Modeling Actions with Ramifications in Nondeterministic, Concurrent, and Continuous Domains— And A Case Study

Michael Thielscher
Dresden University of Technology
01062 Dresden, Germany
mit@inf.tu-dresden.de

Abstract

Combining into a consistent theory co-existing models for different phenomena in reasoning about actions can be a problem as challenging as addressing new aspects. We present a uniform theory for reasoning about actions with indirect effects in nondeterministic, concurrent, and continuous domains. We report on a case study to which our theory has been successfully applied.

Introduction

Research on reasoning about actions in dynamic environments has made rapid progress in the recent past: Initiated by new, solid solutions to the Frame Problem in the early 1990s,[1] a variety of advanced aspects of complex environments has been successfully addressed, among which are: indirect effects of actions, concurrency, uncertainty, sensing actions, and continuous change in conjunction with so-called natural actions, to mention just the ones on which most of recent work has focused.

However, the existence of models for all of these and other aspects does not imply that there be a unique model which covers them all. Rather, extensions of basic solutions to the Frame Problem have mostly been investigated in isolation. As a consequence, combining co-existing models for different phenomena is often a problem as challenging as addressing further aspects.

In this paper, we present a classical logic formalism that uniformly addresses the diverse phenomena of ramifications (i.e., indirect effects), nondeterminism, concurrency, and continuous change. Our formalism is based on the concept of state update axioms of the Fluent Calculus as one established solution to the Frame Problem (Thielscher 1999b), which roots in the logic programming formalism of (Hölldobler & Schneeberger 1990).

We have successfully applied our theory to the Traffic World, a complex dynamic domain which has recently been posed as a challenge to the scientific community (Sandewall 1999). The crucial property of this domain is that it shows all of the aforementioned phenomena, and hence cannot be fully axiomatized by means of co-existing but not readily compatible solutions to each aspect alone. Using the combined theory of the Fluent Calculus, in (Henschel & Thielscher 1999) we have developed an axiomatization of the full Traffic World,[2] which proves the expressiveness of our framework.

In the next section, we give a short introduction to the case study of the Traffic World, and we briefly recapitulate the fundamentals of the Fluent Calculus.[3] We then discuss a first challenge which is raised by combining the two aspects of uncertainty and natural actions and which concerns the specification of action preconditions. Thereafter, we present a theory which integrates ramification, nondeterminism, and concurrency in the Fluent Calculus, followed by incorporating continuous change and natural actions. Throughout the paper, key axioms taken from (Henschel & Thielscher 1999) serve as examples.

The Basic Fluent Calculus

The Traffic World consists of a net of road segments on which cars are moving. Cars may speed up, slow down, and turn at intersections (that is, nodes at which segments meet). Speed changes are approximated as instantaneous, for the sake of simplicity. Each car has its own top speed, roads have speed limits, and cars must keep a certain safety distance to the car in front. One or more cars may be under our control, others are not. A logical axiomatization of the Traffic World can be used to solve, using automated deduction, various kinds of problems, such as: What can be concluded about the future of a (possibly incompletely) given state? How did a given state evolve, say, a traffic jam? How can certain goals be achieved? Rigorous axiomatizations moreover allow for proving general properties, such as congestion avoidance under certain conditions etc. For details we refer to (Sandewall 1999).

[1] A good overview of today's established action formalisms is provided by the set of reference articles published in (Sandewall 1998).

[2] It is worth mentioning that the Fluent Calculus was the first of the standard approaches in which a solution to the challenge problem was submitted.

[3] Due to lack of space we can provide only a brief description of the Fluent Calculus; for a complete introduction see (Thielscher 1999b)—or get an online tutorial at `http://pikas.inf.tu-dresden.de/~mit/FC/Tutorial/index.htm`.

The Traffic World involves a variety of aspects of real-world action domains:

- **Continuous change.** The position of a car which moves with constant velocity changes continuously.
- **Concurrency.** Two or more cars may arrive at an intersection at the same time and even with the intention to turn onto the same road segment.
- **Natural actions.** Cars are assumed to automatically slow down as soon as they have reached the safety distance to the car in front.
- **Ramification.** If the driver of a car hits the break and slows down as effect, then another car traveling behind and keeping just the safety distance must slow down, too, as an indirect effect, which in turn may cause a third car to slow down traveling behind the second one, and so on.
- **Nondeterminism.** Cars which are not under our control may choose either direction at intersections.

The Fluent Calculus uses the basic entity of a fluent, which is an atomic component of descriptions of world states. While fluents are generally assumed to be stable in between the occurrence of two consecutive actions, a fluent may internally represent an arbitrarily complex, continuous process. The central fluent used for the Traffic World, denoted by $\mathtt{Movement}(\mathtt{x}, \mathtt{d}, \mathtt{v}, \mathtt{t}, \mathtt{r}, \mathtt{n})$, represents such a process, namely, the constant movement of a car $\mathtt{x}$. The other parameters are: the distance absolved on the current road segment at the time of initiation of the particular movement ($\mathtt{d}$), the velocity ($\mathtt{v}$), the time when the movement was initiated ($\mathtt{t}$), the road segment ($\mathtt{r}$), and the node the car is heading for ($\mathtt{n}$).

