text{Occupies}(ob,r),S0) \quad \text{(Oc4)}$$

The circumscription policy is to minimise AbState, Ab, and AbSpace, prioritised in that order, allowing all other predicates and functions to vary. This is still just a partial solution, however, since it only works for problems in which the only given information concerns the initial situation. If we have information about the locations of objects in situations other than the initial one, the minimisation of spatial occupancy will go awry. A fuller solution is presented at the end of this section. But first, I want to take a step back, in order to clarify this approach of distinguishing certain situations from the rest, and restricting the minimisation of spatial occupancy to those alone. Otherwise, it may seem that the solutions I am proposing are an artefact of Baker's approach to the frame problem, and that in the context of a different approach to the frame problem, a solution to the spatial occupancy problem would be possible which didn't require certain situations to be distinguished from the rest.

The ontology of the situation calculus is free of any commitment to the actual occurrence of any particular sequence of actions and is neutral about whether any particular state of affairs actually comes about. Every model of a collection of sentences describing the effects of actions potentially includes every possible sequence of actions, and every possible state of affairs. In the context of such models, it makes no sense to speak of preferring those which have situations with less spatial occupancy, because every model potentially contains every possible situation, with every possible degree of spatial occupancy.

But the assumption behind the idea of minimising spatial occupancy is that we have incomplete knowledge about certain situations. A better way of putting it is to say that we don't know which combinations of fluents certain situation names refer to. Then, what minimising spatial occupancy really does is to prefer models in which those situation names refer to combinations of fluents with less spatial occupancy. The same collections of fluents appear in every model, but the particular collection of fluents denoted by a given name can vary. This is what we are really doing when we distinguish certain situations from the rest and confine minimisation to those situations. The situations we distinguish are those of which we can give only a partial description, which we do by referring to the situation by a name.

To make this clearer, let's consider a simple and slightly informal example which embodies some of the same problems we have encountered with the formal treatment of spatial occupancy. Suppose we have a language of sorted predicate calculus which includes sets and set membership. Let's consider only interpretations in which sets and set membership are interpreted in the usual way. Now suppose we have the following sentences.

$$\text{Special(S1)} \wedge \text{Special(S2)} \wedge \text{Special(S3)} \qquad \text{(Ex9)}$$

$$\text{B} \in \text{S1} \wedge \text{B} \in \text{S2} \wedge \text{B} \in \text{S3} \wedge \text{C} \in \text{S3} \qquad \text{(Ex10)}$$

What we want to do is minimise membership of the special sets S1, S2 and S3. Note that S1, S2 and S3 can denote the same object. The natural way to do this would seem to be to circumscribe these sentences along with the following sentence, minimising Ab and allowing Special, S1, S2, and S3 to vary.

$$\text{Ab(s,f)} \leftarrow \text{Special(s)} \wedge \text{f} \in \text{s} \qquad \text{(Ex11)}$$

But consider the two models illustrated in Figure 1. It can easily be verified that both M1 and M2 are models of the circumscription. Yet in M1, S2 denotes a proper subset of the set it denotes in M2. Otherwise, the models are identical. The circumscription has failed to capture what we wanted, which was to prefer interpretations in which the names S1, S2 and S3 each denote the smallest possible sets.

Now consider instead the two sets AB1 = $\{\langle t,f\rangle \mid$ M1 satisfies Ab(t,f)$\}$ and AB2 = $\{\langle t,f\rangle \mid$ M2 satisfies Ab(t,f)$\}$, where t is one of the terms S1, S2 or S3. From Figure 1, we can see that AB1 is $\{\langle S1,B\rangle,\langle S2,B\rangle,\langle S3,B\rangle,\langle S3,C\rangle\}$ whilst AB2 is $\{\langle S1,B\rangle,\langle S2,B\rangle,\langle S3,B\rangle,\langle S3,C\rangle,\langle S2,C\rangle\}$. In other words, AB1 $\subset$ AB2. So the subset relation between sets defined in this way seems to capture the preference relation between models we were seeking, where the obvious circumscription policy failed.

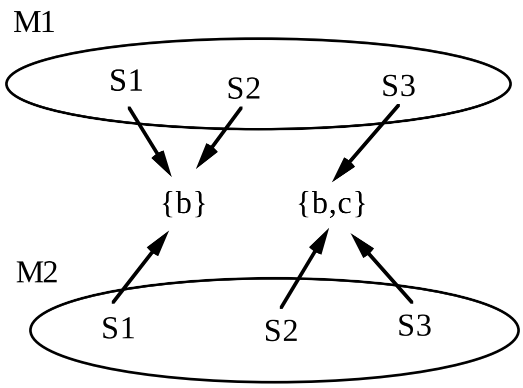

Figure 1.

