

# On the Logic of Dynamic Systems

Michael Thielscher

FG Intellektik, TH Darmstadt

Alexanderstr. 10, D-64283 Darmstadt (Germany)

E-mail: mit@intellektik.informatik.th-darmstadt.de

From: AAAI Technical Report SS-95-07. Compilation copyright © 1995, AAAI (www.aaai.org). All rights reserved.

A common assumption underlying most formal theories of actions, such as [Sandewall, 1992; Gelfond and Lifschitz, 1993; Kartha and Lifschitz, 1994; Thielscher, 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 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.<sup>1</sup>

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 naturally provides two main characteristic features: First of all, the notion of parallelism, which includes the concurrent execution of actions as well as the simultaneous occurrence of events, is an intrinsic element of our theory. Secondly, since parallelism means that normally the various changes during a state transition are simultaneously caused by several reasons, we can easily model delayed effects of actions by initiating additional independent events which eventually trigger a particular effect.

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: the *persistence assumption*<sup>2</sup>, which enables one to state explicitly only the things that change during a single state transition while everything else is implicitly assumed to remain as it is. Secondly, *atomic causal laws* are used to state relationships between single cause-effect-pairs. Then, usually several of these atomic laws apply to the current state of a dynamic system so that a combination of laws determines the complete transition step. The use of atomic causal relationships is especially necessary in theories which in-

volve 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.
- A *planning* problem is the question 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 general planning problems where, in contrast to the classical AI definition of planning, the properties we strive for can 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<sup>3</sup>, one is faced with 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 [Sandewall, 1992; Gelfond and Lifschitz, 1993] in so far as the sequence of actions performed during the period in question can as well contain uncertainties so that the given observations may also give rise to additional knowledge about this sequence.

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.

## Specifying Dynamic Systems

A formal specification of a dynamic system consists of these two components: 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. The second component describes the behavior of the system as regards state transitions. In this and the next two sections, we focus on deterministic dynamic systems.

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

<sup>2</sup>also called *frame assumption* or *inertia principle*

<sup>3</sup>called *chronicle completion* in [Sandewall, 1992]

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*, on 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 concerning single fluents and states, we can construct (propositional) formulae and define a corresponding entailment relation following the standard way:

**Definition 2** Let  $\mathcal{F}$  be a set of fluents. The set of *fluent formulae* (over  $\mathcal{F}$ ) 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 defined as follows:

- $s \models f$  iff  $f \in s$  where  $f \in \mathcal{F}$ .
- $s \models \neg F$  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.

**Example 1** To formalize a basic version of the Yale Shooting domain [Hanks and McDermott, 1987], 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  $\{alive, load\} \not\models shoot$ . ■

## Causal Laws

The main challenge when specifying the causal model of a dynamic system is the problem of finding a compact representation of the corresponding causal model  $\Phi$ . The most fundamental concept related to this is the *persistence* principle which enables us to only specify the fluents that change their value during a particular state transition while all other fluents are implicitly taken to keep their value [McCarthy, 1963; McCarthy and Hayes, 1969; Ford and Hayes, 1991; Bibel and Thielscher, 1994].

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 category 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 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 fluents  $(\mathcal{F}_s = \{alive, loaded\}, \mathcal{F}_m = \{bang, bullet, load, shoot\}, \mathcal{F}_a = \{load, shoot\})$ , where the new momentary fluents *bang* and *bullet* describe, respectively, the temporary acoustical occurrence of a shot and a flying bullet. We then might have the following extract of the causal model specification:

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

In words, shooting with a previously loaded gun causes the gun to become unloaded ( $loaded \in sf^-$ ) 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 ( $alive \in sf^-$ ) 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 compactification of defining a successor state  $\Phi(s)$  by providing an instruction how to compute it, the formalization above still requires an exhaustive description as regards the space of states  $s$ . Therefore, the second major principle towards a compact specification of 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 necessary 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  $mf^+ \subseteq \mathcal{F}_m$ . ■

Hence, as before a causal law consists of three sets of fluents specifying the desired effects. A law is applicable in a state whenever its condition is contained in the state description.<sup>4</sup> In what follows, to select the four components of some causal law  $\ell = c : \langle sf^-, sf^+, mf^+ \rangle$ , we use the four functions  $cond(\ell) := c$ ,  $static^-(\ell) := sf^-$ ,  $static^+(\ell) := sf^+$  and  $moment^+(\ell) := mf^+$ .