The key feature of the Fluent Calculus is that it introduces an explicit notion of a state to the Situation Calculus. This requires the meticulous distinction between situations (which are characterized by sequences of actions) and states (which are characterized by truth-assignments to fluents). Formally, the Fluent Calculus is an order-sorted second order language with equality, which includes the predefined sorts *sit*, *action*, *fluent*, and *state*. Fluents are reified propositions. That is to say, the symbol $\mathtt{Movement}$ from above denotes a function symbol which maps a 6-tuple of the right sort onto a *fluent*. Fluents can be joined together by the binary function symbol "$\circ$" to make up states. We write this symbol in infix notation. The function shall satisfy the laws AC1, i.e., associativity, commutativity, and existence of a unit element, denoted by $\emptyset$. Associativity allows us to omit parentheses in nested applications of $\circ$.

The standard function $\mathit{State} : \mathit{sit} \mapsto \mathit{state}$ relates a situation to the state of the world in that situation. The following axiom, for instance, specifies two movements taking place in the initial situation $\mathtt{S_0}$:

$$
\begin{aligned}
&(\exists \mathtt{z})\, \mathtt{State}(\mathtt{S_0}) = \\
&\quad \mathtt{Movement}(\mathtt{X_1}, \mathtt{2.5km}, \mathtt{70kmph}, \mathtt{9:30:00}, \mathtt{R_{14}}, \mathtt{N_8}) \circ \\
&\quad \mathtt{Movement}(\mathtt{X_2}, \mathtt{3.7km}, \mathtt{50kmph}, \mathtt{9:32:30}, \mathtt{R_6}, \mathtt{N_{22}}) \circ \mathtt{z}
\end{aligned}
$$

That is, of $\mathtt{State}(\mathtt{S_0})$ it is known that it includes the two fluents mentioned and possibly some other fluents $\mathtt{z}$.[4]

For convenience, we will frequently use the expressions $\mathit{Holds}(f, z)$ —denoting that f holds in state z —and the common $\mathit{Holds}(f, s)$—stating that fluent f holds in situation s—, though they are not part of the signature but mere abbreviations of equality sentences:

$$
\begin{aligned}
\mathit{Holds}(f, z) &\stackrel{\text{def}}{=} (\exists z')\, z = f \circ z' \\
\mathit{Holds}(f, s) &\stackrel{\text{def}}{=} \mathit{Holds}(f, \mathit{State}(s))
\end{aligned}
$$

So-called state constraints are used to restrict the space of states to those that can actually occur. The following, for instance, says that one and the same car can never execute two movements in the same situation:

$$
\begin{aligned}
&\mathtt{Holds}(\mathtt{Movement}(\mathtt{x}, \mathtt{d_1}, \mathtt{v_1}, \mathtt{t_1}, \mathtt{r_1}, \mathtt{n_1}), \mathtt{s}) \\
&\wedge \mathtt{Holds}(\mathtt{Movement}(\mathtt{x}, \mathtt{d_2}, \mathtt{v_2}, \mathtt{t_2}, \mathtt{r_2}, \mathtt{n_2}), \mathtt{s}) \\
&\quad \supset \mathtt{d_1} = \mathtt{d_2} \wedge \mathtt{v_1} = \mathtt{v_2} \wedge \mathtt{t_1} = \mathtt{t_2} \wedge \mathtt{r_1} = \mathtt{r_2} \wedge \mathtt{n_1} = \mathtt{n_2}
\end{aligned}
$$

A further example of a state constraint in the Traffic World can be found in the section on ramifications.

Fundamental for any Fluent Calculus axiomatization is the axiom set

$$\mathit{EUNA}[\circ, \emptyset; \mathit{fluent}, \mathit{state}] \qquad \text{(F1)}$$

which accompanies domain-dependent unique name-assumptions by the axioms AC1 for $\circ; \emptyset$ along with the two axioms of irreducibility and of Levi, which entail inequality of two state terms that are composed of different fluents; see (Henschel & Thielscher 1999). In addition, we have the foundational axiom

$$\mathit{State}(s) \neq f \circ f \circ z \qquad \text{(F2)}$$

by which double occurrences of fluents are prohibited in any state which is associated with a situation.

So-called state update axioms specify the entire relation between the states at two consecutive situations. In the basic case of deterministic actions with only direct and closed effects,[5] a mere equation relates a successor state $\mathit{State}(\mathit{Do}(a, s))$ [6] to the preceding state $\mathit{State}(s)$:

$$
\begin{aligned}
&\mathit{Poss}(A(\vec{x}), s) \wedge \Delta(\vec{x}, s) \supset \\
&\quad (\exists \vec{y})\, \mathit{State}(\mathit{Do}(A(\vec{x}), s)) \circ \vartheta^- = \mathit{State}(s) \circ \vartheta^+
\end{aligned}
$$

where the standard predicate $\mathit{Poss}(a, s)$ denotes that action a is possible in situation s and where ϑ^- are the negative effects and ϑ^+ the positive effects, resp., of action $A(\vec{x})$ under condition $\Delta(\vec{x}, s)$ (sequence $\vec{y}$ contains the variables in ϑ^-, ϑ^+ which are not among $\vec{x}$).[7]

More complex phenomena require more complex forms of state update axioms, as we will see later in the paper.

