The Aspect Calculus

David Plaisted

July 1, 2019

The Aspect Calculus

David A. Plaisted

UNC Chapel Hill, Chapel Hill, NC 27599-3175
plaisted@cs.unc.edu

Abstract. For theorem proving applications, the aspect calculus for reasoning about states and actions has some advantages over existing situation calculus formalisms, and also provides an application domain and a source of problems for first-order theorem provers. The aspect calculus provides a representation for reasoning about states and actions that is suited to modular domains. An aspect names a portion of a state, that is, a substate, such as a room in a building or a city in a country. Aspects may have aspects of their own. A state is assumed to be either a *leaf state* that cannot be further decomposed, or to be composed of substates, and actions associated with one substate do not influence other, disjoint substates. This feature can reduce the number of frame axioms that are needed if the domain has a modular structure. It can also permit planning problems on independent substates to be solved independently to some degree. However, interactions between independent substates are also permitted.

Keywords: Situation calculus, frame problem, aspects, equational reasoning

1 Introduction

The situation calculus permits reasoning about properties of situations that result from a given situation by sequences of actions [MH69]. In the situation calculus, situations (states) are represented explicitly by variables, and actions a map states s to states $do(a, s)$. Predicates and functions on a situation or state are called *fluents*. In some formalisms, a situation denotes a state of the world, specifying the values of fluents, so that two situations are equal if the values of all their fluents are the same. Other formalisms reserve the term situation for a sequence of states. A problem with the situation calculus or any formalism for reasoning about actions is the necessity to include a large number of *frame axioms* that express the fact that actions do not influence many properties (fluents) of a state. Since the early days of artificial intelligence research the frame problem has been studied, beginning with McCarthy and Hayes [MH69]. Lin [Lin08] has written a recent survey of the situation calculus.

Reiter [Rei91] proposed an approach to the frame problem in first-order logic that avoids the need to specify all of the frame axioms. The method of Reiter, foreshadowed by Haas [Haa87], Pednault [Ped89], Schubert [Sch90] and Davis [Dav90], essentially solves the frame problem by specifying that a change in the truth value of a fluent, caused by an action, is equivalent to a certain condition on the action. In this formalism, it is only necessary to list the actions that change each fluent, and it is not necessary to specify the frame axioms directly. If an action does not satisfy the condition, the fluent is not affected. In the following discussion the term “Reiter’s formalism” will be used for simplicity even though others have also contributed to its development. The *fluent calculus* [Thi98] is another interesting approach to the frame problem. In this approach, a state is a conjunction of known facts

In the present paper, yet another approach to the frame problem using *aspects* is presented. This approach is based on the idea that the world is hierarchical or modular to a large extent. Aspects permit one to structure fluents and actions in a modular way.

The aspect formalism considers a situation, or state, to be composed of substates, These substates are named by aspects. Substates may have substates of their own. The aspect calculus constructs a tree of aspects. For example, the top node could be ”earth”, its children could be various countries, each country could have its states as children, and each state could have its cities as children. An aspect is a sequence of identifiers such as (earth, USA, North Carolina, Chapel Hill.) The aspect calculus is suitable if actions in a substate do not have much influence on fluents from a disjoint substate, roughly speaking. Thus the action of teaching a class in Chapel Hill would have aspect (earth, USA, North Carolina, Chapel Hill) and would only influence fluents that also had the same aspect, or an aspect referring to a part of Chapel Hill. This action would not have any effect on fluents with aspects of a different city, state, or country. However, calling someone in Washington DC from Chapel Hill would influence fluents in both cities and would have to be given an aspect of (earth, USA). Instead of sequences of names, the formal theory of aspects uses sequences of numbers.

Hayes actually mentioned “frames” which are very similar to aspects as a possible solution to the frame problem. He did not reject frames, but felt that they would not solve the frame problem in all cases. He wrote [Hay73], “In the long run I believe that a mixture of frame rules and consistency-based methods will be required for non-trivial problems ... ” (page 56).

Petrick [Pet08] has adapted Reiter’s formalism to knowledge and belief and has also introduced the notion of a Cartesian situation that can decompose a situation into parts, in a way that appears to be similar to the aspect calculus. However, his formalism also considers a situation to include a sequence of states.

The aspect calculus has some advantages over Reiter’s formalism, especially in its suitability for first-order theorem provers. In Reiter’s formalism, the successor state axiom for a fluent essentially says that the fluent is true on a situation $do(a, s)$ for fluent a and situation s if a is an action that makes the fluent true, or if the fluent was already true and a is not one of the actions that makes the fluent false. This requires one to know under what conditions an action changes the value of the fluent to ”true” or ”false.” If for example the action is nondeterministic this may be difficult to know. Also, to formulate the successor state axiom, one needs a theory of equality between actions. If there are only a small number of actions that can make a fluent false, then Reiter’s formalism is concise because one need not list all of the actions that do not influence the fluent (the frame axioms for the fluent). However, if there are many actions (possibly thousands or millions) that influence the fluent, then this successor state axiom can become very long. Further, when converting Reiter’s approach to clause form, one needs an axiom of the form ”For all actions a, $a = a_1 \vee a = a_2 \vee \cdots \vee a = a_n$” where a_i are all the possible actions, as well as the axioms $a_i \neq a_j$ for all $i \neq j$. If there are many actions, the first axiom will be huge. It is also difficult for many theorem provers to handle axioms of this form.

Even the successor state axiom itself, when translated into clause form, produces clauses having a disjunction of an equation and another literal. Using $\Phi(p, s)$ to denote the value of fluent p on situation s, a simple form of the successor state axiom would be

$$\Phi(p, do(x, s)) \equiv [(\Phi(p, s) \wedge (x \neq a_1) \wedge (x \neq a_2)) \vee (x = b_1 \vee x = b_2)]$$

where a_1 and a_2 are the only actions that can make p false and b_1 and b_2 are the only actions that make p true. Consider an even simpler form:

$$\Phi(p, do(x, s)) \equiv [(\Phi(p, s) \wedge (x \neq a_1)) \vee (x = b_1)]$$

The clause form of the latter is $\neg\Phi(p, do(x, s)) \vee \Phi(p, s) \vee x = b_1, \neg\Phi(p, do(x, s)) \vee x \neq a_1 \vee x = b_1, x \neq b_1 \vee \Phi(p, do(x, s)), \neg\Phi(p, s) \vee x = a_1 \vee \Phi(p, do(x, s))$. Such conjunctions of equations and inequations can be difficult for theorem provers to handle, especially if there are more actions in which case there would be more equations and inequations in the clauses.

The aspect calculus by contrast introduces many axioms that are unit equations, which are particularly easy for many theorem provers to handle. If the underlying domain is first-order then the aspect calculus is entirely expressed in first-order logic, so powerful first-order theorem provers can be applied to planning problems by framing a query of the form "There exists a situation having certain properties" and attempting to prove it. For this, a reflexive and transitive predicate *reachable* can be defined, the axioms $reachable(s, do(a, s))$ can be added for all actions a, and theorems of the form $(\exists s)(reachable(s_0, s) \wedge A[s])$ can be proved where s_0 is some starting state and A is a first-order formula. However, Reiter's formalism can handle domains without a clear hierarchical structure, especially if there are only a small number of actions that influence each fluent. Also, the aspect calculus does not handle knowledge and belief. Reiter's formalism attempts to make it easy to decide if a fluent is true on a situation obtained from a starting situation by a sequence of actions. The aspect calculus by contrast only attempts to preserve provability in the underlying theory while reducing the number of frame axioms.

