The Logic of Dynamic Systems

Michael Thielscher
Intellektik, Informatik, TH Darmstadt
Alexanderstraße 10, D–64283 Darmstadt (Germany)
mit@intellektik.informatik.th-darmstadt.de

Abstract

A formal theory of actions and change in dynamic systems is presented. Our formalism is based on the paradigm that state transitions in a system naturally occur while time passes by; one or more agents have the possibility to direct the development of the system by executing actions. Our theory covers concurrency of actions and events and includes a natural way to express delayed effects and nondeterminism. A uniform semantics for specifications of dynamic systems is developed, which enables us to express solutions to problems like temporal projection, planning and postdiction in terms of logical entailment.

1 Introduction

A common assumption underlying most formal theories of actions, such as [Gelfond and Lifschitz, 1993; Sandewall, 1994], is that state transitions in dynamic systems only occur when some agent executes an action—otherwise the state of the system is assumed to be stable. As pointed out for instance in [Pollack, 1992], this view is often too restrictive if one intends to model more realistic scenarios where one or more autonomous agents act in a complex world. In this paper, we present an alternative theory of dynamic systems that is based on a different paradigm: State transitions naturally occur while time passes by. A reasoning agent might influence and direct the development of the system by initiating actions but the system is not assumed to pause in case there are no explicit actions.[1]

The aim of this paper is to develop a logic-oriented, formal theory of dynamically changing worlds and the process of acting in them. We will show that our view provides naturally two main characteristic features. First, the notion of parallelism is an intrinsic element of our theory. This notion includes the concurrent execution of actions as well as the simultaneous occurrence of events. Second, we can easily model delayed effects of actions by initiating additional independent events which eventually trigger a particular effect. This is the case since parallelism normally means that the various changes during a state transition are simultaneously caused by several reasons.

When specifying a dynamic system, a major challenge is to find a compact description of the underlying causal model, which defines the space of possible state changes in the course of time. Our theory includes these two fundamental concepts: First, the *persistence assumption*[2], which enables one to state explicitly only what changes during a single state transition while everything else is implicitly assumed to remain unchanged. Second, *atomic causal laws* are used to state relationships between single cause-effect-pairs. Since usually several of these atomic laws apply to the current state of a dynamic system, a combination of such laws determines the complete transition step. The use of atomic causal relationships is especially necessary in theories which involve concurrency of actions and events.

The major application of formal specifications of dynamic systems is to address one of the following three problem categories:

- In a *temporal projection* problem, one is interested in the result of executing a particular sequence of actions starting in a particular state of the system.
- In a *planning* problem, the question is whether a sequence of actions, taken as a *plan*, can be found whose execution in the system results in the satisfaction of a given goal. Below, we will illustrate that our theory enables us to formulate more general planning problems compared to the classical AI definition of planning. We are able to specify that the properties we strive for might be distributed over several states of the system and need not necessarily be satisfied in a single final state, until otherwise required.
- In a *postdiction* problem[3], one is faced with a narrative which is represented by a number of observations regarding a system's development during a specific period. These observations are used to derive more information about what has happened. Our theory generalizes former work such as [Gelfond

[1] A similar principle was recently introduced in [Große, 1994] in the context of a modal logic approach to reasoning about actions.

[2] also called *frame assumption* or *inertia principle*

[3] called *chronicle completion* in [Sandewall, 1994]

and Lifschitz, 1993; Sandewall, 1994] in so far as the sequence of actions performed during the period in question is not necessarily completely known.

In the course of this paper, we will give precise definitions of all these three problem categories in terms of our theory. We will illustrate that this results in a model-based semantics for formal specifications of dynamic systems along with particular problem instances, which enables us to define solutions to such problems in terms of logical entailment.

The paper is organized as follows. In Section 2, we define *deterministic* dynamic systems, where a unique successor state can always be determined provided the current state is completely known. For sake of simplicity, we focus on so-called *propositional* dynamic systems, where states are described by means of a finite number of propositional constants. We introduce the notion of a logical formula to describe properties of system states. Such a formula can be used, for instance, to specify which combinations of truth values for the underlying propositional constants denote impossible (inconsistent) states. In Section 3, we define the notions of persistence assumption and atomic causal laws; they both are fundamental for a compact and intuitive specification of the behavior of a dynamic system. We introduce a certain specificity criterion in order to resolve conflicts which occur whenever the concurrent execution of actions or the simultaneous occurrence of events, respectively, has different effects than the serial execution. In Section 4, we develop a model-based semantics for specifications of dynamic systems and define the process of influencing the development of a system by means of executing actions. This allows for a precise formalization of the three aforementioned problem categories. In Section 5, we extend the basic version of our theory to *nondeterministic* dynamic systems, where uncertainty about the successor state may exist even if the system's current state is completely known. In particular, we argue that the concept of nondeterminism provides an interesting and reasonable solution to the problem of concurrently executed actions whose causal laws propose mutually exclusive effects: Instead of declaring such situations impossible, which is the standard proposal in literature (e.g., [Lin and Shoham, 1992; Baral and Gelfond, 1993; Große, 1994]), we take only the disputed effects as uncertain (c.f. [Bornscheuer and Thielscher, 1994]).

The reader should be aware of the purpose of this paper, which is unusual in so far as it merely consists of definitions rather than describing concrete results. We propose a formal framework to specify and to reason about dynamic systems. Thus, our intention is similar to the purpose of the topical systematic approaches [Sandewall, 1994; Gelfond and Lifschitz, 1993], namely, to provide a high-level framework which deserves adequate proof theories or can be used to assess the range of applicability of existing approaches. A brief discussion of this aspect in Section 6 concludes the paper.