[4] A word on the notation: Predicate and function symbols, including constants, start with a capital letter whereas variables are in lower case, sometimes with sub- or superscripts. Free variables in formulas are assumed universally quantified. Throughout the paper, action variables are denoted by the letter a, situation variables by the letter s, fluent variables by the letter f, and state variables by the letter z, all possibly with sub- or superscript.

[5] By closed effects we mean that an action does not have an unbounded number of direct effects.

[6] As in the standard Situation Calculus, $\mathit{Do}(a, s)$ denotes the situation reached after performing action a in situation s.

[7] This scheme is the reason for not stipulating that "$\circ$" be idempotent, contrary to what one might intuitively expect instead of (F2). For if the function were idempotent, then the equation would not imply that $\mathit{State}(\mathit{Do}(a, s))$ does not include ϑ^- .

Preconditions of Natural Actions in Nondeterministic Worlds

In formalisms based on the Situation Calculus, such as (Reiter 1991), as well as in the Fluent Calculus a usual premise is that the preconditions of an action $A(\vec{x})$ can be described by a definitional formula

$$\mathit{Poss}(A(\vec{x}), s) \equiv \pi_A(\vec{x}, s)$$

where the first-order formula π_A does not include the predicate *Poss* and describes the conditions on parameters $\vec{x}$ and situation s under which the action is possible. This assumption usually generalizes to the case of non-deterministic worlds as well as to the case of natural actions, i.e., which occur automatically. Surprisingly, the premise fails if the two aspects are combined.

The reason lies in the fact that all natural actions which are possible must actually occur (Reiter 1996). Yet in a non-deterministic world, several instances of a natural action may be possible but only one of them can actually take place. In this case, the possibility of one natural action depends on other actions not being possible at the same time.

An example from the Traffic World is the natural action of someone else's car turning at intersections. A car has a choice among several alternatives, but in any concrete model of the world it can turn onto one segment only. Under these circumstances, a definitional precondition axiom of the aforementioned form does not exist. Rather, the specification needs to be split into two parts, the first of which is of the form

$$\mathit{Poss}(A(\vec{x}, \vec{y}), s) \supset \pi_A(\vec{x}, s) \wedge \widehat{\pi}_A(\vec{x}, \vec{y}, s)$$

where $\vec{y}$ are the parameters among which a non-deterministic choice has to be made and where $\pi_A(\vec{x}, s) \wedge \widehat{\pi}_A(\vec{x}, \vec{y}, s)$ describes the necessary precondition of $A(\vec{x}, \vec{y})$ in s. The second part of the precondition axiomatization stipulates uniqueness of the choice:

$$\pi_A(\vec{x}, s) \supset (\exists! \vec{y})\, \mathit{Poss}(A(\vec{x}, \vec{y}), s)$$

Based on this generalization, the following two axioms describe the precondition of someone else's car arriving and nondeterministically turning at intersections:

$$\begin{array}{l} \mathtt{Poss}(\mathtt{ArriveAt}(\mathtt{x}, \mathtt{r_1}, \mathtt{n}, \mathtt{r_2}, \mathtt{t}), \mathtt{s}) \supset \\ \quad (\exists \mathtt{d}, \mathtt{v}, \mathtt{t_0}, \mathtt{n_2})\, (\mathtt{Holds}(\mathtt{Movement}(\mathtt{x}, \mathtt{d}, \mathtt{v}, \mathtt{t_0}, \mathtt{r_1}, \mathtt{n}), \mathtt{s}) \wedge \\ \qquad \mathtt{d} + \mathtt{v}(\mathtt{t} - \mathtt{t_0}) = \mathtt{Length}(\mathtt{r_1}) \wedge \\ \qquad \mathtt{Connects}(\mathtt{r_2}, \mathtt{n}, \mathtt{n_2}) \wedge \mathtt{r_1} \neq \mathtt{r_2}) \end{array}$$

where $\mathtt{ArriveAt}(\mathtt{x}, \mathtt{r_1}, \mathtt{n}, \mathtt{r_2}, \mathtt{t})$ denotes the natural action at time $\mathtt{t}$ of car $\mathtt{x}$ arriving at node $\mathtt{n}$ as the end of segment $\mathtt{r_1}$ and turning onto segment $\mathtt{r_2}$; $\mathtt{Length}(\mathtt{r})$ denotes the length of segment $\mathtt{r}$; and $\mathtt{Connects}(\mathtt{r}, \mathtt{n_1}, \mathtt{n_2})$ denotes that road segment $\mathtt{r}$ connects nodes $\mathtt{n_1}$ and $\mathtt{n_2}$. Since there can only be one road segment $\mathtt{r_2}$ onto which the car may turn, the second axiom requires uniqueness of this parameter:

$$\begin{array}{l} (\exists \mathtt{d}, \mathtt{v}, \mathtt{t_0}, \mathtt{n_2})\, (\mathtt{Holds}(\mathtt{Movement}(\mathtt{x}, \mathtt{d}, \mathtt{v}, \mathtt{t_0}, \mathtt{r_1}, \mathtt{n}), \mathtt{s}) \wedge \\ \qquad \mathtt{d} + \mathtt{v}(\mathtt{t} - \mathtt{t_0}) = \mathtt{Length}(\mathtt{r_1})) \\ \quad \supset (\exists! \mathtt{r_2})\, \mathtt{Poss}(\mathtt{ArriveAt}(\mathtt{x}, \mathtt{r_1}, \mathtt{n}, \mathtt{r_2}, \mathtt{t}), \mathtt{s}) \end{array}$$

Concurrency with Ramification

Concurrency

The Fluent Calculus for concurrent actions (Thielscher 2000b) is based on the additional pre-defined sort *conc*, of which *action* is a sub-sort. Single actions which are performed simultaneously are joined together with a new binary function. This function shares with the function combining fluents to states the properties of associativity, commutativity, and existence of a unit element. Hence, the symbol "$\circ$" is overloaded as denotation for both. The constant "ϵ" (read: *no-op*) acts as the unit element wrt. $\circ$ applied to terms of sort *conc*. In summary, the concurrent Fluent Calculus relies on the equality axioms (c.f. (F1)),

$$\mathit{EUNA}[\circ, \epsilon; \mathit{action}, \mathit{conc}] \tag{F3}$$

In what follows, variables of the new sort are denoted by the letter c, possibly with sub- or superscript.

State update axioms for concurrent actions are recursive. They specify the effect of an action relative to the effect of arbitrary other, concurrent actions:

$$\begin{array}{l} \mathit{Poss}(\alpha(\vec{x}) \circ c, s) \wedge \Delta(\vec{x}, c, s) \supset \\ \quad (\exists \vec{y})\, \mathit{State}(\mathit{Do}(\alpha(\vec{x}) \circ c, s)) \circ \vartheta^- = \mathit{State}(\mathit{Do}(c, s)) \circ \vartheta^+ \end{array}$$

That is, ϑ^- and ϑ^+ are the additional negative and positive, resp., effects which occur if α is performed besides c. Here, α can be a single action or a compound action which produces synergic effects, that is, effects which no single action would have if performed alone. With the help of recursive state update axioms, the effect of, say, two simultaneous but independent actions can be inferred by first inferring the effect of one of them and, then, inferring the effect of the other action on the result of the first inference. The recursion relies on the base case of the empty action, which has no effect:

$$\mathit{State}(\mathit{Do}(\epsilon, s)) = \mathit{State}(s)$$

Two or more actions may interfere when performed concurrently, which is why condition Δ in the above state update axiom may restrict the applicability of the implication in view of the concurrent action c.

Ramification

In the Fluent Calculus with ramifications (Thielscher 1997), indirect effects of actions are accounted for by the successive application of so-called causal relationships. Their axiomatization is based on defining a predicate $\mathit{Causes}(z_0, z_0^+, z_0^-, z_1, z_1^+, z_1^-)$, which means that in the current state z_0 the occurred positive and negative effects z_0^+, z_0^- give rise to an additional, indirect effect resulting in the updated state z_1 and the updated current effects z_1^+, z_1^-. For instance, the following causal relationship implies that if a car $\mathtt{x}$ has lowered its speed as a direct or indirect effect of some action, then this causes another car $\mathtt{x}'$ traveling behind in the global safety distance ς to slow down as well:

$$\begin{array}{l} \mathtt{Causes}(z_0, z_0^+, z_0^-, z_1, z_1^+, z_1^-) \equiv \\ \ (\exists t_0, x, d, v, t, r, n, x', d', v', t') \\ \ \ [\mathtt{Start}(z_0) = t_0 \wedge \\ \ \ \mathtt{Holds}(\mathtt{Movement}(x, d, v, t, r, n), z_0) \wedge \\ \ \ \mathtt{Holds}(\mathtt{Movement}(x', d', v', t', r, n), z_0) \wedge \\ \ \ \varsigma = d + v(t_0 - t) - (d' + v'(t_0 - t')) \wedge \\ \ \ v < v' \wedge \\ \ \ z_1 \circ \mathtt{Movement}(x', d', v', t', r, n) = \\ \ \ \ \ z_0 \circ \mathtt{Movement}(x', d' + v'(t_0 - t'), v, t_0, r, n) \wedge \\ \ \ z_1^+ = z_0^+ \circ \mathtt{Movement}(x', d' + v'(t_0 - t'), v, t_0, r, n) \wedge \\ \ \ z_1^- = z_0^- \circ \mathtt{Movement}(x', d', v', t', r, n)] \\ \ \vee \ldots \end{array}$$