I won't make any strong claims about this kind of preference relation here, except that it captures a certain sort of minimisation which is hard to describe using ordinary circumscription. The minimisation of spatial occupancy, like that of set membership here, is an example of this. As in this example, the minimisation we seek for spatial occupancy is relative to the names of things rather than the things those names denote.

Let's return to Axiom (Oc4). In principle, we could name and partially describe a whole narrative of situations and actions (see [Miller & Shanahan, 1994]). But the solution represented by (Oc4) fails where there is spatial occupancy information about situations other than the initial one. Axiom (Oc5) below represents a more complete solution to the spatial occupancy problem, but it also incorporates an innovation to the situation calculus, namely the idea of narrative time [Miller & Shanahan, 1994].[6] An extra sort for times is introduced, with variables t, t0, t1, t2, etc. For any given time t, there is a unique situation, corresponding to the set of fluents that hold in that situation, and denoted by the term State(t). The language we obtain by incorporating these innovations into L1 will be denoted L2 (see Appendix A).

$$\text{AbSpace}(r,t) \leftarrow \exists ob\ \text{Holds}(\text{Occupies}(ob,r),\text{State}(t)) \qquad \text{(Oc5)}$$

As before, we minimise AbState, Ab and AbSpace, prioritised in that order, allowing all other predicates and functions to vary. The effect of (Oc5) with this circumscription policy is to ensure that, for any time t, the situation denoted by State(t) has the least possible spatial occupancy. As with the solution using (Oc4), Baker's solution to the frame problem ensures that the result of any sequence of actions applied to a situation denoted by the State function also has the least possible spatial occupancy.

5. Separating the Spatial and Temporal Defaults

The following results demonstrate that the minimisation of spatial occupancy according to the policy described above does not interfere with the minimisation of the effects of actions. These results are important because they show that Baker's solution to the frame problem is still a solution, even though an extra kind of minimisation is being introduced. One of the lessons of the Hanks-McDermott problem [1987] was that circumscription can be sensitive to apparently small changes to the formula being

[6] A similar step is taken, for similar reasons, by Crawford and Etherington [1992].

circumscribed. Circumscription needs to be used in ways that can be shown to be more robust.

Let CIRC[ϕ ; P*] denote the circumscription of ϕ minimising P* and allowing all other predicates and functions to vary, where P* is a set of predicates. Let CIRC[ϕ ; P_1* > . . . > P_n*] denote the circumscription of ϕ minimising P_1* to P_n*, prioritised in that order, and allowing all other predicates and functions to vary, where P_1* to P_n* are sets of predicates. Let CIRC[ϕ ; P_1* > . . . > P_n* ; Q*] denote the circumscription of ϕ minimising P_1* to P_n*, prioritised in that order, and allowing Q* to vary, where P_1* to P_n* are sets of predicates and Q* is a set of predicates, functions and constants. The following theorem is due to Lifschitz [1985].

Theorem 1. If P_1* to P_n* are sets of predicates and Q* is a set of predicates, functions and constants, then CIRC[ϕ ; P_1* > . . . > P_n* ; Q*] is equivalent to

$$\text{CIRC}[\phi ; P_1^* > \ldots > P_{n-1}^* ; P_n^* \cup Q^*] \wedge \text{CIRC}[\phi ; P_n^* ; Q^*]$$

Let $\text{CIRC}_S[\phi]$ denote the circumscription of ϕ minimising AbState, Ab, and AbSpace, prioritised in that order, and allowing all other predicates, S0 and the Result function to vary.

Theorem 2. If ϕ is a formula of L1 which does not mention the predicate AbSpace, then $\text{CIRC}_S[\phi \wedge \text{(Oc4)}]$ is equivalent to CIRC[ϕ ; AbState > Ab] $\wedge$ CIRC[$\phi \wedge$ (Oc4) ; AbSpace]. In other words, for simple projection problems, the minimisation of AbSpace does not interfere with the commonsense law of inertia, and vice versa.