2 Specifying Dynamic Systems

A formal specification of a dynamic system consists of two components. First, a collection of *fluents* [McCarthy and Hayes, 1969], which are used to describe particular states of the system. For sake of simplicity, we restrict attention to a finite set of propositional constants here. Second, the behavior of the system as regards state transitions needs to be described. In this and the next two sections, we focus on deterministic dynamic systems. This is reflected in the following definition, where a transition function determines unique successor states.

Definition 1 A *deterministic, propositional dynamic system* is a pair $(\mathcal{F}, \Phi)$ consisting of a finite set of symbols $\mathcal{F}$, called *fluents*, and a partially defined mapping $\Phi : \mathcal{C} \mapsto \mathcal{C}$, called *causal model*. The range of the latter is a particular set of subsets of $\mathcal{F}$, i.e., $\mathcal{C} \subseteq 2^{\mathcal{F}}$. ■

Each subset s of $\mathcal{F}$ determines a (not necessarily possible) *state* of the dynamic system at hand. Each fluent $f \in s$ is then said to be *true* in s while each fluent $f \in \mathcal{F} \setminus s$ is taken to be *false*. The set $\mathcal{C}$ is intended to contain all so-called *consistent* states—only for these states s the *successor* state $\Phi(s)$ is defined through the exhaustively given causal model.

Based on the notion of truth wrt single fluents and states, we can construct (propositional) formulae and define entailment following the standard way:

Definition 2 Let $\mathcal{F}$ be a set of fluents. The set of *fluent formulae* (based on $\mathcal{F}$) is the smallest set such that each element $f \in \mathcal{F}$ is a fluent formula; and if F and G both are fluent formulae then $\neg F$, $(F \wedge G)$, $(F \vee G)$, and $(F \rightarrow G)$ are also fluent formulae.

Given a state $s \subseteq \mathcal{F}$ and a fluent formula F, the notion of F being *true in* s, written $s \models F$, is inductively defined as follows:

- $s \models f$ iff $f \in s$, for each $f \in \mathcal{F}$.
- $s \models \neg F$ iff $s \not\models F$.
- $s \models (F \wedge G)$ iff both $s \models F$ and $s \models G$.
- $s \models (F \vee G)$ iff $s \models F$ or $s \models G$ (or both).
- $s \models (F \rightarrow G)$ iff $s \not\models F$ or $s \models G$ (or both).

■

Fluent formulae can be used, for instance, to specify consistency of states more compactly by means of a particular formula C such that a state $s \in \mathcal{F}$ is defined to be consistent iff $s \models C$. Then, the set $\mathcal{C}$, which contains all consistent states of a dynamic system (see Definition 1), is implicitly given by $\mathcal{C} = \{s \subseteq \mathcal{F} \mid s \models C\}$.

In order to integrate the paradigm mentioned in the introduction, we call some distinguished fluents $\mathcal{F}_a \subset \mathcal{F}$ *actions*; these are fluents which an agent can make true in the current state in order to influence the system's behavior. Hence, *actions are nothing else than elements of a state description*. The formal notion of executing actions will be given below, in Section 4.

Example 1 The Yale Shooting domain will be used, in several variants, as the running example throughout the paper. To formalize a first version, consider the set of fluents $\mathcal{F} = \{alive, loaded, load, shoot\}$—where *alive* and *loaded* are used to describe the state of the turkey and the gun, respectively, while *load* and *shoot* are action fluents to describe the events of loading the gun and shooting with it, respectively. The particular state $s = \{alive, load\} \subseteq \mathcal{F}$, for instance, describes the facts that the turkey is alive, that the gun is unloaded, and that the agent intends to execute the action *load*. The successor state $\Phi(s)$ might then be defined as $\{alive, loaded\}$, stating that the turkey is still alive and that the gun is now loaded. Furthermore, one might wish to specify that the agent cannot simultaneously load the gun and shoot. This can be achieved by means of the consistency criterion $C = \neg(load \wedge shoot)$ such that, say, $s \models C$ due to $\{alive, load\} \not\models shoot$. ■

3 Causal Laws

The main challenge when specifying a dynamic system is the problem of finding a compact representation of the corresponding causal model Φ. The most fundamental concept related to this is the principle of *persistence*, which enables us to only specify the fluents that change their value during a particular state transition; all other fluents are implicitly taken to keep their value.

In our theory, we make a distinction between so-called *static* fluents $\mathcal{F}_s$ that "tend to persist," i.e., which are assumed to keep their value until the contrary is explicitly stated (and, hence, to which the persistence assumption should apply), and so-called *momentary* fluents $\mathcal{F}_m$ that "tend to disappear" [Lifschitz and Rabinov, 1989]. For instance, shooting with a previously loaded gun causes a bang which, however, does not persist and abates immediately. As an important subclass of momentary fluents we have the action fluents $\mathcal{F}_a$. Altogether, the set of fluents $\mathcal{F}$ describing a dynamic system consists essentially of three components $(\mathcal{F}_s, \mathcal{F}_m, \mathcal{F}_a)$ where $\mathcal{F}_a \subseteq \mathcal{F}_m$ and $\mathcal{F}_s \cap \mathcal{F}_m = \emptyset$.