The aspect calculus has other advantages independent of its suitability for theorem provers. Locality can be incorporated into the planning process. For example, if one wants to obtain a state t from s and the only difference is that a room in a building has changed, then one can first look for a plan that does not change anything outside the room. If that does not work, one can look for a plan that only changes rooms on that floor, changes to the other rooms being only temporary. If that does not work, one can look for a plan that only changes properties of the building, and nothing outside of it, and so on. Also, if the state space is finite, then the search space for planning problems in the aspect calculus is also finite. With Reiter's approach [Rei91], situations contain sequences of states, so the search space can be infinite. Planning in disjoint sub-states (aspects) of a state can be done independently to some extent. This reduces redundancies due to the order of actions involving independent modules not affecting the result.

Further, a possible problem with Reiter's approach, noted in Scherl and Levesque [SL93], is the *ramification problem*, namely, it can be difficult to incorporate *constraints* between fluents, such as when one fluent implies another. The successor state axiom essentially implies that the only way a fluent can become true is for an action to make it true. A great deal of work [Sha99,LR94,DT07,McI00,Ter00,MM97] has been done to handle the ramification problem in Reiter's system. No special treatment for the ramification problem is needed in the aspect calculus, but the theory needs to be hierarchical, that is, it should be possible to assign aspects so that disjoint aspects are largely independent.

2 Underlying Theory

We assume that there is some underlying set $\mathcal{T}$ of axioms in first-order logic concerning states, fluents, and actions. The semantics of this axiomatization will have domains for states and actions, with fluents mapping from states to various domains. We do not necessarily assume that $\mathcal{T}$ is encoded in any particular situation calculus, such as Reiter's [Rei91]. We will modify such a state theory $\mathcal{T}$ to obtain an axiomatization $\mathcal{T}^{aspect}$ that in some cases can more economically encode frame axioms than $\mathcal{T}$ does. In some cases $\mathcal{T}^{aspect}$ can be custom designed without transformation from a theory $\mathcal{T}$.

Actions in $\mathcal{T}$ are typically indicated by the letter a, possibly with subscripts, and fluents are typically indicated by the letters p and q, possibly with subscripts. $\mathcal{F}$ is the set of all fluents and $\mathcal{A}$ is the set of actions. States are denoted by s, t, and u, possibly with subscripts. The set of states is $\mathcal{S}$.

If a is an action and s is a state then $do(a, s)$ is the result of applying action a in state s. For nondeterminism, instead of $do(a, s) = t$ one would write $do(a, s, t)$ indicating that t is a possible result of applying action a in state s. It appears that the aspect formalism can handle this without a problem, but this has not been formally investigated. If p is a fluent then $\Phi(p, s)$ is the value of p on state s. Thus fluents are essentially functions from states to various domains. If the value of a fluent is $true$ or $false$, and it is not parameterized, then $\Phi(p, s)$ may be written as $p(s)$ instead. The semantics (interpretation) of the underlying theory $\mathcal{T}$ is assumed to have sorts for fluents, states, and actions, in addition to possibly others.

The semantics of operations is defined by assertions of the following form:

$$\lambda x_1 x_2 \ldots x_n . E[x_1, \ldots, x_n] : \psi_i \cdots \psi_n \to \psi_0$$

indicating that in the expression E, x_i are assumed to have sort ψ_i and E returns a value of sort ψ_0. One can then define the semantics of do and Φ as follows:

$\lambda as.do(a, s) : \mathcal{A} \times \mathcal{S} \to \mathcal{S}$

$\lambda ps.\Phi(p, s) : \mathcal{F} \times \mathcal{S} \to \mathcal{D}$ for some domain $\mathcal{D}$

We assume that $\mathcal{T}$ satisfies the action dependency condition if the fluents of $do(a, s)$ only depend on the fluents of s. This is formally defined as follows:

Definition 1 *The theory $\mathcal{T}$ satisfies the* action dependency condition *if* $\mathcal{T} \models (\forall s, t \in \mathcal{S})(\forall a \in \mathcal{A}), ((\forall p \in \mathcal{F})\Phi(p, s) = \Phi(p, t)) \to ((\forall p \in \mathcal{F})\Phi(p, do(a, s)) = \Phi(p, do(a, t)))$.

This constraint must be satisfied in order to use the aspect representation.

Example 1. We give an example state theory $\mathcal{L}_n$ in the "classical representation." For simplicity, fluents are written as $ron(i,s)$, $lon(i,s)$, $lonall(s)$, $ronall(s)$, and $onall(s)$ instead of $\Phi(ron(i),s)$, $\Phi(lon(i),s)$, $\Phi(lonall,s)$, $\Phi(ronall,s)$, and $\Phi(onall,s)$, respectively. In general, fluents are functions, but because these are all Booleans, we write $ron(i,s)$ instead of $ron(i,s) = true$, et cetera.

Suppose there are two banks of n switches that can be turned on and off and each switch controls a light. So there are actions $lton(i)$ (turn i on in the left bank) and $ltof(i)$ (turn i off in the left bank) for $1 \leq i \leq n$, also $rton(i)$ and $rtof(i)$ for the right bank. There are also fluents $lon(i,s)$ and $ron(i,s)$ telling whether the i-th light is on in the left and right banks. There is also a fluent $lonall(s)$ telling whether all the lights are on in the left bank, and similarly $ronall$ for the right bank, and $onall(s)$ for both banks being all on. A state is defined by whether the switches are on or off; all fluents other than $lon(i)$ and $ron(i)$ are functions of these. Thus there are 4^n states in all, one for each combined setting of the $2n$ switches. We can indicate a state in which $lon(i) = b_i$ and $ron(i) = c_i$ for Booleans b_i, c_i by $[b_1, \ldots, b_n, c_1, \ldots, c_n]_S$ where the subscript S may be omitted. The fluents $lonall, ronall$, and $onall$ can be determined from b_i and c_i and are not explicitly shown in this notation.

In the following equations for $\mathcal{L}_n$, the free variables s are states and are universally quantified. $(A5)^c$ through $(A8)^c$ are the frame axioms, and they make this representation quadratic in n.

$$lon(i, do(lton(i), s)) \wedge ron(i, do(rton(i), s)), 1 \leq i \leq n \text{ (A1)}^c$$
$$\neg lon(i, do(ltof(i), s)) \wedge \neg ron(i, do(rtof(i), s)), 1 \leq i \leq n \text{ (A2)}^c$$
$$lonall(s) \equiv lon(1, s) \wedge \cdots \wedge lon(n, s) \text{ (A3)}_l^c$$
$$ronall(s) \equiv ron(1, s) \wedge \cdots \wedge ron(n, s) \text{ (A3)}_r^c$$
$$onall(s) \equiv lonall(s) \wedge ronall(s) \text{ (A4)}^c$$
$$lon(i, do(lton(j), s)) \equiv lon(i, s), 1 \leq i, j \leq n, i \neq j \text{ (A5)}_l^c$$
$$ron(i, do(rton(j), s)) \equiv ron(i, s), 1 \leq i, j \leq n, i \neq j \text{ (A5)}_r^c$$
$$lon(i, do(rton(j), s)) \equiv lon(i, s), 1 \leq i, j \leq n \text{ (A6)}_l^c$$
$$ron(i, do(lton(j), s)) \equiv ron(i, s), 1 \leq i, j \leq n \text{ (A6)}_r^c$$
$$lon(i, do(ltof(j), s)) \equiv lon(i, s), 1 \leq i, j \leq n, i \neq j \text{ (A7)}_l^c$$
$$ron(i, do(rtof(j), s)) \equiv ron(i, s), 1 \leq i, j \leq n, i \neq j \text{ (A7)}_r^c$$
$$lon(i, do(rtof(j), s)) \equiv lon(i, s), 1 \leq i, j \leq n \text{ (A8)}_l^c$$
$$ron(i, do(ltof(j), s)) \equiv ron(i, s), 1 \leq i, j \leq n \text{ (A8)}_r^c$$
$$\exists s(s \in S)\text{(A9)}^c$$

3 Aspects

The theory $\mathcal{T}$ will be extended to a theory $\mathcal{T}^{aspect}$ that may permit many of the frame axioms of $\mathcal{T}$ to be omitted but will still permit the same plans to be derived. $\mathcal{T}^{aspect}$ is constructed so that any model M of the underlying theory $\mathcal{T}$ can be extended to a model M^{aspect} of $\mathcal{T}^{aspect}$. This implies the relative consistency of $\mathcal{T}^{aspect}$ with respect to $\mathcal{T}$, which essentially means that incorrect plans cannot be derived in $\mathcal{T}^{aspect}$. With notation as in the introduction, this means that $(\exists s)(reachable(s_0, s) \wedge A[s])$ is derivable in $\mathcal{T}^{aspect}$ iff it is derivable in $\mathcal{T}$, but $\mathcal{T}^{aspect}$ may have many fewer frame axioms. When presenting aspects the model M^{aspect} is essentially being described.