(The ellipsis indicates that the Traffic World axiomatization includes more causal relationships. For the definition of $\mathit{Start}(z)$ see foundational axiom (F7) below.) Applications of the above causal relationship ensure that the following state constraint will not be violated, which says that cars must in any situation keep the safety distance:

$$\begin{array}{l} \mathtt{Holds}(\mathtt{Movement}(x_1, d_1, v_1, t_1, r, n), s) \wedge \\ \mathtt{Holds}(\mathtt{Movement}(x_2, d_2, v_2, t_2, r, n), s) \wedge \\ x_1 \neq x_2 \wedge \mathtt{Start}(s) = t \supset \\ \ \ d_1 + v_1(t - t_1) - (d_2 + v_2(t - t_2)) \geq \varsigma \wedge \\ \ \ [d_1 + v_1(t - t_1) - (d_2 + v_2(t - t_2)) = \varsigma \supset v_2 \leq v_1] \end{array} \tag{1}$$

Causal relationships are repeatedly applied until a state is obtained which does not violate the state constraints. Our causal relationship from above is an excellent demonstration of the power of ramification: It is intended that if a car leading a whole convoy slows down, then in a manner of falling dominoes the effect of deceleration gets propagated. To this end, the general idea of ramification needs to be combined with the recursive effect specifications needed for concurrency. The combined axiomatization will allow for reasoning about interesting cases such as the following. Suppose car X_1 is followed by faster car X_2, which is in turn followed by car X_3, the fastest of all. Suppose further that X_2 reaches the safety distance to X_1 at the very same moment as X_3 approaches X_2. Then X_2 assumes the speed of X_1 and X_3 that of X_2 as direct effects; thereafter, ramification changes the speed of X_3 again, namely, to the new speed of X_2.

The Combination

The challenge with combining the two phenomena of concurrency and ramification is that two or more actions may produce direct effects which only together cause some indirect effect. (The scenario just mentioned constitutes such a case.) Therefore, ramifications must not be inferred separately for each member of a compound action. Hence, a new form of recursive effect specifications is required by which an intermediate state is determined in which all direct effects are realized but which is not yet the overall successor state. To this end, we introduce three functions, $\mathit{DSucc}, \mathit{DEff}^+, \mathit{DEff}^-$, mapping a concurrent action and a state to states which denote, resp., the world state resulting from the direct effects of the concurrent action and the combined positive and negative effects. On this basis, the new general form of state update axioms is as follows.[8]

$$\begin{array}{l} \mathit{Poss}(\alpha(\vec{x}) \circ c, z) \wedge \Delta(\vec{x}, c, z) \supset \\ \ (\exists \vec{y}, z^+, z^-) \\ \ \ [z^+ = \vartheta^+ \wedge z^- = \vartheta^- \wedge \\ \ \ \mathit{DSucc}(\alpha(\vec{x}) \circ c, z) \circ z^- = \mathit{DSucc}(c, z) \circ z^+ \wedge \\ \ \ \mathit{DEff}^+(\alpha(\vec{x}) \circ c, z) = \mathit{DEff}^+(c, z) \circ z^+ \wedge \\ \ \ \mathit{DEff}^-(\alpha(\vec{x}) \circ c, z) = \mathit{DEff}^-(c, z) \circ z^-] \end{array}$$

The base case for this recursion is given by this foundational axiom:

$$\begin{array}{l} \mathit{DSucc}(\epsilon, z) = z \wedge \\ \mathit{DEff}^+(\epsilon, z) = \emptyset \wedge \mathit{DEff}^-(\epsilon, z) = \emptyset \end{array} \tag{F4}$$

Based on the combined direct effects of a concurrent action, ramification yields all indirect effects:

$$\begin{array}{l} \mathit{Ramify}(\mathit{DSucc}(c, z), \mathit{DEff}^+(c, z), \mathit{DEff}^-(c, z), \\ \ \ \ \ \mathit{Succ}(c, z)) \end{array} \tag{F5}$$

where $\mathit{Ramify}(z_0, z^+, z^-, z)$ means that the successive application of (zero or more) causal relationships to state z_0 and effects z^+, z^- results in state z.