Based on this sophistication, the persistence principle is integrated into our framework by defining that, for each state s, the successor state $\Phi(s)$ is specified via an associated triple of sets of fluents $\langle sf^-, sf^+, mf^+ \rangle$. Here, sf^- contains the static fluents which change their truth value to false during the state transition, i.e., which are removed from s; sf^+ contains the static fluents which change their truth value to true, i.e., which are added to s; and mf^+ contains all momentary fluents which are true in $\Phi(s)$. All other static fluents in s continue to be (and no other static fluents become) element of $\Phi(s)$ while all momentary fluents except those in mf^+ shall not be contained in the resulting state.

Example 2 Consider an extension of the Yale Shooting domain with static fluents $\mathcal{F}_s = \{alive, loaded\}$ and momentary fluents $\mathcal{F}_m = \{bang, bullet, load, shoot\}$, where the additional fluents *bang* and *bullet* describe, respectively, the temporary acoustical occurrence of a shot and a flying bullet. We then might specify the following:

	State s	:	$\langle sf^-, sf^+, mf^+ \rangle$
(a)	$\{alive, loaded, shoot\}$	:	$\langle \{loaded\}, \emptyset, \{bang, bullet\} \rangle$
(b)	$\{alive, bang, bullet\}$	:	$\langle \{alive\}, \emptyset, \emptyset \rangle$

In words, shooting with a previously loaded gun causes the gun to become unloaded ($loaded \in sf^-_{(a)}$) and the occurrence of two events, *bang* and *bullet*; and the flying bullet is intended to hit the turkey and, hence, causes it to drop dead ($alive \in sf^-_{(b)}$) during the following state transition. This example illustrates how our paradigm allows for a natural formalization of delayed effects (here, the victim's death as a final result of having shot with the gun). Using this specification, we obtain, for instance, $\Phi(\Phi(\{alive, loaded, shoot\})) = \Phi(\{alive, bang, bullet\}) = \emptyset$. Note that finally both fluents *bang* and *bullet* disappear automatically because they are momentary. ■

Although the persistence assumption allows for a compaction of defining a successor state $\Phi(s)$ by providing instructions to computing it, the formalization above still requires an exhaustive description as regards the space of states s. Therefore, the second major principle of specifying the behavior of a dynamic system consists in splitting the definition of a single state transition into separate *atomic* laws of causality, which then are usually applicable in multiple states. This is especially essential in theories which involve concurrency since it enables one to specify the effects of each single action (or event like *bullet*) separately:

Definition 3 Let $\mathcal{F} = \mathcal{F}_s \dot\cup \mathcal{F}_m$ be a set of static and momentary fluents. A structure $c : \langle sf^-, sf^+, mf^+ \rangle$ is called a *causal law* if $c \subseteq \mathcal{F}$, called the *condition*; $sf^-, sf^+ \subseteq \mathcal{F}_s$; and $mf^+ \subseteq \mathcal{F}_m$. ■

A causal law is applicable in a state whenever its condition is contained in the state description.[4] In what follows, to select the four components of some causal law $\ell = c : \langle sf^-, sf^+, mf^+ \rangle$, we use the four functions $cond(\ell) \stackrel{\text{def}}{=} c$, $static^-(\ell) \stackrel{\text{def}}{=} sf^-$, $static^+(\ell) \stackrel{\text{def}}{=} sf^+$ and $moment^+(\ell) \stackrel{\text{def}}{=} mf^+$. For convenience, we furthermore use the following abbreviation to describe the result of applying a set of causal laws $\mathcal{L}$ to some state s:

$$Trans(\mathcal{L}, s) \stackrel{\text{def}}{=} \big((s \setminus \textstyle\bigcup_{\ell \in \mathcal{L}} static^-(\ell)) \setminus \mathcal{F}_m \big) \cup \textstyle\bigcup_{\ell \in \mathcal{L}} static^+(\ell) \cup \textstyle\bigcup_{\ell \in \mathcal{L}} moment^+(\ell)$$

where $\mathcal{F}_m$ denotes the set of momentary fluents considered in the dynamic system at hand. Hence, we first remove from s all static fluents that are supposed to become false by some causal law in $\mathcal{L}$; afterwards, all momentary fluents are removed; and finally, all static and all momentary fluents are added that are supposed to become true by some law in $\mathcal{L}$.

Example 3 Consider the two causal laws

	Condition	:	$\langle sf^-, sf^+, mf^+ \rangle$	
ℓ_1	$\{shoot, loaded\}$	:	$\langle \{loaded\}, \emptyset, \{bang, bullet\} \rangle$	(1)
ℓ_2	$\{bullet, alive\}$	:	$\langle \{alive\}, \emptyset, \emptyset \rangle$	

whose conditions both are contained, hence satisfied, in the state $s = \{alive, loaded, bullet, shoot\}$ (where *bullet* might result from a previous shot). We obtain $Trans(\{\ell_1, \ell_2\}, s) = \{bang, bullet\}$. ■

[4] It is for the sake of simplicity why we have restricted the condition of a causal law to a set of fluents and defined applicability as validity of the conjunction of these fluents in the state at hand. It is however natural and straightforward to consider arbitrary fluent formulae (c.f. Definition 2) instead.