In $\mathcal{T}^{aspect}$ there are *aspects* and *statelets* in addition to the states, actions, and fluents of $\mathcal{T}$. Also, Ψ is the set of aspects and $\hat{\mathcal{S}}$ is the set of statelets.

The aspects are organized in $\mathcal{T}^{aspect}$ in an *aspect tree*. This can be regarded as part of the model M^{aspect}.

Definition 2 *The* aspect tree Υ *is a finite tree with a root node. Every other node in the tree is either a leaf with no children or else has finitely many children ordered from left to right. The nodes in the tree are labeled with sequences or strings of integers. The root is labeled with* ϵ*, the empty string. If a node* N *is labeled with* α *and has* n *children then its* n *children left to right are labeled* $\alpha 1$ *through* αn*. These sequences or strings of integers are called* aspects*. Aspects are indicated by Greek letters* α, β, γ*, possibly with subscripts. If node* N *with* n *children has aspect* α *then the aspects* $\alpha 1 \cdots \alpha n$ *are called the* children *of aspect* α*, and* α *is called the* parent *of* αi *for all* i*. Sometimes aspects can be written with commas between the numbers, as,* $1, 2, 1$ *or* $(1, 2, 1)$*. If node* L *has aspect* α *and node* N *has aspect* β *and* L *is an ancestor of* N *in the aspect tree, then we say that* α *is an* ancestor *of* β *and* β *is a* descendant *of* α*. Thus if* α *is a prefix of* β *then* α *is an* ancestor *of* β *and* β *is a* descendent *of* α*. If* α *is* $(3, 2, 4)$ *then* αi *is* $(3, 2, 4, i)$*.*

Definition 3 *There is an ordering relation* $<$ *on aspects with* $\alpha < \beta$ *if* α *is an ancestor (proper prefix) of* β*,* $\alpha < \beta$ *iff* $\beta > \alpha$ *and* $\geq, \leq$ *are defined as usual. Thus for example* $1, 2 > 1$*.*

Also, if two aspects α *and* β *are neither ancestors or descendants of one another, so that neither one is a prefix of the other, they are said to be* incomparable*,* independent*, or* disjoint*, written* $\alpha \# \beta$*.*

Actions and fluents have unique aspects assigned to them in $\mathcal{T}^{aspect}$. This assignment has to be done manually. If the theory has a natural hierarchical structure then this should be easier.

We write $a : \alpha$ to indicate that action a has aspect α, and $p : \alpha$ to indicate that a fluent p has aspect α; one can also write $aspect(p) = \alpha$ and $aspect(a) = \alpha$.

Example 2. Continuing with the example from Example 1, for $\mathcal{L}_3$ (three switches in the left and right banks) in $\mathcal{L}_3^{aspect}$ there would be aspects ϵ, (1), (2) and (i, j) for $i = 1, 2$ and $j = 1, 2, 3$. The aspect ϵ refers to the whole problem, (1) to the left bank of switches, (2) to the right bank, and (i, j) to switch j in the left or right bank.

4 Statelets

In addition to states, there is a set $\hat{\mathcal{S}}$ of *statelets* or *modules* in $\mathcal{T}^{aspect}$. Statelets can be indicated by the letters s, t, and u, possibly with subscripts. In $\mathcal{T}^{aspect}$, actions and fluents are extended from states to statelets. Thus

$\lambda as.do(a, s) : \mathcal{A} \times \hat{\mathcal{S}} \to \hat{\mathcal{S}} \cup \{\perp\}$ where $\perp$ is "don't care."

$\lambda ps.\Phi(p, s) : \mathcal{F} \times \hat{\mathcal{S}} \to \mathcal{D} \cup \{\perp\}$ for some domain $\mathcal{D}$ where $\perp$ is "don't care."

An assignment of aspects will be called *unconstraining* if it does not impose additional restrictions on $\mathcal{T}$, in a way that will be made precise later (Definition 16).

In general, statelets have unique aspects; writing $s : \alpha$ indicates that the aspect of statelet s is α. Equality for statelets $s : \alpha$ and $t : \beta$ is defined by their fluents and their aspect, as follows:

$$\text{If for all } p \text{ in } \mathcal{F}, \Phi(p, s) = \Phi(p, t) \text{ and } \alpha = \beta, \text{ then } s = t. \tag{1}$$

Thus statelets are entirely determined by how fluents map them, and by their aspect. This differs from states, which may have additional properties not used by our formalism. Also, there is a new value $\perp$ such that for all fluents p and all states s, $\Phi(p, s) \neq \perp$. Further, $\perp$ is not equal to any state or statelet. If s is a statelet and p is a fluent then $\Phi(p, s)$ can be $\perp$ (don't care).

Example 3. For $\mathcal{L}_3^{aspect}$, there would be 64 statelets at aspect ϵ, indicating the combined setting of all six switches. Therefore if statelet s has aspect ϵ then $lon(i, s)$ and $ron(i, s)$ would be $true$ or $false$ for all i. There would be eight statelets at aspect (1), specifying the combined setting of the three left switches, and similarly eight statelets at aspect (2). Also, there would be two statelets at aspects (i, j) for $i = 1, 2$ and $j = 1, 2, 3$, specifying the two possible settings of the corresponding switch.

Definition 4 *The $\perp$ values of a statelet are specified as follows: If a statelet s has aspect α then $(\forall p \in \mathcal{F})(\forall \beta \in \Psi)[(p : \beta) \to (\beta \geq \alpha \equiv (\Phi(p, s) \neq \perp))]$.*

Letting the aspect be part of the statelet eliminates some complexities from the system; one can then speak unambiguously about the parent and children of a statelet.

Statelets in $\mathcal{L}_n^{aspect}$ can be indicated by $[b_1, \ldots, b_n, c_1, \ldots, c_n]_{\hat{\mathcal{S}}}$ where the b_i and c_i can be Booleans or $\perp$ and the subscript $\hat{\mathcal{S}}$ may be omitted. Technically one should also indicate the aspect as well as the values of the fluents, but in this example the values of the fluents are enough to determine the aspect.

For $\mathcal{L}_3^{aspect}$, if $s : (1)$ (statelet s has aspect (1)) then $lon(i, s) = true$ or $false$ and $ron(i, s) = \perp$ for $i = 1, 2, 3$. Thus statelets at aspect (1) are of the form $[b_1, b_2, b_3, \perp, \perp, \perp]$ where the b_i are Booleans. If $s : (2)$ (statelet s has aspect (2)) then $ron(i, s) = true$ or $false$ and $lon(i, s) = \perp$ for $i = 1, 2, 3$. Thus statelets at aspect (2) are of the form $[\perp, \perp, \perp, c_1, c_2, c_3]$ where the c_i are Booleans. If $s : (1, i)$ then $lon(j) = \perp$ for $j \neq i$ and $ron(j) = \perp$ for $j = 1, 2, 3$ but $lon(i, s) = true$ or $false$. If $s : (2, i)$ then $ron(j) = \perp$ for $j \neq i$ and $lon(j) = \perp$ for $j = 1, 2, 3$ but $ron(i, s) = true$ or $false$. So a statelet at aspect $(1, 2)$ is of the form $[\perp, b_2, \perp, \perp, \perp, \perp]$ and a statelet at aspect $(2, 3)$ is of the form $[\perp, \perp, \perp, \perp, \perp, c_3]$.