Proof. It follows from Theorem 1 that $\text{CIRC}_S[\phi \wedge \text{(Oc4)}]$ is equivalent to CIRC[$\phi \wedge$ (Oc4) ; AbState > Ab] $\wedge$ CIRC[$\phi \wedge$ (Oc4) ; AbSpace]. Consider the first of these conjuncts. Because of the form of (Oc4), and because it is the only predicate to mention AbSpace, CIRC[$\phi \wedge$ (Oc4) ; AbState > Ab] is equivalent to CIRC[ϕ ; AbState > Ab] $\wedge$ (Oc4). So we have $\text{CIRC}_S[\phi \wedge \text{(Oc4)}]$ is equivalent to CIRC[ϕ ; AbState > Ab] $\wedge$ (Oc4) $\wedge$ CIRC[$\phi \wedge$ (Oc4) ; AbSpace], from which the theorem follows directly.

Finally, we have a version of Theorem 2 for Axiom (Oc5).

Theorem 3. If ϕ is a formula of L2 which does not mention the predicate AbSpace, then $\text{CIRC}_S[\phi \wedge \text{(Oc5)}]$ is equivalent to CIRC[ϕ ; AbState > Ab] $\wedge$ CIRC[$\phi \wedge$ (Oc5) ; AbSpace].

Proof. The proof is the same, *mutatis mutandis*, as for Theorem 2.

Note that, following the methodological recommendations of [Lifschitz, 1991] and [Sandewall, 1993], Theorems 2 and 3 apply to a wide class of theories, namely those describable by the languages L1 and L2, respectively.

6. The Example Revisited

Now let's reconsider the example given at the end of Section 2. Let ϕ be the conjunction of (Sp1) to (Sp3) and (Do1) to (Do2) and (Ch1) and (St1) and (Ex4) to

(Ex7). We'll consider the use of (Oc4) first. Let S1=Result(Move(D1,⟨20,20⟩),S0). Does Holds(Location(D1,⟨20,20⟩),S1) now follow from $CIRC_S[\phi \wedge (Oc4)]$?

From (Ex4) to (Ex7) and (Sp3), we have,

$$\exists r1,r2\ [Holds(Occupies(D1,r1),S0) \wedge Holds(Occupies(D2,r2),S0)]$$

From Theorem 2, we have CIRC[$\phi \wedge$ (Oc4); AbSpace]. So, minimising spatial occupancy in S0, we have,

$$\neg \exists ob,r\ [Holds(Occupies(ob,r),S0) \wedge ob \neq D1 \wedge ob \neq D2]$$

from which it's easy to show,

$$Possible(Move(D1,\langle 20,20\rangle),S0)$$

Then Holds(Location(D1,⟨20,20⟩),S1) follows from CIRC[ϕ; AbState < Ab], which itself follows from Theorem 2.

Using Axiom (Oc5), the example has to be represented slightly differently. Instead of (Ex6) and (Ex7), we have,

$$Holds(Location(D1,\langle 0,0\rangle),State(0)) \qquad (Ex12)$$

$$Holds(Location(D2,\langle 10,10\rangle),State(0)) \qquad (Ex13)$$

Now let ϕ be the conjunction of (Sp1) to (Sp3) and (Do1) to (Do2) and (Ch1) and (St1) and (Ex4) to (Ex5) and (Ex12) to (Ex13). We want to show that Holds(Location(D1,⟨20,20⟩),S1) follows from $CIRC_S[\phi \wedge (Oc5)]$, where this time S1=Result(Move(D1,⟨20,20⟩),State(0)). The derivation is then the same, *mutatis mutandis*, as above.

Concluding Remarks

The need to formalise the default rule that space is normally empty has drawn attention to the need for care when combining defaults. In solving the frame problem, one approach is to employ an existence of situations axiom to ensure the correct formalisation of the default rule that actions normally leave fluents unchanged. However the presence of this axiom rules out some apparently intuitive ways of formalising the required default about spatial occupancy.

Does this mean that the existence of situations axiom is, in some sense, wrong? After all, if it is reasonable axiom, we shouldn't expect its presence to give rise to counter-intuitive results, whatever new defaults we add. The difficulty here arises from ambiguity about what exactly a situation is. We must be clear whether a situation is a hypothetical snapshot of the world (in which case there will be a unique situation for every time but not the other way around), or whether it is a snapshot of an actual narrative of events (in which case there will be a unique time for every situation but not the other way around).

McCarthy's original conception of the situation calculus is compatible with either interpretation.[7] But the existence of situations axiom seems to imply the first interpretation. On the other hand, the incomplete information about spatial occupancy which has been the focus of this paper is an example of incomplete information about situations along an actual narrative, which seems to imply the second interpretation. One way around this apparant impasse is to accept the first interpretation, and to relativise default reasoning about spatial occupancy, not to situations, but to the names of situations (the paper's first solution) or to times (the paper's second solution).