It is of course important to take into account the possibility that the simultaneous occurrence of two or more actions (or events) might have different effects than their separate occurrence. As an example, consider a table with a glass of water on it. Lifting the table on any side causes the water to be spilled whereas nothing similar happens if it is lifted simultaneously on opposite sides. In terms of our theory, we can specify this situation by three causal laws, viz

	Condition	:	$\langle sf^-, sf^+, mf^+ \rangle$
ℓ_1	$\{$*lift-left*$\}$	:	$\langle \emptyset, \emptyset, \{$*water-spills*$\}\rangle$
ℓ_2	$\{$*lift-right*$\}$	:	$\langle \emptyset, \emptyset, \{$*water-spills*$\}\rangle$
ℓ_3	$\{$*lift-left*, *lift-right*$\}$	:	$\langle \emptyset, \emptyset, \emptyset \rangle$

(2)

where *lift-left* and *lift-right* both are action fluents and *water-spills* too is a momentary fluent. Unfortunately, however, each law is applicable in the state $s = \{\textit{lift-left}, \textit{lift-right}\}$, thus determining the unintended result $\textit{Trans}(\{\ell_1, \ell_2, \ell_3\}, s) = \{\textit{water-spills}\}$.

In order to avoid this kind of counterintuitive behavior, we employ an additional criterion to suppress the application of some causal law as soon as, roughly spoken, more specific information is available (see also [Baral and Gelfond, 1993; Hölldobler and Thielscher, 1995]). For instance, Law ℓ_3 in (2) should override ℓ_1 and ℓ_2 whenever it is applicable. Formally, we introduce the following partial ordering on causal laws:

Definition 4 Let ℓ_1, ℓ_2 be two causal laws then ℓ_1 is called *more specific* than ℓ_2, written $\ell_1 \prec \ell_2$, iff $\textit{cond}(\ell_1) \supset \textit{cond}(\ell_2)$. ■

E.g., $\ell_3 \prec \ell_1$ and $\ell_3 \prec \ell_2$ but neither $\ell_1 \prec \ell_2$ nor $\ell_2 \prec \ell_1$ in (2).[5]

Based on the specificity criterion, the causal model of a dynamic system is obtained from a set of causal laws as follows:

Definition 5 Let $\mathcal{F}$ be a set of fluents and $\mathcal{L}$ a set of causal laws. For each (consistent) state $s \subseteq \mathcal{F}$ let $\mathcal{L}(s)$ denote the set

$$\{\, \ell \in \mathcal{L} \mid \textit{cond}(\ell) \subseteq s \ \&\ \neg\exists \ell' \in \mathcal{L}.\ \ell' \prec \ell \ \&\ \textit{cond}(\ell') \subseteq s \,\}.$$

Then, $\Phi(s) := \textit{Trans}(\mathcal{L}(s), s)$. ■

In words, $\mathcal{L}(s)$ contains each causal law $\ell \in \mathcal{L}$ which is applicable in s (i.e., $\textit{cond}(\ell) \subseteq s$) unless there is a more specific law $\ell' \in \mathcal{L}$ that is also applicable (i.e., $\ell' \prec \ell$ and $\textit{cond}(\ell') \subseteq s$).

For instance, since the first two causal laws in (2) are less specific than the third one, we now obtain—due to $\mathcal{L}(\{\textit{lift-left}, \textit{lift-right}\}) = \{\ell_3\}$—the successor state $\Phi(\{\textit{lift-left}, \textit{lift-right}\}) = \emptyset$ as intended. On the other hand, we still obtain, say, $\Phi(\{\textit{lift-left}\}) = \{\textit{water-spills}\}$ because Law ℓ_3, though more specific than ℓ_1, is not applicable in this case.

[5]If we allow arbitrary fluent formulae as conditions of causal laws (c.f. Footnote 4), then a law with condition c_1 is said to be more specific than a law with condition c_2 iff $\forall s.\, s \models c_1 \rightarrow c_2$ and $\exists s.\, s \models \neg(c_2 \rightarrow c_1)$. In the propositional case, this is obviously still decidable and a corresponding dependence graph could be computed in a pre-processing step.

One should be aware of the fact that nonetheless it might well happen that two most specific applicable laws have mutually exclusive effects. A reasonable way to handle this problem will be proposed and formalized below, in Section 5. For the moment we assume that the combination of most specific causal laws never leads to contradictory fluent values, i.e., more formally, that

$$\bigcup_{\ell \in \mathcal{L}(s)} \textit{static}^-(\ell) \ \cap \bigcup_{\ell \in \mathcal{L}(s)} \textit{static}^+(\ell) \ = \ \emptyset \qquad (3)$$

for each (consistent) state s (where $\mathcal{L}(s)$ is as in Definition 5).

4 A Model-Based Semantics

Based on the specification of state transition in a dynamic system, we can define its behavior over a longer period and under the influence of one or more agents. To direct the development of a system, these agents are able to (simultaneously) execute actions. The execution of one or more actions in a particular state is modeled by adding the corresponding set of action fluents to the state descriptions before applying the transition function. Since we take action fluents as momentary, these are usually removed during a state transition. The following definition formalizes this concept and extends it to the application of sequences of action sets:

Definition 6 Let $(\mathcal{F}, \Phi)$ be a dynamic system with action fluents $\mathcal{F}_a \subset \mathcal{F}$ and $p = [a_1, \ldots, a_n]$ ($n \geq 0$) be a sequence of sets of action fluents (i.e., $a_i \subseteq \mathcal{F}_a$). Furthermore, let s_0 be a consistent state, then the *application* of p to s_0 yields an infinite sequence of system states $\langle s_1, \ldots, s_n, s_{n+1}, \ldots \rangle$ where