Definition 5 *For states or statelets s and t, one writes $s \equiv_\alpha t$ if for all fluents p with $p : \beta$ and $\beta \geq \alpha$, $\Phi(p, s) = \Phi(p, t)$. Corresponding to this there is the assertion $s \equiv_{[\alpha]} t$ in $\mathcal{T}$ that does not mention aspects. This is defined as $\Phi(p_1, s) = \Phi(p_1, t) \wedge \Phi(p_2, s) = \Phi(p_2, t) \wedge \cdots \wedge \Phi(p_n, s) = \Phi(p_n, t)$ where $\{p_1, p_2, \cdots, p_n\}$ is the set of all fluents having aspects β in M^{aspect} with $\beta \geq \alpha$.*

Thus $s \equiv_\epsilon t$ if s, t agree on all fluents in $\mathcal{F}$. In $\mathcal{L}_3^{aspect}$, $[b_1, b_2, b_3, c_1, c_2, c_3] \equiv_{(1)} [b_1, b_2, b_3, c_1', c_2', c_3']$.

In M^{aspect}, states are related to statelets as follows:

Definition 6 *The* bridging axioms *are the following: If s is a state then s^ϵ is a statelet and $\Phi(p, s^\epsilon) = \Phi(p, s)$ for all $p \in \mathcal{F}$. Also, if a is an action and s, t are states and $do(a, s) = t$ then $do(a, s^\epsilon) = t^\epsilon$ Furthermore, for all statelets s at aspect α there is a state t such that $s \equiv_\alpha t$.*

In $\mathcal{L}_3^{aspect}$, $[b_1, b_2, b_3, c_1, c_2, c_3]_{\mathcal{S}}^\epsilon = [b_1, b_2, b_3, c_1, c_2, c_3]_{\hat{\mathcal{S}}}$.

There is also a function that restricts statelets at an aspect to statelets at another aspect. The function s^β with semantics $\lambda s\beta.s^\beta : \hat{\mathcal{S}} \times \Psi \rightarrow \hat{\mathcal{S}}$ defined as follows:

Definition 7 *If s is a statelet, $s : \alpha$, and $\beta \geq \alpha$ then s^β is defined as the statelet at aspect β such that $\Phi(p, s^\beta) = \Phi(p, s)$ if $aspect(p) \geq \beta$, else $\Phi(p, s^\beta) = \perp$. Thus $s^\beta : \beta$ and $s^\beta \equiv_\beta s$. If $\beta \leq \alpha$ or $\alpha \# \beta$ then s^β is not defined. If s is a state then $s^\alpha = (s^\epsilon)^\alpha$.*

Thus in $\mathcal{L}_3^{aspect}$, $[b_1, b_2, b_3, c_1, c_2, c_3]^{(1)} = [b_1, b_2, b_3, \perp, \perp, \perp]$ and $[b_1, b_2, b_3, c_1, c_2, c_3]^{(1,2)} = [\perp, b_2, \perp, \perp, \perp, \perp]$. Also $[b_1, b_2, b_3, \perp, \perp, \perp]^{(1,3)} = [\perp, \perp, b_3, \perp, \perp, \perp]$. In general, $\hat{\mathcal{S}}$ is $\{s^\alpha : s \in \mathcal{S}, \alpha \in \Psi\}$.

There is also a function f^α, the composition function, that combines statelets (submodules) at aspects $\alpha 1 \cdots \alpha n$ to produce a statelet (module) at aspect α. It has the semantics $\lambda \alpha s_1 s_2 \ldots s_n . f^\alpha(s_1, \ldots, s_n) : \Psi \times \hat{\mathcal{S}}^n \rightarrow \hat{\mathcal{S}}$ where n is the number of children of α.

Recall that αi is the sequence α with i added to the end.

Definition 8 *Suppose α is an aspect with n children (which are $\alpha 1, \ldots, \alpha n$). Suppose $s_1 : \alpha 1, \cdots, s_n : \alpha n$ for statelets s_i. Then $f^\alpha(s_1, s_2, \cdots, s_n) = s$ where s is a statelet at aspect α, and where for fluent p, if $p : \beta$ with $\beta \geq \alpha i$ then $\Phi(p, s) = \Phi(p, s_i)$. For fluents p with $p : \alpha$, $\Phi(p, s)$ is defined by the leaf dependency constraint, Definition 9, below. For other fluents p with $p : \beta$ for $\beta \not\geq \alpha$, $\Phi(p, s) = \perp$.*

In $\mathcal{L}_3^{aspect}$, $f^\epsilon([b_1, b_2, b_3, \perp, \perp, \perp], [\perp, \perp, \perp, c_1, c_2, c_3]) = [b_1, b_2, b_3, c_1, c_2, c_3]$. The first argument of f^ϵ is a statelet at aspect (1) and the second argument is a statelet at aspect (2). The value of f^ϵ is a statelet at aspect ϵ. Also, $f^{(1)}([b_1, \perp, \perp, \perp, \perp, \perp], [\perp, b_2, \perp, \perp, \perp, \perp], [\perp, \perp, b_3, \perp, \perp, \perp]) = [b_1, b_2, b_3, \perp, \perp, \perp]$. $f^{(1)}$ can also be written f^1.

From this definition it follows that $s \equiv_{\alpha i} s_i$ for all i and $s^{\alpha i} = s_i$. Definition 8 also implies the following *aspect composition equation* for non-leaf aspects α and statelets s at aspect α:

$$f^\alpha(s^{\alpha 1}, s^{\alpha 2}, \cdots, s^{\alpha n}) = s \tag{2}$$

In addition, there is a dependency constraint on fluents. That is, non-leaf fluents have to depend on fluents at the leaves of the aspect tree.

Definition 9 *The* leaf dependency constraint *on fluents is the following: If p is a fluent at non-leaf aspect α, s and t are statelets at aspect α, and $\Phi(q, s) = \Phi(q, t)$ for all fluents q at leaf aspects γ with $\gamma > \alpha$, then $\Phi(p, s) = \Phi(p, t)$.*

In terms of $\mathcal{T}$, this is expressed as a collection of assertions

$$\{A(p, q_1, \cdots, q_n) : (p : \alpha),\ \alpha \textit{ is a non-leaf aspect, and } \{q_1, \cdots, q_n\} \textit{ is the set of fluents at leaf aspects } \gamma \textit{ with } \gamma > \alpha\}$$

where $A(p, q_1, \cdots, q_n)$ is the following assertion:

$$\textit{For all states } s \textit{ and } t,$$
$$\Phi(q_1, s) = \Phi(q_1, t) \wedge \cdots \wedge \Phi(q_n, s) = \Phi(q_n, t) \rightarrow \Phi(p, s) = \Phi(p, t)$$

The leaf dependency constraint on fluents is necessary for f^α to be a mathematical function. This is the first constraint that must be satisfied when assigning aspects to actions and fluents. If one wants a statelet s to have properties that do not depend on the children aspects, then one can add a "virtual" child of the aspect of s that includes the extra information about s.

For $\mathcal{L}_3^{aspect}$, the leaf aspects are $(1, i)$ and $(2, i)$ for $i = 1, 2, 3$. The fluents at these aspects are $lon(i)$ and $ron(i)$, respectively. Thus the values of all other fluents have to be determined by these. For statelet s at aspect ϵ or (1), $\Phi(lonall, s)$ is determined by $\Phi(lon(i), s)$ for $i = 1, 2, 3$. Also, for $\mathcal{L}_3^{aspect}$ and statelet s at aspect ϵ or (2), $\Phi(ronall, s)$ is determined by $\Phi(ron(i), s)$ for $i = 1, 2, 3$ so this constraint is satisfied. Similarly, for statelet s at aspect ϵ, $\Phi(onall, s)$ is determined by the values $\Phi(lon(i), s)$ and $\Phi(ron(i), s)$ for $i = 1, 2, 3$.