For convenience, we furthermore use the following abbreviation to describe the result of applying a set of causal laws  $\mathcal{L}$  to some state  $s$ :

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

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 that are supposed to become true by some law in  $\mathcal{L}$  are added.

**Example 3** Consider the two causal laws

	Condition	:	$\langle sf^-, sf^+, mf^+ \rangle$
$\ell_1$	$\{shoot, loaded\}$	:	$\langle \{loaded\}, \emptyset, \{bang, bullet\} \rangle$
$\ell_2$	$\{bullet, alive\}$	:	$\langle \{alive\}, \emptyset, \emptyset \rangle$

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

<sup>4</sup>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 be aware of 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, namely

	Condition	:	$\langle sf^-, sf^+, mf^+ \rangle$
$\ell_1$	$\{lift-left\}$	:	$\langle \emptyset, \emptyset, \{water-spills\} \rangle$
$\ell_2$	$\{lift-right\}$	:	$\langle \emptyset, \emptyset, \{water-spills\} \rangle$
$\ell_3$	$\{lift-left, lift-right\}$	:	$\langle \emptyset, \emptyset, \emptyset \rangle$

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

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, 1993]). For instance, Law  $\ell_3$  in (2) should override  $\ell_1$  and  $\ell_2$  whenever it is applicable. Formally, we introduce the following partial ordering on causal laws:

**Definition 4** A causal law  $\ell_1$  is *more specific* than a causal law  $\ell_2$ , written  $\ell_1 \prec \ell_2$ , iff  $cond(\ell_1) \supset cond(\ell_2)$ . ■

For instance,  $\ell_3 \prec \ell_1$  and  $\ell_3 \prec \ell_2$  but neither  $\ell_1 \prec \ell_2$  nor  $\ell_2 \prec \ell_1$  in (2).<sup>5</sup>

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  $s$  over  $\mathcal{F}$  let  $\mathcal{L}(s)$  denote the set

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

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

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

For instance, since the first two causal laws in (2) are less specific than the third one, we now obtain, due to  $\mathcal{L}(\{lift-left, lift-right\}) = \{\ell_3\}$ , the successor state  $\Phi(\{lift-left, lift-right\}) = \emptyset$  as intended. On the other hand, we still obtain, say,  $\Phi(\{lift-left\}) = \{water-spills\}$

<sup>5</sup>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.

because Law  $\ell_3$ , though more specific than  $\ell_1$ , is not applicable in this case.

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. For the moment, we assume that the combination of most specific causal laws never leads to contradictory fluent values, i.e., more formally, that

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

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

## 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 sets of actions:

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

Note that some sets of actions  $a_i$  might be empty, i.e. the 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  $\Phi$  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 some state  $s_i$  to become inconsistent. 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  $[\emptyset, \{\text{shoot}\}]$  to the initial state  $s_0 = \{\text{bang, alive, loaded}\}$  yields

$$s_1 = s_0 \cup \emptyset = \{\text{bang, alive, loaded}\}$$

$$\begin{aligned} s_2 &= \Phi(s_1) \cup \{\text{shoot}\} = \{\text{alive, loaded, shoot}\} \\ s_3 &= \Phi(s_2) = \{\text{alive, bang, bullet}\} \\ s_4 &= \Phi(s_3) = \emptyset \\ s_5 &= \Phi(s_4) = \emptyset \\ &\vdots \end{aligned}$$