- $s_1 = s_0 \cup a_1$;
- $s_{i+1} = \Phi(s_i) \cup a_{i+1}$, for each $1 \leq i < n$; and
- $s_{i+1} = \Phi(s_i)$ for each $i \geq n$

provided each state $s_1, \ldots, s_n$ is consistent—otherwise the application of p to s_0 is undefined. If it is defined then the triple $(p, s_0, \langle s_1, \ldots \rangle)$ is a *development* in $(\mathcal{F}, \Phi)$. ■

Note that some sets of actions a_i might be empty, i.e., an agent has the possibility to pause for a moment and let the system act autonomously. Note further that, after having executed the entire sequence of actions, the resulting state is not necessarily stable, i.e., the system might run into a limit cycle by oscillating among a number of states. Although a transition function Φ should be designed such that no inconsistent state results from a consistent one (c.f. Definition 1), the process of adding action fluents may cause inconsistency of some state s_i. This is the reason for the additional consistency requirement above.

Example 4 Consider the Yale Shooting domain in the formalization of Example 2 along with the causal model determined by the two causal laws in (1). The application of the sequence $[\emptyset, \{\textit{shoot}\}]$ to the initial state $s_0 = \{\textit{bang}, \textit{alive}, \textit{loaded}\}$ yields

$$\begin{aligned} s_1 &= s_0 \cup \emptyset = \{\textit{bang}, \textit{alive}, \textit{loaded}\} \\ s_2 &= \Phi(s_1) \cup \{\textit{shoot}\} = \{\textit{alive}, \textit{loaded}, \textit{shoot}\} \end{aligned}$$

$$\begin{array}{lllll} s_3 & = & \Phi(s_2) & = & \{alive, bang, bullet\} \\ s_4 & = & \Phi(s_3) & = & \emptyset \\ s_5 & = & \Phi(s_4) & = & \emptyset \\ & \vdots & & & \end{array}$$

■

In the course of the development of a system, we can make observations concerning its various states. An observation can be formulated as a fluent formula associated with a particular time point. We then call a formal development in the sense of Definition 6 a *model* of an observation iff the corresponding fluent formula is true in the corresponding state of the development:

Definition 7 Let $(\mathcal{F}, \Phi)$ be a dynamic system. An expression $[i]\psi$ is called an *observation* if $i \in \mathbb{N}_0$ and ψ is a fluent formula. Such an observation *holds* in a development $(p, s_0, \langle s_1, \ldots \rangle)$ iff $s_i \models \psi$. A *model* of a set Ψ of observations is a development in $(\mathcal{F}, \Phi)$ where each element of Ψ holds. ■

For instance, $[0]alive \wedge \neg bullet$ and $[3]\neg alive$ are two observations that can be formulated in our Yale Shooting domain. Then, this development is a model wrt the causal laws in (1):

$$(\,[\{shoot\}]\,,\,\{alive, loaded\}\,,\\ \langle\{alive, loaded, shoot\}, \{alive, bang, bullet\}, \emptyset, \emptyset, \ldots\rangle\,)$$

since we have $s_0 = \{alive, loaded\} \models alive \wedge \neg bullet$ and $s_3 = \emptyset \models \neg alive$. The reader is invited to verify that not only in this development but in every model of the two observations above the additional observation $[1]shoot \wedge loaded$ holds—hence, we are allowed to conclude that a *shoot* action must have taken place and that the gun was necessarily loaded at the beginning.[6]

In general, we define the following notion of entailment on the space of observations:

Definition 8 Let $(\mathcal{F}, \Phi)$ be a dynamic system and Ψ a set of observations. Ψ *entails* an additional observation $[i]\psi$, written $\Psi \models_\Phi [i]\psi$, iff $[i]\psi$ holds in each model of Ψ. ■

Based on this model-theoretic formalization of dynamic systems, we can classify some important and well-known problem categories as instances given a system $(\mathcal{F}, \Phi)$, depending on what information is provided:

- A *temporal projection* problem consists of an initial state s_0 along with a sequence of sets of actions $p = [a_1, \ldots, a_n]$. The question is to compute the resulting state after having applied p to s_0.

  In terms of our theory, the problem is essentially to find a model for the particular set of observations that describes the given initial state and exactly those occurrences of action fluents which are determined by p, i.e.,

$$\begin{array}{rcl} \Psi & = & \{\ [0] \bigwedge_{f \in s_0} f \wedge \bigwedge_{f \in \mathcal{F} \setminus s_0} \neg f\,, \\ & & \quad [1] \bigwedge_{f \in a_1} f \wedge \bigwedge_{f \in \mathcal{F}_a \setminus a_1} \neg f\,, \\ & & \qquad \vdots \\ & & \quad [n] \bigwedge_{f \in a_n} f \wedge \bigwedge_{f \in \mathcal{F}_a \setminus a_n} \neg f\ \} \end{array}$$

  where $\mathcal{F}_a$ denotes the underlying action fluents.

- A *classical planning* problem consists of an initial state s_0 and a fluent formula g, called the *goal*. The question is to find a sequence of sets of actions p whose application to s_0 yields a sequence of system states containing one particular state s_n which satisfies g.

  In terms of our theory, the problem is essentially to find a model for this set of observations:

$$\begin{array}{rcl} \Psi & = & \{\ [0] \bigwedge_{f \in s_0} f \wedge \bigwedge_{f \in \mathcal{F} \setminus s_0} \neg f\,, \\ & & \quad [n] g\ \} \end{array}$$

  for some $n \in \mathbb{N}$.