Ramification is the repeated application of causal relationships:

$$\begin{array}{l} \mathit{Ramify}(z_1, z_1^+, z_1^-, z_2) \equiv \\ \ (\exists z_2^+, z_2^-)\, (z_1, z_1^+, z_1^-, z_2, z_2^+, z_2^-) \in \mu[\mathit{Causes}] \end{array} \tag{F6}$$

where $(\vec{x}, \vec{y}) \in \mu[P]$ abbreviates the following formula, which is a standard second-order schema to axiomatize that $(\vec{x}, \vec{y})$ belongs to the reflexive and transitive closure of predicate P:

$$\forall \Pi \left\{ \begin{array}{l} (\forall \vec{u})\, \Pi(\vec{u}, \vec{u}) \wedge \\ (\forall \vec{u}, \vec{v}, \vec{w})\, [\, \Pi(\vec{u}, \vec{v}) \wedge P(\vec{v}, \vec{w}) \supset \Pi(\vec{u}, \vec{w})\,] \\ \qquad \qquad \supset \Pi(\vec{x}, \vec{y}) \end{array} \right\}$$

An example of a recursive effect specification is the following, which describes the direct effect of a car $\mathtt{x}$ arriving at an intersection $\mathtt{n}$ and the immediate turn onto another segment r_2. For a better understanding of the axiom, we note that our model of the Traffic World handles jams at intersections by virtual waiting areas in each outgoing road segment, which house all cars waiting in line for that segment to become free up to the safety distance; see (Henschel & Thielscher 1999).

$$\begin{array}{l} \mathtt{Poss}(\mathtt{ArriveAt}(x, r_1, n, r_2, t) \circ c, z) \wedge \\ \neg \mathtt{Cancels}(c, \mathtt{ArriveAt}(x, r_1, n, r_2, t), z) \supset \\ \ (\exists d, p, v, t', t_0) \\ \ \ [z^+ = \mathtt{Waiting}(x, n, r_2, p) \circ \mathtt{Counter}(n, r_2, p+1) \wedge \\ \ \ z^- = \mathtt{Movement}(x, d, v, t', r_1, n) \circ \mathtt{Counter}(n, r_2, p) \wedge \\ \ \ \mathtt{DSucc}(\mathtt{ArriveAt}(x, r_1, n, r_2, t) \circ c, z) \circ z^- = \\ \qquad \qquad \qquad \mathtt{DSucc}(c, z) \circ z^+ \wedge \\ \ \ \mathtt{DEff}^+(\mathtt{ArriveAt}(x, r_1, n, r_2, t) \circ c, z) = \\ \qquad \qquad \qquad \mathtt{DEff}^+(c, z) \circ z^+ \wedge \\ \ \ \mathtt{DEff}^-(\mathtt{ArriveAt}(x, r_1, n, r_2, t) \circ c, z) = \\ \qquad \qquad \qquad \mathtt{DEff}^-(c, z) \circ z^-] \end{array}$$

[8] In anticipation of the integration of continuous change, the situation argument s is replaced by the state argument z, and the expression $\mathit{State}(\mathit{Do}(c, s))$ is replaced by $\mathit{Succ}(c, z)$—denoting the successor state of performing concurrent action c in state z—, in all specifications related to update in the following.

Put in words, if the action is not canceled by the concurrent actions, then car x is now waiting in line, there is one more car waiting to leave node n for segment r_2, car x is no longer moving, and the previous value of the counter becomes false.

Turning is a nice example also to demonstrate the problem of interfering concurrent actions. If more than one ArriveAt actions take place at the same time and do not conflict, e.g., because they occur at different intersections, then the recursive application of the above axiom ensures that each turn has the effect as specified. However, if two or more cars arrive at the same node with the intention to turn onto the same segment, then the consequents of the corresponding instances are in conflict. The reason being that different decompositions of the concurrent action lead to different queues, that is, the incoming cars are not placed in a unique order.

The conflict admits an elegant solution by defining cancellation of actions in the following way. Suppose a car attempts to turn onto a segment which another car has chosen, too, having higher priority according to the right-of-way regulation at the intersection in question. Then the former cancels the latter, thus avoiding that the decomposition starts with the action that has higher priority. As a consequence, only those decompositions of a concurrent action are possible without cancellation where all turns onto one segment are inferred according to the priority ordering:[9]

$$\begin{array}{l}\mathtt{Cancels}(c, \mathtt{ArriveAt}(x_1, r_1, n, r_2, t), z) \equiv \\ \quad (\exists x_2, r_1')\,(\,\mathtt{In}(\mathtt{ArriveAt}(x_2, r_1', n, r_2, t), c) \wedge \\ \qquad\qquad \mathtt{Priority}(r_1, n, r_2) > \mathtt{Priority}(r_1', n, r_2))\end{array}$$

where $\mathtt{Priority}(r_1, n, r_2)$ denotes the priority that cars coming from segment r_1 have at node n regarding a turn into r_2, and where we use the following macro:

$$\mathit{In}(c_1, c) \stackrel{\text{def}}{=} (\exists c')\, c = c_1 \circ c'$$

Integrating Continuous Change

Continuous Change and Natural Actions

The basic representation mechanism for continuous change is the introduction of process fluents. The Fluent Calculus for continuous change moreover relies on the distinction between deliberative and so-called natural actions. The latter are not subject to the free will of a planning agent. Rather they happen automatically under specific circumstances. If, for instance, a car which is not under our control has absolved the entire length of a segment, then it will automatically perform an ArriveAt action, thus causing an 'autonomous' update of the system state.