Definition 10 *The* combining axiom *is the following: For all aspects α with n children and for all states $s_1, \cdots, s_n$ there is a state s such that*

$$s \equiv_{[\alpha 1]} s_1 \wedge \cdots \wedge s \equiv_{[\alpha n]} s_n.$$

The combining axiom is the second constraint that must be satisfied when assigning aspects to fluents and actions. This is satisfied for $\mathcal{L}_3^{aspect}$ because all combinations of all switch settings are permitted. This follows from $(A9)^c$ and the effects of the actions.

5 Actions

For action a at aspect α, $do(a, s)$ is defined for statelets s at aspect β iff $\beta \leq \alpha$. Otherwise, $do(a, s) = \perp$. Thus $do(a, s)$ is not always a statelet or a state, because it can be $\perp$. If $do(a, s) \neq \perp$ then $aspect(do(a, s)) = aspect(s)$.

There are some locality constraints on actions that need to be respected for $\mathcal{T}^{aspect}$ to be unconstraining. Taken collectively, these are the third constraint that must be satisfied by the assignment of aspects to fluents and actions.

Definition 11 *The* locality constraints *on actions are as follows: Suppose $a : \alpha$ and $p : \beta$. If $\alpha \# \beta$ then $\Phi(p, do(a, s)) = \Phi(p, s)$ for all states s. (Formally, this has to be a theorem of $\mathcal{T}$ for all such α and β). Also, if $s \equiv_\alpha t$ for states s and t (expressed in $\mathcal{T}$ by $s \equiv_{[\alpha]} t$) and $\beta \geq \alpha$ then $\Phi(p, do(a, s)) = \Phi(p, do(a, t))$.*

These constraints are satisfied for $\mathcal{L}_3^{aspect}$ because the action $lton(i)$ does not change the values of any fluents except $lon(i)$ at aspect $(1, i)$ and possibly $lonall$ and $onall$, but these are at aspects (1) and ϵ which are smaller than the aspect $(1, i)$ of $lton(i)$. Similar comments apply to $rton(i)$ and the fluents $ron(i)$, $ronall$, and $onall$.

5.1 Frame Axioms

Definition 12 *If for fluent* p *and action* a *and for some state* s, $\Phi(p,s) \neq \Phi(p, do(a,s))$ *then we say that action* a influences *fluent* p. *If for all states* s *and* t, $\Phi(p,s) = \Phi(p, do(a,s))$ *(if this is a theorem of* $\mathcal{T}$*) then* a does not influence p.

Frame axioms are encoded in the aspect system by the following *action locality* equation:

$$do(a, f^{\alpha}(s_1 \cdots s_n)) = f^{\alpha}(s_1 \cdots do(a, s_i) \cdots s_n) \tag{3}$$

for all a, α such that $aspect(a) \geq \alpha i$. Also, there is the *fluent locality* equation:

$$\Phi(p, f^{\alpha}(s_1 \cdots s_n)) = \Phi(p, s_i) \tag{4}$$

for all p, α such that $aspect(p) \geq \alpha i$.

These equations imply that if one has $p : \alpha$ and $a : \beta$ and α, β are incomparable then a does not influence p. This is how frame axioms are encoded in the aspect system. For $\mathcal{L}_3^{aspect}$, $do(lton(2), f^{(1)}(s_1, s_2, s_3)) = f^{(1)}(s_1, do(lton(2), s_2), s_3)$ because turning on left switch 2 does not influence left switches 1 or 3. Also, $\Phi(lon(2), f^{(1)}(s_1, s_2, s_3)) = \Phi(lon(2), s_2)$ because the fluent $lon(2)$ only depends on the setting of switch 2.

Definition 13 *Given* $\mathcal{T}$, *an aspect tree* Υ, *and an assignment* Π *of aspects to fluents and actions,* $\mathcal{M}_{\Upsilon,\Pi}$ *(or just* $\mathcal{M}$*) is the conjunction of Equation 1 for statelet equality, the bridging axioms, Definition 6, the aspect composition equation, Equation 2, and the locality axioms, Equations 3 and 4.*

Theorem 1. *From* $\mathcal{M}$ *it follows that if one has* $p : \alpha$ *and* $a : \beta$ *and* α, β *are incomparable then* $\Phi(p, do(a,s)) = \Phi(p,s)$ *for statelets* s *such that* $s : \gamma$ *where* γ *is the greatest lower bound of* α *and* β, *that is,* γ *is the largest aspect such that* $\gamma < \alpha$ *and* $\gamma < \beta$.

Proof. Since γ is the greatest lower bound of α and β, $\alpha > \gamma i$ for some i and $\beta > \gamma j$ for some $j \neq i$. Suppose γ has n children. Then $s = f^{\gamma}(s^{\gamma 1} \cdots s^{\gamma n})$ by Equation 2. Thus $\Phi(p, do(a,s)) = \Phi(p,s)$ is equivalent to $\Phi(p, do(a, f^{\gamma}(s^{\gamma 1} \cdots s^{\gamma n}))) = \Phi(p, f^{\gamma}(s^{\gamma 1} \cdots s^{\gamma n}))$. However, by Equation 4, $\Phi(p, f^{\gamma}(s^{\gamma 1} \cdots s^{\gamma n})) = \Phi(p, s^{\gamma i})$. Also, by Equation 3, $do(a, f^{\gamma}(s^{\gamma 1} \cdots s^{\gamma n})) = f^{\gamma}(s^{\gamma 1}, \cdots, do(a, f^{\gamma j}), \cdots, s^{\gamma n})$. Thus $\Phi(p, do(a, f^{\gamma}(s^{\gamma 1} \cdots s^{\gamma n}))) = \Phi(p, f^{\gamma}(s^{\gamma 1}, \cdots, do(a, f^{\gamma j}), \cdots, s^{\gamma n})) = \Phi(p, s^{\gamma i})$, again by Equation 4, so the equation $\Phi(p, do(a,s)) = \Phi(p,s)$ holds with both sides equal to $\Phi(p, s^{\gamma i})$. $\square$

Lemma 1. *Suppose* ξ *is an aspect and for some* i, $aspect(p) \geq \xi i$ *and* $aspect(a) \geq \xi i$. *Suppose* $s : \xi$ *and let* s *be* $f^{\xi}(s_1 \cdots s_n)$. *Then from* $\mathcal{M}$ *it follows that* $\Phi(p, do(a,s)) = \Phi(p,s)$ *implies* $\Phi(p, do(a, s_i)) = \Phi(p, s_i)$, *and the reverse implication also holds.*

Proof. Suppose $\Phi(p, do(a,s)) = \Phi(p,s)$. Then by Equation 3, $do(a,s) = f^{\alpha}(s_1 \cdots, do(a, s_i) \cdots s_n)$, so by Equation 4, $\Phi(p, do(a,s)) = \Phi(p, do(a, s_i))$, and by Equation 4 again, $\Phi(p,s) = \Phi(p, s_i)$. Therefore $\Phi(p, do(a, s_i)) = \Phi(p, do(a,s)) = \Phi(p,s) = \Phi(p, s_i)$ so $\Phi(p, do(a, s_i)) = \Phi(p, s_i)$. The reverse implication is shown in a similar way. $\square$

Theorem 2. *From $\mathcal{M}$ it follows that if one has $p : \alpha$ and $a : \beta$ and α, β are incomparable then $\Phi(p, do(a, s)) = \Phi(p, s)$ for statelets s with $s : \xi$ where $\xi \leq \alpha$ and $\xi \leq \beta$. Also, it follows that $\Phi(p, do(a, s)) = \Phi(p, s)$ for all states s.*

Proof. The first part follows by repeated application of Lemma 1. For the rest, letting ξ be ϵ, $\Phi(p, do(a, s)) = \Phi(p, s)$ for statelets s with $s : \epsilon$ and therefore by the bridging axioms, $\Phi(p, do(a, s)) = \Phi(p, s)$ for all states s. □

This result shows that in $\mathcal{T}^{aspect}$ one can omit any frame axioms involving fluents and actions at incomparable aspects.