- A *postdiction* problem consists of a narrative given by a set of observations Ψ along with a sequence of sets of actions $p = [a_1, \ldots, a_n]$. The question is to decide whether an additional observation is a logical consequence of this scenario.

  In terms of our theory, the problem is essentially to decide entailment wrt the particular set of observations that includes the given ones and describes exactly those occurrences of action fluents which are determined by p, i.e.,

$$\begin{array}{rl} \Psi \cup \{ & [1] \bigwedge_{f \in a_1} f \wedge \bigwedge_{f \in \mathcal{F}_a \setminus a_1} \neg f\,, \\ & \qquad \vdots \\ & [n] \bigwedge_{f \in a_n} f \wedge \bigwedge_{f \in \mathcal{F}_a \setminus a_n} \neg f\ \}\,. \end{array}$$

Due to the fact that our theory generalizes several formal approaches to reasoning about actions, we are able to formulate, for instance, more general planning problems where the initial situation is only partially defined and where the goal specification is not necessarily required to hold in a single state. Or, we can formulate general postdiction problems where the sequence of actions is only incompletely specified, etc.

5 Nondeterminism

In this section, we extend the concepts developed so far to so-called *nondeterministic* dynamic systems. Nondeterminism occurs when uncertainty about the successor state exists even in case the current state is completely known. This is reflected in the following definition, where the causal model consists of a relation on

[6] One referee raised the question whether one should in general assume minimality of actions being performed when trying to find models for a given set of observations. Although this assumption might be helpful (if taken as preference ordering) e.g. in case of planning problems, we definitely do not want to employ this criterion as a global restriction on the notion of a model. If, for instance, we observe the gun being unloaded at the beginning and being loaded after a while, we can conclude safely only that at least one load action has eventually taken place; we ought not to disregard the possibility that the gun has been loaded and shot with several times before, because there is nothing illogical about this. Besides, our definition guarantees the property of *restricted monotonicity* [Lifschitz, 1993], which means that additional observations can never force the revision of previous conclusions.

pairs of states (see, e.g., [Thielscher, 1994]) instead of a function as in Definition 1:

Definition 9 A *nondeterministic, propositional dynamic system* is a pair $(\mathcal{F}, \Phi)$ consisting of a set of fluents $\mathcal{F}$ and a relation $\Phi \subseteq 2^{\mathcal{F}} \times 2^{\mathcal{F}}$. ■

Given a state $s \subseteq \mathcal{F}$, each s' such that $(s, s') \in \Phi$ is called a *possible successor* state. A state is now said to be inconsistent in case it has no successor at all.

The concept of nondeterminism is reflected in an extended notion of a causal law where several expressions $\langle sf_i^-, sf_i^+, mf_i^+ \rangle$ can be associated with a single condition; each triple then determines a possible alternative:

Definition 10 Let $\mathcal{F} = \mathcal{F}_s \dot{\cup} \mathcal{F}_m$ be a set of static and momentary fluents. An *extended causal law* is a structure $c : \{\langle sf_1^-, sf_1^+, mf_1^+ \rangle, \ldots, \langle sf_n^-, sf_n^+, mf_n^+ \rangle\}$ where $n \geq 1$; $c \subseteq \mathcal{F}$; $sf_i^-, sf_i^+ \subseteq \mathcal{F}_s$; and $mf_i^+ \subseteq \mathcal{F}_m$ ($1 \leq i \leq n$). ■

Example 5 The Russian Turkey scenario (see, e.g., [Sandewall, 1994]) is obtained from the Yale Shooting domain by adding an action fluent *spin*. The effect of spinning its cylinder is that the firearm becomes randomly loaded or not, regardless of its state before. This nondeterministic effect can be modeled by the following extended causal law:

$$\begin{array}{lcl} \text{Condition} & : & \{\langle sf_i^-, sf_i^+, mf_i^+ \rangle\} \\ \hline\hline \{spin\} & : & \left\{ \begin{array}{l} \langle \{loaded\}, \emptyset, \emptyset \rangle, \\ \langle \emptyset, \{loaded\}, \emptyset \rangle \end{array} \right\} \end{array} \tag{4}$$

■

As for the special case of deterministic systems, the combination of all most specific laws shall determine the behavior of the system at hand. Hence, for each state s we define the set $\mathcal{L}(s)$ as

$$\{ \ell \in \mathcal{L} \mid cond(\ell) \subseteq s \ \& \ \neg \exists \ell' \in \mathcal{L}.\ \ell' \prec \ell \ \& \ cond(\ell') \subseteq s \},$$

similar to Definition 5, where $\mathcal{L}$ denotes the underlying set of (extended) causal laws. Now, let $\mathcal{L}(s)$ be the set $\{c_1 : \mathcal{A}_1, \ldots, c_k : \mathcal{A}_k\}$ ($k \geq 0$) and define $Poss(\mathcal{L}(s))$ as

$$\left\{ \{c_1 : a_1, \ldots, c_k : a_k\} \;\middle|\; a_i \in \mathcal{A}_i \, (1 \leq i \leq k) \right\}$$

containing each possible selection and combination of alternatives. Each element in $Poss(\mathcal{L}(s))$ determines a possible successor state of s, i.e.,

$$(s, s') \in \Phi \quad \text{iff} \quad \exists \mathcal{P} \in Poss(\mathcal{L}(s)).\ s' = Trans(\mathcal{P}, s).$$

For instance, consider (4) as the only applicable causal law in the state $s = \{alive, spin\}$. Then, $Poss(\mathcal{L}(s))$ is

$$\left\{ \{spin : \langle \{loaded\}, \emptyset, \emptyset \rangle\}, \{spin : \langle \emptyset, \{loaded\}, \emptyset \rangle\} \right\}$$

hence, $(s, \{alive\}) \in \Phi$ and $(s, \{alive, loaded\}) \in \Phi$.