The crucial notion underlying the Fluent Calculus for continuous change is that of a situation tree with trajectories (Thielscher 1999a). In any situation, natural actions may cause an autonomous evolution of the state associated with that situation.[10] To this end, each situation has a trajectory. A trajectory is a sequence of states. The world state resulting from a deliberative action is the first one on a new trajectory. The further evolution of that trajectory is then determined by the natural actions that are expected to happen. The performance of a deliberative action, on the other hand, brings about another situation again, with its own trajectory; see Fig. 1.

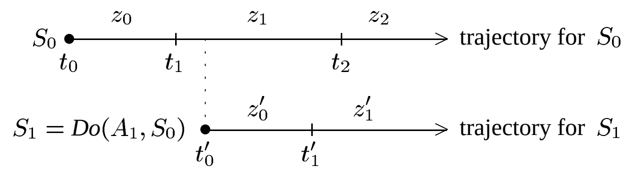

Figure 1: Each situation has its own trajectory, which describes how the state evolves according to the expected natural actions. In the example shown here, at the time t_0' when the deliberative action A_1 is performed in situation S_0, the world is no longer in the initial state z_0 due to a natural action happening at time $t_1 < t_0'$, which causes state z_1 to arise. The effect of A_1 is to transform z_1 into z_0', which thus becomes the initial state of the trajectory for situation $\mathit{Do}(A_1, S_0)$.

The Combination

Integrating the concepts of continuous change and natural actions into the Fluent Calculus for concurrency and ramification requires to reason about the updating of states that are not necessarily associated with a situation. This is the reason for replacing the situation argument in precondition and effect axioms by the more general state argument (recall Footnote 8). This shift is straightforward since the expression $\mathit{Holds}(f, s)$ means nothing but $\mathit{Holds}(f, \mathit{State}(s))$ anyway.

The combined Fluent Calculus uses the pre-defined fluent $\mathit{StartTime}(t)$, where t is of sort $\mathbb{R}$, determining the time $\mathit{Start}(z)$ at which a state arises:

$$\begin{array}{l}(\exists! t)\, \mathit{Holds}(\mathit{StartTime}(t), z) \\ \quad \supset (\forall t)\,(\mathit{Holds}(\mathit{StartTime}(t), z) \supset \mathit{Start}(z) = t)\end{array} \tag{F7}$$

The starting time of all states associated with a situation is unique:

$$\begin{array}{l}(\exists! t)\, \mathit{Holds}(\mathit{StartTime}(t), s) \wedge \\ \mathit{Start}(s) = \mathit{Start}(\mathit{State}(s))\end{array} \tag{F8}$$

The evolution of a trajectory is modeled using the predicate $\mathit{Trajectory}(z, z')$, which indicates that z' occurs on the trajectory rooted in z. To model correctly the evolution at any state, all natural actions that are possible next determine the successor state:

$$\begin{array}{l}\mathit{Trajectory}(z, z) \wedge \\ [\mathit{Trajectory}(z, z') \wedge \mathit{NextNatActions}(c, z') \supset \\ \qquad\qquad \mathit{Trajectory}(z, \mathit{Succ}(c, z'))]\end{array} \tag{F9}$$

[9] To see this, suppose A_1 and A_2 are turn actions such that $\mathtt{Cancels}(A_2, A_1, z)$. Then the only decomposition which avoids cancellation is, $\mathtt{DSucc}(A_1 \circ A_2, z) \rightarrow \mathtt{DSucc}(A_1, z) \rightarrow \mathtt{DSucc}(\epsilon, z)$, where A_1 is performed 'first' in z.

[10] Our approach differs in this respect from the approach of (Reiter 1996), where natural actions are included in situation terms. This intertwining the two kinds of actions has been shown unsuited for planning under incomplete information (Thielscher 1999a).

where

$$\textit{NextNatActions}(c,z) \stackrel{\text{def}}{=}$$
$$(\exists t)\,[\,\textit{Start}(z) \le t \wedge \textit{ExpectedNatActions}(z,t,c) \wedge \neg(\exists t',c')\,(\textit{ExpectedNatActions}(z,t',c') \wedge \textit{Start}(z) \le t' < t)\,]$$

which in turn makes use of the following macro:

$$\textit{ExpectedNatActions}(z,t,c) \stackrel{\text{def}}{=} c \neq \epsilon \wedge (\forall a)\,[\textit{In}(a,c) \equiv \textit{Natural}(a) \wedge \textit{Time}(a) = t \wedge \textit{Poss}(a,z)] \wedge (\forall a,c')\, c \neq a \circ a \circ c'$$

Here, $\textit{Time}(a)$ denotes the time of action a, for example, $\texttt{Time}(\texttt{ArriveAt}(\texttt{c},\texttt{r}_1,\texttt{n},\texttt{r}_2,\texttt{t})) = \texttt{t}$; and $\textit{Natural}(a)$ means that a is a natural action. Each domain is assumed to include a suitable axiom defining the positive instances of *Natural*.