6 Encoding a domain in the aspect formalism

A domain in the aspect calculus can be obtained in two ways: 1. By systematic translation from an existing domain. 2. By custom design. We first discuss the first possibility.

Definition 14 *Suppose one has an underlying state theory $\mathcal{T}$ with states, actions, and fluents and some axioms relating them. We want to encode $\mathcal{T}$ in the aspect formalism to obtain $\mathcal{T}^{aspect}$ that encodes as many of the frame axioms of $\mathcal{T}$ as possible in a more efficient manner, but does not imply frame axioms that are not theorems of $\mathcal{T}$. Suppose that an aspect tree Υ has been defined and aspects have been assigned for fluents and actions. Let $\mathcal{T}'$ be some theory such that $\mathcal{T}' \cup \mathcal{M}_{\Upsilon,\Pi}$ is equivalent to $\mathcal{T} \cup \mathcal{M}_{\Upsilon,\Pi}$. Typically $\mathcal{T}'$ can be $\mathcal{T}$ with frame axioms implied by $\mathcal{M}_{\Upsilon,\Pi}$ deleted. Then $\mathcal{T}^{aspect}_{\Upsilon,\Pi}$ is $\mathcal{T}' \cup \mathcal{M}_{\Upsilon,\Pi}$ for some such $\mathcal{T}'$.*

Thus there is some flexibility in defining $\mathcal{T}^{aspect}_{\Upsilon,\Pi}$. For concreteness, here is a more specific definition:

Definition 15 *Suppose one has an underlying theory $\mathcal{T}$ that can be expressed as $d_1 \wedge d_2 \wedge \cdots \wedge d_n$. Let $\mathcal{T}'$ be $e_1 \wedge e_2 \wedge \cdots \wedge e_m$ where $\{e_1, e_2, \cdots, e_m\} = \{d_i : 1 \leq i \leq n, \mathcal{M} \not\models d_i\}$. Then $\mathcal{T}^{aspect}_{\Upsilon,\Pi}$ is $\mathcal{T}' \cup \mathcal{M}$.*

Here is an example of such an underlying theory, in this case in first-order logic:

7 Switches Example

Let $\mathcal{T}$, that is, $\mathcal{L}_n$, be the theory from Example 1. We construct the theory $\mathcal{L}^{aspect}_n$ in the aspect representation.

7.1 Aspect Representation

The aspect tree Υ has a root node with two children, child 1 for the left bank and child 2 for the right bank. Each child has in turn n children numbered 1 through n, one for each switch. So the aspects are $\epsilon, (1), (2), (1,1), (1,2), \cdots, (1,n)$, $(2,1), (2,2), \cdots$, $(2,n)$. The actions $lton(i)$ and $ltof(i)$ have aspects $(1,i)$, and $rton(i)$ and $rtof(i)$

have aspects $(2, i)$. Also, $lon(i)$ has aspect $(1, i)$ and $ron(i)$ has aspect $(2, i)$. The fluent $lonall$ has aspect 1, $ronall$ has aspect 2, and $onall$ has aspect ϵ.

This is $\mathcal{T}'$, that is, $\mathcal{L}_n^{aspect}$; frame axioms are not needed and are omitted. Also, free occurrences of s refer to universally quantified states as before.

$$lon(i, do(lton(i), s)) \wedge ron(i, do(rton(i), s)), 1 \leq i \leq n \text{ (A1)}^c$$
$$\neg lon(i, do(ltof(i), s)) \wedge \neg ron(i, do(rtof(i), s)), 1 \leq i \leq n \text{ (A2)}^c$$
$$lonall(s) \equiv lon(1, s) \wedge \cdots \wedge lon(n, s) \text{ (A3)}_l^c$$
$$ronall(s) \equiv ron(1, s) \wedge \cdots \wedge ron(n, s) \text{ (A3)}_r^c$$
$$(onall(s) \equiv lonall(s) \wedge ronall(s)) \text{ (A4)}^c$$
$$(\exists s)(s \in S)\text{(A9)}^c$$

Here is $\mathcal{M}_\Upsilon$, consisting of the necessary portion (fluents at leaf aspects) of the bridging axioms, Definition 6, the locality axioms, Equations 3 and 4, and the aspect composition equation, Equation 2.

$$(lon(i, s) = lon(i, s^\epsilon)) \wedge (ron(i, s) = ron(i, s^\epsilon)), 1 \leq i \leq n$$
$$do(lton(i), s)^\epsilon = do(lton(i), s^\epsilon), 1 \leq i \leq n$$
$$do(rton(i), s)^\epsilon = do(rton(i), s^\epsilon), 1 \leq i \leq n$$
$$do(ltof(i), s)^\epsilon = do(ltof(i), s^\epsilon), 1 \leq i \leq n$$
$$do(rtof(i), s)^\epsilon = do(rtof(i), s^\epsilon), 1 \leq i \leq n$$
$$\forall t \in \hat{\mathcal{S}}(t : \alpha \rightarrow \exists s \in \mathcal{S}(s^\alpha = t))$$

In the following lines, the t_i refer to statelets.

$$do(lton(i), f^\epsilon(t_1, t_2)) = f^\epsilon(do(lton(i), t_1), t_2)$$
$$do(rton(i), f^\epsilon(t_1, t_2)) = f^\epsilon(t_1, do(rton(i), t_2))$$
$$do(ltof(i), f^\epsilon(t_1, t_2)) = f^\epsilon(do(ltof(i), t_1), t_2)$$
$$do(rtof(i), f^\epsilon(t_1, t_2)) = f^\epsilon(t_1, do(rtof(i), t_2))$$
$$do(lton(i), f^1(t_1, \ldots, t_n)) = f^1(t_1, \ldots, do(lton(i), t_i), \ldots, t_n)$$
$$do(rton(i), f^2(t_1, \ldots, t_n)) = f^2(t_1, \ldots, do(rton(i), t_i), \ldots, t_n)$$
$$do(ltof(i), f^1(t_1, \ldots, t_n)) = f^1(t_1, \ldots, do(ltof(i), t_i), \ldots, t_n)$$
$$do(rtof(i), f^2(t_1, \ldots, t_n)) = f^2(t_1, \ldots, do(rtof(i), t_i), \ldots, t_n)$$

$$lon(i, f^\epsilon(t_1, t_2)) = lon(i, t_1)$$
$$ron(i, f^\epsilon(t_1, t_2)) = ron(i, t_2)$$
$$lonall(f^\epsilon(t_1, t_2)) = lonall(t_1)$$
$$ronall(f^\epsilon(t_1, t_2)) = ronall(t_2)$$
$$lon(i, f^1(t_1, \ldots, t_n)) = lon(i, t_i), 1 \leq i \leq n$$
$$ron(i, f^2(t_1, \ldots, t_n)) = ron(i, t_i), 1 \leq i \leq n$$

In the following lines, t refers to a statelet and superscripts refer to the composition function of Definition 8.

$$t : \epsilon \rightarrow f^{\epsilon}(t^1, t^2) = t$$
$$t : 1 \rightarrow f^1(t^{1,1}, \ldots, t^{1,n}) = t$$
$$t : 2 \rightarrow f^2(t^{2,1}, \ldots, t^{2,n}) = t$$

Equation 1 for statelet equality also is included in $\mathcal{M}$. The $onall$, $lonall$, and $ronall$ predicates are allowed in the aspect representation because they are determined by fluents at their descendant leaves in the aspect tree according to the leaf dependency constraint of Definition 9.