The concept of nondeterminism provides us with an interesting solution to the problem of concurrently executed actions with mutually exclusive effects. Consider, for instance, the two causal laws

$$\begin{array}{lcl} \text{Condition} & : & \langle sf^-, sf^+, mf^+ \rangle \\ \hline\hline \{push\text{-}door\} & : & \langle \emptyset, \{open\}, \emptyset \rangle \\ \hline \{pull\text{-}door\} & : & \langle \{open\}, \emptyset, \emptyset \rangle \end{array} \tag{5}$$

where *push-door* and *pull-door* denote action fluents while the static fluent *open* describes the state of the door under consideration here. Now, assume three agents acting concurrently: The first one tries to push the door, the second one tries to pull it, and the third agent intends to lift the left hand side of a table inside the room (c.f. (2)). Assume further that the door is closed at the moment and no water spills out of the glass situated on the table, then this situation can be expressed by the state $s = \{push\text{-}door, pull\text{-}door, lift\text{-}left\}$. Now, aside from ℓ_1 in (2) both causal laws in (5) are applicable. However, the first one requires the door to be open in the succeeding state ($open \in sf^+$) while the second one requires the contrary ($open \in sf^-$). Hence, our consistency condition, (3), is not satisfied here.

Most classical AI formalizations of concurrent actions, such as [Lin and Shoham, 1992; Baral and Gelfond, 1993; Große, 1994], declare situations like s inconsistent and, hence, do not allow any conclusions whatsoever about the successor state. Indeed it is impossible that both actions *push-door* and *pull-door* are successful. However, in [Bornscheuer and Thielscher, 1994] we argue (in the context of a theory developed in [Gelfond and Lifschitz, 1993; Baral and Gelfond, 1993]) that it is reasonable to draw some conclusions at least about uninvolved fluents; e.g., we would like to conclude that the third agent is successful in lifting the table, which causes the water to be spilled out. Preventing global inconsistency in case of local conflicts is the basic intention in this idea.

The notion of nondeterminism provides us with the possibility to draw conclusions like the one just mentioned. Instead of declaring the successor state of s as completely undefined, we take only the disputed fluent(s) (here: *open*) as uncertain while any other effect (here: *water-spills*, coming from (2)) occurs as intended. In our example, we then obtain two possible successor states of s, namely, $\{open, water\text{-}spills\}$ and $\{water\text{-}spills\}$—providing us with the conclusion that *water-spills* is an obligatory effect of s.

This strategy of conflict solving is integrated in the following definition of how to obtain the causal model in case of nondeterministic systems:

Definition 11 Let $\mathcal{F}$ be a set of fluents and $\mathcal{L}$ a set of (extended) causal laws. For each (consistent) state $s \subseteq \mathcal{F}$ let $\mathcal{L}(s)$ denote the set

$$\{ \ell \in \mathcal{L} \mid cond(\ell) \subseteq s \ \& \ \neg \exists \ell' \in \mathcal{L}.\ \ell' \prec \ell \ \& \ cond(\ell') \subseteq s \}.$$

Now, if $\mathcal{L}(s) = \{c_1 : \mathcal{A}_1, \ldots, c_k : \mathcal{A}_k\}$ ($k \geq 0$) then let $Poss(\mathcal{L}(s))$ be the set

$$\left\{ \{c_1 : a_1, \ldots, c_k : a_k\} \;\middle|\; a_i \in \mathcal{A}_i \, (1 \leq i \leq k) \right\}$$

and define $(s, s') \in \Phi$ iff

$$\exists \mathcal{P} \in Poss(\mathcal{L}(s)),\ sf^{\lightning} \in Confl(\mathcal{P}).\ s' = Trans(\mathcal{P}, s) \setminus sf^{\lightning}$$

where

$$Confl(\mathcal{P}) \ := \ \bigcup_{\ell \in \mathcal{P}} static^-(\ell) \ \cap \ \bigcup_{\ell \in \mathcal{P}} static^+(\ell).$$

■

The set $Confl(\mathcal{P})$ is intended to contain all disputed fluents (c.f. (3)), and each possible combination of these fluents determines a possible successor state.[7]

Finally, the semantics developed in the previous section is extended to nondeterministic dynamic systems in the following way (c.f. Definition 6):

Definition 12 Let $(\mathcal{F}, \Phi)$ be a nondeterministic dynamic system with action fluents $\mathcal{F}_a \subset \mathcal{F}$, and let $p = [a_1, \ldots, a_n]$ ($n \geq 0$) be a sequence of sets of action fluents (i.e., $a_i \subseteq \mathcal{F}_a$). Furthermore, let s_0 be a consistent state, then a triple $(p, s_0, \langle s_1, \ldots, s_n, s_{n+1}, \ldots \rangle)$ is a *development* iff

- $s_1 = s_0 \cup a_1$;
- $s_{i+1} = s'_i \cup a_{i+1}$, where $(s_i, s'_i) \in \Phi$, for each $1 \leq i < n$;
- $(s_i, s_{i+1}) \in \Phi$ for each $i \geq n$

and each state $s_1, \ldots, s_n$ is consistent. ■

6 Summary and Outlook

We have presented a formal theory of dynamic systems that is based on a different paradigm compared to standard approaches such as [Gelfond and Lifschitz, 1993; Sandewall, 1994]: State transitions naturally occur while time passes by—the system does not necessarily keep stable until agents perform actions. We have illustrated that our framework allows for a natural treatment of concurrent actions and simultaneous events as well as delayed effects. Based on the underlying specification of a dynamic system, we have developed a semantics for observations, i.e., fluent formulae associated with time points. We have investigated both deterministic and nondeterministic systems, and we have integrated a reasonable way to handle the problem of concurrent actions with mutually exclusive effects.