The same strategy could be used in the context of other sorts of incompleteness in the description of a narrative besides spatial occupancy. The more general problem of narratives in the situation calculus has recently been studied by a number of authors [Crawford & Etherington, 1992], [Pinto & Reiter, 1993], [Miller & Shanahan, 1994], and their techniques could contribute to a more refined solution to the spatial occupancy problem. This is the subject of ongoing work.

Acknowledgements

Thanks to Rob Miller, Mark Ryan and Mikhail Soutchanski. The author is supported by an EPSRC Advanced Research Fellowship.

References

[Allen, 1984] J.F.Allen, Towards a General Theory of Action and Time, *Artificial Intelligence*, vol 23 (1984), pages 123-154.

[Baker, 1991] A.B.Baker, Nonmonotonic Reasoning in the Framework of the Situation Calculus, *Artificial Intelligence*, vol 49 (1991), pages 5-23.

[Crawford & Etherington, 1992] J.M.Crawford and D.W.Etherington, Formalizing Reasoning about Change: A Qualitative Reasoning Approach, *Proceedings AAAI 92*, pages 577-583.

[Davis, 1987] E.Davis, A Framework for Qualitative Reasoning about Solid Objects, reprinted in D.S.Weld and J.de Kleer (editors), Readings in Qualitative Reasoning about Physical Systems, Morgan Kaufmann, 1990, pages 603-609.

[Davis, 1990] E.Davis, Representations of Commonsense Knowledge, Morgan Kaufmann (1990).

[Davis, 1993] E.Davis, The Kinematics of Cutting Solid Objects, *Annals of Mathematics and Artificial Intelligence*, vol 9 (1993), pages 253-305.

[Hanks & McDermott, 1987] S.Hanks and D.McDermott, Nonmonotonic Logic and Temporal Projection, *Artificial Intelligence*, vol 33 (1987), pages 379-412.

[Hayes, 1985] P.J.Hayes, Ontology for Liquids, in *Formal Theories of the Commonsense World*, ed Hobbs J.R. and Moore R.C., Ablex (1985), pages 71-107.

[Kaufman, 1991] S.G.Kaufman, A Formal Theory of Spatial Reasoning, *Proceedings 1991 Knowledge Representation Conference*, pages 347-356.

[Kowalski & Sergot, 1986] R.A.Kowalski and M.J.Sergot, A Logic-Based Calculus of Events, *New Generation Computing*, vol 4 (1986), pages 67-95.

[7] This was confirmed by McCarthy in conversation.

[Lifschitz, 1985] V.Lifschitz, Computing Circumscription, *Proceedings IJCAI 85*, pages 121-127.

[Lifschitz, 1987] V.Lifschitz, Formal Theories of Action, *Proceedings of the 1987 Workshop on the Frame Problem*, pages 35-57.

[Lifschitz, 1991] V.Lifschitz, Toward a Metatheory of Action, *Proceedings 1991 Knowledge Representation Conference*, pages 376-386.

[Lin & Shoham, 1992] F.Lin and Y.Shoham, Concurrent Actions in the Situation Calculus, *Proceedings AAAI 92*, pages 590-595.

[McCarthy & Hayes, 1969] J.McCarthy and P.J.Hayes, Some Philosophical Problems from the Standpoint of Artificial Intelligence, in *Machine Intelligence 4*, ed D.Michie and B.Meltzer, Edinburgh University Press (1969), pages 463-502.

[McDermott, 1982] D.McDermott, A Temporal Logic for Reasoning about Processes and Plans, *Cognitive Science*, vol 6 (1982), pages 101-155.

[Miller & Shanahan, 1994] R.S.Miller and M.P.Shanahan, Narratives in the Situation Calculus, *The Journal of Logic and Computation*, vol 4, no 5 (1994).

[Pinto & Reiter, 1993] J.Pinto and R.Reiter, Temporal Reasoning in Logic Programming: A Case for the Situation Calculus, *Proceedings ICLP 93*, pages 203-221.

[Randell, *et al.*, 1992] D.A.Randell, Z.Cui and A.G.Cohn, A Spatial Logic Based on Regions and Connection, *Proceedings 1992 Knowledge Representation Conference*, pages 165-176.