No states other than the ones according to axiom (F9) may occur on a trajectory:

$$\textit{ActualState}(s,t,z) \wedge \textit{ActualState}(s,t,z') \supset z = z' \qquad \text{(F10)}$$

where $\textit{ActualState}(s,t,z)$ is true if z is the state of the world in situation s at time t:

$$\textit{ActualState}(s,t,z) \stackrel{\text{def}}{=} \textit{Trajectory}(\textit{State}(s),z) \wedge t > \textit{Start}(s) \wedge (\forall a)\,(\textit{Natural}(a) \wedge \textit{Poss}(a,z) \supset \textit{Time}(a) > t)$$

State constraints, such as (1), need to be generalized so that they apply to all actual states in a situation. Thus, a state constraint in the general setting becomes an implication of the form, $\textit{ActualState}(s,t,z) \supset \Phi(z)$.

The final complication raised by combining concurrency, ramification, and natural actions is that natural actions may by coincidence happen at the very same time at which a compound deliberative action shall be performed. This requires to infer the direct and indirect effects of all actions together:

$$\textit{Poss}(c,s) \wedge \textit{ActualState}(s,\textit{Time}(c),z) \supset \textit{ExpectedNatActions}(z,\textit{Time}(c),c') \supset \textit{State}(\textit{Do}(c,s)) = \textit{Succ}(c \circ c',z) \qquad \text{(F11)}$$

Having made the shift from situation to state arguments in precondition axioms, the expression $\textit{Poss}(c,s)$ is now a mere macro. It shall be true iff all actions in c occur at the same time t, are not natural actions, and are possible in conjunction with all natural actions that are expected at time t:

$$\textit{Poss}(c,s) \stackrel{\text{def}}{=} \textit{TimeUniform}(c) \wedge (\forall a)(\textit{In}(a,c) \supset \neg\textit{Natural}(a)) \wedge (\forall z,c')\,[\,\textit{ActualState}(s,\textit{Time}(c),z) \wedge \textit{ExpectedNatActions}(z,\textit{Time}(c),c') \supset \textit{Poss}(c \circ c',z)]$$

with

$$\textit{TimeUniform}(c) \stackrel{\text{def}}{=} (\forall a,a')\,(\textit{In}(a,c) \wedge \textit{In}(a',c) \supset \textit{Time}(a) = \textit{Time}(a'))$$

where we have the following final foundational axiom, which defines the time of a time-uniform compound action:

$$\textit{TimeUniform}(c) \supset \textit{In}(a,c) \supset \textit{Time}(c) = \textit{Time}(a) \qquad \text{(F12)}$$

Summary

We have presented a uniform classical logic formalism for reasoning about actions which covers a variety of complex phenomena such as ramifications, concurrency, and continuous change. To the best of our knowledge, this is the first framework which allows to model domains involving all of these aspects, since in particular the Ramification Problem has mostly been investigated in isolation. We have illustrated how this theory has been successfully applied to a case study that has recently been posed as a challenge to the scientific community. Future work consists in extending the theory further, especially by integrating our Fluent Calculus model of sensing actions (Thielscher 2000a).

References

Henschel, A., and Thielscher, M. 1999. The LMW traffic world in the fluent calculus. `http://www.ida.liu.se/ext/etai/lmw/TRAFFIC/001`.

Hölldobler, S., and Schneeberger, J. 1990. A new deductive approach to planning. *New Generation Computing* 8:225–244.

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

Reiter, R. 1996. Natural actions, concurrency and continuous time in the situation calculus. In Aiello, L. C.; Doyle, J.; and Shapiro, S., eds., *Proc. of the International Conference on Principles of Knowledge Representation and Reasoning*, 2–13. Cambridge, MA: Morgan Kaufmann.

Sandewall, E., ed. 1998. *Electronic Transactions on Artificial Intelligence 2(3–4). A Collection of Refereed Reference Articles.* `http://www.ep.liu.se/ej/etai/1998`.

Sandewall, E. 1999. Logic Modelling Workshop. `http:// www.ida.liu.se/ext/etai/lmw/`.

Thielscher, M. 1997. Ramification and causality. *Artificial Intelligence* 89(1–2):317–364.

Thielscher, M. 1999a. Fluent Calculus planning with continuous change. *Electronic Transactions on Artificial Intelligence*. (Submitted.) `http://www.ep.liu.se/ea/cis/1999/011/`.

Thielscher, M. 1999b. From Situation Calculus to Fluent Calculus: State update axioms as a solution to the inferential frame problem. *Artificial Intelligence* 111(1–2):277–299.

Thielscher, M. 2000a. Representing the knowledge of a robot. In Cohn, A.; Giunchiglia, F.; and Selman, B., eds., *Proc. of the International Conference on Principles of Knowledge Representation and Reasoning*. Breckenridge, CO: Morgan Kaufmann.

Thielscher, M. 2000b. Solving the inferential frame problem for concurrent actions. (Submitted.) `http://pikas.inf.tu-dresden.de/~mit/publications/conferences/FCconc.ps`.