The purpose of this paper was to propose a general and uniform specification language for dynamic systems in conjunction with an intuitive semantics. We have not yet tackled the question of an adequate proof theory. The problem of finding such a calculus and proving its soundness and completeness constitutes the paramount aspect of future work. Our proposal is meant as a challenge: How can existing approaches be adopted to the kind of dynamic systems being specified on the basis of our theory? For instance, how could a variant of the situation calculus [McCarthy and Hayes, 1969] be designed—called, say, *state calculus*—where the common successor state argument $Result(a, s)$ is replaced by $Result(s)$, i.e., which no longer depends on the execution of some action? Other promising directions are the adaption of resource-oriented approaches to reasoning about actions and change [Große *et al.*, 1992], dynamic logic [Harel, 1984], or a first-order encoding following [Elkan, 1992]. Last but not least, though in its current state it covers neither the static-momentary distinction nor specificity or nondeterminism, the modal logic approach [Große, 1994] should be a candidate worth being considered since it already includes the principle of taking actions as part of state descriptions. Conversely, our semantics provides a tool for a formal assessment of its range of applicability, similar to investigations carried out in [Sandewall, 1994].

[7]Note that we are supposed to remove the set $sf^{\lightning}$ from $Trans(\mathcal{P}, s)$ due to the fact that all elements in $Confl(\mathcal{P})$ are first of all added when computing $Trans(\mathcal{P}, s)$.

Acknowledgments

The author thanks Christoph Herrmann for helpful comments on an earlier version of this paper.

References

[Baral and Gelfond, 1993] C. Baral and M. Gelfond. Representing Concurrent Actions in Extended Logic Programming. In R. Bajcsy, ed., *Proc. of IJCAI*, p. 866–871, Chambéry, France, Aug. 1993. Morgan Kaufmann.

[Bornscheuer and Thielscher, 1994] S.-E. Bornscheuer and M. Thielscher. Representing Concurrent Actions and Solving Conflicts. In B. Nebel and L. Dreschler-Fischer, ed.'s, *KI-94: Advances in Artif. Intell.*, vol. 861 of *LNAI*, p. 16–27, Saarbrücken, Germany, Sep. 1994. Springer.

[Elkan, 1992] C. Elkan. Reasoning about Action in First-Order Logic. In *Proc. of the Conf. of the Canadian Society for Computational Studies of Intell.*, Vanvouver, Canada, May 1992. Morgan Kaufmann.

[Gelfond and Lifschitz, 1993] M. Gelfond and V. Lifschitz. Representing Action and Change by Logic Programs. *J. of Logic Programming*, 17:301–321, 1993.

[Große *et al.*, 1992] G. Große, S. Hölldobler, J. Schneeberger, U. Sigmund, and M. Thielscher. Equational Logic Programming, Actions, and Change. In K. Apt, ed., *Proc. of IJCSLP*, p. 177–191, Washington, 1992. MIT Press.

[Große, 1994] G. Große. Propositional State-Event Logic. In C. MacNish, D. Peirce, and L. M. Peireira, ed.'s, *Proc. of the European Workshop on Logics in AI (JELIA)*, vol. 838 of *LNAI*, p. 316–331. Springer, 1994.

[Harel, 1984] D. Harel. Dynamic logic. In D. Gabbay and F. Guenther, ed.'s, *Handbook of Philosophical Logic*, vol. II, p. 497–604. D. Reidel Publishing Company, 1984.

[Hölldobler and Thielscher, 1995] S. Hölldobler and M. Thielscher. Computing Change and Specificity with Equational Logic Programs. *Annals of Mathematics and Artificial Intelligence*, 1995.

[Lifschitz and Rabinov, 1989] V. Lifschitz and A. Rabinov. Things That Change by Themselves. In *Proc. of IJCAI*, p. 864–867, Detroit, MI, 1989.

[Lifschitz, 1993] V. Lifschitz. Restricted monotonicity. In *Proc. of AAAI*, Washington, DC, July 1993. MIT Press.

[Lin and Shoham, 1992] F. Lin and Y. Shoham. Concurrent Actions in the Situation Calculus. In *Proc. of AAAI*, p. 590–595, San Jose, CA, 1992. MIT Press.

[McCarthy and Hayes, 1969] J. McCarthy and P. J. Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence. *Machine Intell.*, 4:463–502, 1969.

[Pollack, 1992] M. E. Pollack. The uses of plans. *Artif. Intell.*, 57:43–68, 1992.

[Sandewall, 1994] E. Sandewall. *Features and Fluents.* Oxford University Press, 1994.

[Thielscher, 1994] M. Thielscher. Representing Actions in Equational Logic Programming. In P. Van Hentenryck, ed., *Proc. of ICLP*, p. 207–224, Santa Margherita Ligure, Italy, June 1994. MIT Press.