A close examination shows that most of $\mathcal{L}_n^{aspect}$ is of complexity (size) linear in n. However, the number of axioms involving f^{α} for various aspects α is bounded by the depth of the aspect tree times the number of fluents and actions. Assuming the depth of the aspect tree is small compared to n, the complexity of $\mathcal{L}_n^{aspect}$ will be small relative to the classical version. Many of the lines in $\mathcal{L}_n^{aspect}$ mention the n variables t_i and this gives another quadratic factor, but the constant factor is at least smaller than the quadratic factor for the classical theory. However, even this factor can be reduced; the idea is to make the aspect tree a binary tree. This increases the number of equations while keeping the total number linear, but each equation will have a constant size. Also, the depth of the aspect tree will be at most logarithmic in n assuming the aspect tree is an approximately balanced binary tree. Many of the axioms are equations, which tend to be easy for first-order provers to handle, so the new equations should not make planning harder than for the classical representation.

7.2 Transmitting Switch Settings

Without going into detail, the switches example can be modified by also having an action $ltr(i)$ that transmits the state of left switch i to left switch $i + 1$, $1 \leq i < n$ and similarly rtr for right switches. Then $ltr(i)$ could have aspect 1 but not $(1, i)$ or $(1, i + 1)$ and $rtr(i)$ could have aspect 2 but not $(2, i)$ or $(2, i + 1)$, even though the switches they modify have aspects that are children of 1 and 2, respectively. The axiom for $ltr(2)$ in $\mathcal{T}$, for example, could be

$$lon(3, do(ltr(2), s)) \equiv lon(2, s).$$

This example shows how sub-modules (incomparable aspects) are not completely independent but can influence one another. Now, the aspect representation would automatically give frame axioms implying that ltr does not modify right switches and rtr does not modify left switches. However, it would not, for example, give the frame axiom that $ltr(2)$ does not modify left switch 1, because the aspects of left switch 1 and the action are not incomparable. Such frame axioms would be included in $\mathcal{T}'$. On this example, a custom translation can give a more succinct representation of these frame axioms. In particular, one can axiomatize ltr and rtr in $\mathcal{T}$ as follows, where $1 \leq i < n$, the statelets s_j with $s_j : (1, j)$ are universally quantified in the first two equations, and $s_j : (2, j)$ in the second two equations. Also, $ltr'(i, s_i)$ returns a statelet at aspect $(1, i+1)$ and $rtr'(i, s_i)$ returns a statelet at aspect $(2, i+1)$.

$$lon(i+1, ltr'(i, s_i)) = lon(i, s_i)$$
$$do(ltr(i), f^1(s_1, \cdots, s_n)) = f^1(s_1, \cdots, s_i, ltr'(i, s_i), s_{i+2}, \cdots, s_n)$$
$$ron(i+1, rtr'(i, s_i)) = ron(i, s_i)$$
$$do(rtr(i), f^2(s_1, \cdots, s_n)) = f^2(s_1, \cdots, s_i, rtr'(i, s_i), s_{i+2}, \cdots, s_n)$$

The binary tree idea can further reduce the complexity, as before.

This example and many similar examples involving transmitting information between disjoint aspects can be handled in a more systematic way by allowing some actions to have a set of aspects instead of just a single aspecct. The $ltr(i)$ action would have aspects $(1, i)$ and $(1, i+1)$. Without a fully rigorous treatment, the idea is to modify Equation 3 as follows for actions a with more than one aspect and more than one i such that αi is a prefix of at least one of the aspects of action a:

$$do(a, f^\alpha(s_1, \ldots, s_n)) = f^\alpha(s'_1, \ldots, s'_n)$$

where $s'_i = s_i$ if the aspect αi is disjoint from all the aspects of action a. Otherwise, s'_i is a statelet defined by axioms such as the first and third equations above. However, if there is only one i such that αi is a prefix of at least one of the aspects of a, then the original form of Equation 3 can be used. Thus for example $do(ltr(i), f^\epsilon(s_1, s_2)) = f^\epsilon(do(ltr(i), s_1), s_2)$.

8 The Unconstraining Property

Definition 16 *An assignment of aspects to actions and fluents in a state theory $\mathcal{T}$ that satisfies the action dependency condition, Definition 1, is* unconstraining *if it satisfies the locality constraints on actions of Definition 11, the leaf dependency constraint on fluents of Definition 9, and the combining axiom of Definition 10; that is, these must be theorems of $\mathcal{T}$.*

Theorem 3. *For any state theory $\mathcal{T}$ satisfying the action dependency condition and any aspect tree Υ, it is possible to find an assignment of aspects to fluents and actions that is unconstraining.*

Proof. The leaf dependency constraint on fluents can be satisfied by putting all fluents at the same leaf of Υ if necessary and the locality constraints on actions can be satisfied by assigning all actions the aspect of ϵ at the root of the tree. However, such an assignment of aspects would not encode any frame axioms, so it would not serve any purpose. □

8.1 Relative Consistency

We now show that if $\mathcal{M}_{\Upsilon,\Pi}$ is unconstraining and $\mathcal{T}$ satisfies the action dependency condition then $\mathcal{T}^{aspect}$ is relatively consistent with $\mathcal{T}$. This implies that $\mathcal{T}^{aspect}$ does not imply any new theorems on the assertions over the symbols in $\mathcal{T}$.

Theorem 4. *Suppose $\mathcal{T}$ is a theory of states, actions, and fluents that satisfies the action dependency condition, Definition 1. Suppose that an aspect tree Υ is chosen and aspects are assigned to fluents and actions of $\mathcal{T}$ in an unconstraining manner (Definition 16). Then $\mathcal{T}^{aspect}$ is relatively consistent with $\mathcal{T}$.*

Proof. We show that any model M of $\mathcal{T}$ can be extended to a model M^{aspect} of $\mathcal{T}^{aspect}$. M^{aspect} interprets the symbols of $\mathcal{T}$ on the domains of $\mathcal{T}$ the same way that M does. M^{aspect} has additional domains, the set of statelets and the set of aspects, and an additional element $\perp$ that can be the value of fluents and of $do(a, s)$ in M^{aspect}. Also, M^{aspect} has the functions f^α for aspects α in Υ mapping from tuples of statelets to statelets, and the function mapping states and statelets s to statelet s^α, for aspects α.

For every state s of M, there is a statelet s^ϵ of M^{aspect} such that for all fluents $p \in \mathcal{F}$, $\Phi(p, s^\epsilon) = \Phi(p, s)$. Two statelets that have the same aspect and the same value on all fluents in $\mathcal{F}$ are equal in M^{aspect}; other statelets are not equal in M^{aspect}.

The functions s^α from states or statelets s to statelets are defined as in Definition 7. The functions f^α are defined by $f^\alpha(s^{\alpha 1} \cdots s^{\alpha n}) = s^\alpha$ where the aspect α has n children. In M^{aspect}, the set $\hat{\mathcal{S}}$ of statelets is $\{s^\alpha : s \in \mathcal{S}, \alpha \in \Psi\}$.

Fluents of M are extended from states of M to statelets in M^{aspect}. The value $\perp$ is allowed as a value of $\Phi(p, s)$ for fluents p and statelets s, where $\perp$ is not equal to any state or statelet and $\Phi(p, s) \neq \perp$ for fluents p and states s. A fluent p of M that is assigned an aspect of α in M^{aspect} is defined on all statelets s^β for $\beta \leq \alpha$ and $\Phi(p, s^\beta) = \Phi(p, s^\epsilon)$ for all such β. If a statelet s has aspect γ and $\gamma \not\leq \alpha$ then $\Phi(p, s^\gamma) = \perp$.