[Sandewall, 1993] E.Sandewall, The Range of Applicability of Nonmonotonic Logics for the Inertia Problem, *Proceedings IJCAI 93*, pages 738-743.

[Shanahan, 1994] M.P.Shanahan, A Circumscriptive Calculus of Events, to appear in *Artificial Intelligence* (1994).

[Shoham, 1985] Y.Shoham, Naïve Kinematics: One Aspect of Shape, *Proceedings IJCAI 85*, pages 436-442.

[Shoham, 1988] Y.Shoham, Reasoning About Change: Time and Change from the Standpoint of Artificial Intelligence, MIT Press (1988).

Appendix A: The Languages L1 and L2

The language L1 is a many-sorted subset of first-order predicate calculus with equality. It includes all the usual connectives and quantifiers, and well-formed formulae of L1 are constructed in the standard way, except that the functions, constants and predicates of L1 are constrained as follows. L1 contains variables for regions (r, r0, r1, r2, etc.), points (p, p0, p1, p2, etc.), distances (d, d0, d1, d2, etc.), objects (ob, ob0, ob1, ob2, etc.), fluents (f, f0, f1, f2, etc.), situations (s, s0, s1, s2 etc), and actions (a, a0, a1, a2, etc.). These are the only sorts in L1.

L1 includes the following functions: Shape from objects to regions, Occupies from objects and regions to fluents, Location from objects and points to fluents, Displace from regions and points to regions, and Result from actions and situations to situations. L1 also includes a function which maps pairs of reals x and y to points, written $\langle x,y \rangle$. L1 contains the situation constant S0, and constants for (a subset of) the reals written using the standard decimal notation. Besides the above, a well-formed formula of L1 may mention other functions and constants of any sort except situations.

L1 also includes the infix predicates ∈, <, ≤, >, and ≥, as well as the following predicates: Holds which takes as arguments a fluent and a situation, Possible which takes an action and a situation, AbState which takes a situation, Ab which takes an action, a fluent and a situation, and AbSpace which takes a region. These are the only predicates in L1.

The language L2 is simply L1 augmented with a sort for times (t, t0, t1, t2, etc.) and a function State from times to situations, but without the situation constant S0. In L2, the predicate AbSpace takes as arguments a region and a time.

Appendix B: Cutting an Object in Half

The following formulae (Ex14) to (Ex17) illustrate how the languages L1 and L2 might be used to formalise domains in which objects come into being and cease to exist. The action Cut(ob), when applied to a disc of radius d, destroys ob and replaces it by two objects Top(ob) and Bot(ob), whose shapes are slightly truncated semi-circles as shown in Figure 2. The conventional centre of both Top(ob) and Bot(ob) is the centre of the disc ob, although ⟨0,0⟩ is not contained in either shape.

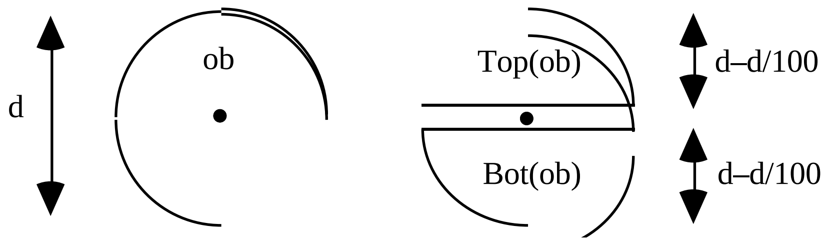

Figure 2

[Holds(Location(Top(ob),p), Result(Cut(ob),s)) ∧ (Ex14)
Holds(Location(Bot(ob),p), Result(Cut(ob),s))] ←
[Holds(Location(ob,p),s) ∧ Shape(ob) = Disc(d)]

¬∃p [Holds(Location(ob,p), Result(Cut(ob),s))] ← (Ex15)
Shape(ob) = Disc(d)

⟨x,y⟩ ∈ Shape(Top(ob)) ↔ (Ex16)
[Shape(ob) = Disc(d) ∧ Distance(⟨x,y⟩,⟨0,0⟩) ≤ d ∧
y ≥ d/100]

⟨x,y⟩ ∈ Shape(Bot(ob)) ↔ (Ex17)
[Shape(ob) = Disc(d) ∧ Distance(⟨x,y⟩,⟨0,0⟩) ≤ d ∧
y ≤ – d/100]

The possibility of domains like this precludes the use of simple domain closure to solve the spatial occupancy problem, as argued in Section 3, since different objects exist (in the sense of being spatially located) in different situations.