Actions a of M are extended from states to statelets in M^{aspect}. Actions a of M with aspect α satisfy $do(a, s^\beta) = \perp$ for statelets s^β with $\beta \not\leq \alpha$. Actions a of M with aspect α are defined on all statelets s^β for $\beta \leq \alpha$. The value of $do(a, s^\beta)$ in this case is a statelet t with $t : \beta$ such that $t \equiv_\alpha do(a, s^\epsilon)$. This completely defines t because fluents q with aspects γ with $\alpha \# \gamma$ satisfy $\Phi(q, t) = \perp$ in M^{aspect}, and fluents with aspects γ with $\gamma \leq \alpha$ are specified by the leaf dependency constraint on fluents. This completely defines M^{aspect}.

It remains to show that M^{aspect} is a model of $\mathcal{T}^{aspect}$. Now, M^{aspect} is a model of $\mathcal{T}$ because it agrees with M there. So it remains to show that M^{aspect} is a model of $\mathcal{M}_\Upsilon$. Recall from Definition 13 that $\mathcal{M}_\Upsilon$ is the conjunction of Equation 1 for statelet equality, the bridging axioms, Definition 6, the aspect composition equation, Equation 2, and the action and fluent locality axioms, Equations 3 and 4.

The issue is that one can have $s^\alpha = t^\alpha$ even for unequal states s and t, so one has to show that all the functions and properties depend only on the fluents of s^α and not directly on s.

The proof is routine, so the details are omitted. □

9 Solving planning problems bottom up

For a binary relation R, $R(x, y)$ indicates that $(x, y) \in R$. If A is a logical assertion then $\{x : A\}$ is the set of x having property A. If α is an aspect then $x : \alpha$ iindicates that x has aspect α. Let M be a model of $\mathcal{T}^{aspect}$ and let the relations R^α, R_1^α, and R_2^α

be defined as follows, where a superscript of $*$ indicates transitive closure and I^α is the identity relation on statelets at aspect α:

$$R^\alpha = (R_1^\alpha \cup R_2^\alpha)^* \cup I^\alpha \tag{5}$$

$$R_1^\alpha = \{(f^\alpha(s_1 \cdots s_n), f^\alpha(t_1 \cdots t_n)) : R^{\alpha i}(s_i, t_i), 1 \leq i \leq n\} \tag{6}$$

$$R_2^\alpha = \{(s,t) : M \models do(a,s) = t, a \in \mathcal{A}, s : \alpha, aspect(a) \geq \alpha\} \tag{7}$$

R^α gives the set of pairs (s,t) of statelets at aspect α such that t is reachable in M from s by a finite sequence of actions at aspect α or larger aspects. Computing R^α can be helpful for solving planning problems by exhaustive search, and it avoids repetitive search due to actions on independent (incomparable) aspects commuting. Of course, if the number of states is finite, R^α will always be finite. This differs from Reiter's formalism [Rei91], in which the number of situations can be infinite even if the number of states is finite because situations are defined by sequences of states.

Theorem 5. *With R^α defined as in Equations 5,6, and 7, $R^\alpha(s,t)$ for statelets s and t with $s : \alpha$ and $t : \alpha$ iff there is a sequence $s_1 : \alpha, s_2 : \alpha, \ldots, s_n : \alpha$ of statelets where $s = s_1$, $t = s_n$, and for all i, $1 \leq i < n$, there is an action a_i with $a_i : \beta_i$ such that $\beta_i \geq \alpha$ and $M \models s_{i+1} = do(a_i, s_i)$.*

The proof is omitted for lack of space. Of course, this implies that $R^\epsilon(s^\epsilon, t^\epsilon)$ for states s and t iff t can be obtained in M from s by a finite sequence of actions.

10 Conclusion

The aspect calculus expresses frame axioms involving incomparable aspects of a state efficiently for modular theories. Aspects are sequences of integers that correspond to substates of a state; for example, in the sequence (i,j,k), i may indicate the earth, j a country, and k a state in a country. These sequences are ordered so that sequences are larger than (greater than) their proper prefixes. Fluents are assigned aspects based on which portion of the state they describe, so a fluent may have an aspect corresponding to North Carolina if it describes something about North Carolina. Then actions can be assigned aspects that are the greatest lower bound (longest common prefix) of the aspects of all fluents that they influence or depend on. This implies for example that actions in North Carolina do not influence fluents from outside North Carolina, thereby encoding many frame axioms.

This formalism is entirely in first-order logic, and powerful first-order theorem provers can be applied to it if the underlying theory $\mathcal{T}$ is first-order. When converted to clause form, the resulting clauses appear to be easier for first-order theorem provers to handle than clauses from Reiter's formalism. Also, for some theories, clauses from Reiter's formalism can become very long. Two examples are given and relative consistency is shown assuming that the unconstraining property holds. This formalism also permits an exhaustive method of solving planning problems that has some advantages for modular domains. The ramification problem does not require any special methods in the aspect calculus. However, this formalism does not handle knowledge and belief, but is only concerned with logical correctness. It is also only suitable for modular theories.

References

[Dav90] E. Davis. *Representations of Commonsense Knowledge*. Morgan Kaufmann, 1990.

[DT07] Marc Denecker and Eugenia Ternovska. Inductive situation calculus. *Artif. Intell.*, 171(5-6):332–360, 2007.

[Haa87] A. R. Haas. The case for domain-specic frame axioms. In F. M. Brown, editor, *The Frame Problem in artificial intelligence. Proceedings of the 1987 workshop*, pages 343–348. Morgan Kaufmann, 1987.

[Hay73] Pat Hayes. The frame problem and related problems in artificial intelligence. In A. Elithorn and D. Jones, editors, *Artificial and Human Thinking*, pages 45–59. Jossey-Bass, Inc. and Elsevier Scientific Publishing Company, 1973.

[Lin08] Fangzhen Lin. Situation calculus. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, *Handbook of Knowledge Representation*, pages 649–669. Elsevier, 2008.

[LR94] Fangzhen Lin and Raymond Reiter. State constraints revisited. *Journal of Logic and Computation*, 4(5):655–678, 1994.

[McI00] Sheila A. McIlraith. Integrating actions and state constraints: A closed-form solution to the ramification problem (sometimes). *Artificial Intelligence*, 116(1):87 – 121, 2000.

[MH69] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[MM97] Pedro A. Matos and João P. Martins. Contextual logic of change and the ramification problem. In Ernesto Coasta and Amilcar Cardoso, editors, *Progress in Artificial Intelligence*, pages 267–278, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[Ped89] E. P. D. Pednault. ADL: Exploring the middle ground between STRIPS and the situation calculus. In *Proceedings of the International Conference on Principles of Knowledge Representation (KR-98)*, pages 324–332. Morgan Kaufmann, Inc., 1989.

[Pet08] Ronald P. A. Petrick. Cartesian situations and knowledge decomposition in the situation calculus. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, pages 629–639, 2008.

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

[Sch90] Lehnart Schubert. Monotonic solution of the frame problem in the situation calculus: An efficient method for worlds with fully specified actions. In Henry E. Kyburg, Ronald P. Loui, and Greg N. Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, volume Volume 5, pages 23–67. Kluwer Academic Publishers, Dordrecht / Boston / London, 1990.

[Sha99] Murray Shanahan. The ramification problem in the event calculus. In *Proceedings of the 16th International Joint Conference on Artifical Intelligence - Volume 1*, IJCAI'99, pages 140–146, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[SL93] Richard B. Scherl and Hector J. Levesque. The frame problem and knowledge-producing actions. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pages 689–697, Washington, D.C., USA, 1993. AAAI Press/MIT Press.

[Ter00] Eugenia Ternovska. Id-logic and the ramification problem for the situation calculus. In *ECAI*, 2000.

[Thi98] Michael Thielscher. Introduction to the fluent calculus. *Electron. Trans. Artif. Intell.*, 2:179–192, 1998.