■

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  $\psi$  is a fluent formula. Such an observation *holds* in a development  $(p, s_0, \langle s_1, \dots \rangle)$  iff  $s_i \models \psi$ . A *model* of a set  $\Psi$  of observations is a development where each element of  $\Psi$  holds. ■

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

$$\begin{aligned} &(\{\text{alive, loaded}\}, \{\{\text{shoot}\}\}, \\ &\quad \langle \{\text{alive, loaded, shoot}\}, \{\text{alive, bang, bullet}\}, \emptyset, \emptyset, \dots \rangle) \end{aligned}$$

due to  $s_0 = \{\text{alive, loaded}\} \models \text{alive} \wedge \neg\text{bullet}$  and  $s_3 = \emptyset \models \neg\text{alive}$ . The reader is invited to verify that not only in this development but in each model of our observations the additional observation  $[1]\text{shoot} \wedge \text{loaded}$  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.

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

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

Based on this model-based theory of dynamic systems, we can classify some important and well-known problem categories as instances, depending on what information is provided:

- A *temporal projection* problem consists of an initial state  $s_0$  and a sequence of sets of actions  $p = [a_1, \dots, a_n]$ . 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 describe the given initial state and exactly those occurrences of action fluents which are determined by  $p$ , i.e.

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

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:

$$\Psi = \{ [0] \bigwedge_{f \in s_0} f \wedge \bigwedge_{f \in \mathcal{F} \setminus s_0} \neg f, \\ [n]g \}$$

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

- A *postdiction* problem consists of a set of observations  $\Psi$  and a sequence of sets of actions  $p = [a_1, \dots, a_n]$ . 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 include the given ones and describe exactly those occurrences of action fluents which are determined by  $p$ , i.e.

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

Due to the fact that our theory generalizes several formal approaches to reasoning about actions, we can, for instance, formulate 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 partially known etc.

### Non-Determinism

In this section, we extend the concepts developed so far to so-called *non-deterministic* dynamic systems. Non-determinism occurs when there is uncertainty about the successor state 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 pairs of states instead of a function as in Definition 1:

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

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

The concept of non-determinism 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  $\mathcal{F} = \mathcal{F}_s \cup \mathcal{F}_m$  be a set of static and momentary fluents. An *extended causal law* is a structure  $c : \{ \langle sf_1^-, sf_1^+, mf_1^+ \rangle, \dots, \langle sf_n^-, sf_n^+, mf_n^+ \rangle \}$  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, 1992]) is obtained from the Yale Shooting domain by adding an action fluent *spin*. The effect of spinning its cylinder is that the gun becomes randomly loaded or not, regardless of its state before. This non-deterministic effect can be modeled by the following extended causal law:

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

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

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

similar to Definition 5, where  $\mathcal{L}$  denotes the underlying set of (extended) causal laws. Then, let  $\mathcal{L}(s)$  be the set  $\{ c_1 : \mathcal{A}_1, \dots, c_k : \mathcal{A}_k \}$  ( $k \geq 0$ ), where each  $\mathcal{A}_i$  is a set of alternative triples. We define  $\text{Poss}(\mathcal{L}(s))$  as

$$\{ \{ c_1 : a_1, \dots, c_k : a_k \} \mid a_i \in \mathcal{A}_i (1 \leq i \leq k) \}$$

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

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

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

$$\{ \{ \text{spin} : \langle \{ \text{loaded} \}, \emptyset, \emptyset \rangle \}, \{ \text{spin} : \langle \emptyset, \{ \text{loaded} \}, \emptyset \rangle \} \},$$

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

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

$$\text{Condition} : \langle sf^-, sf^+, mf^+ \rangle \\ \hline \{ \text{push-door} \} : \langle \emptyset, \{ \text{open} \}, \emptyset \rangle \\ \{ \text{pull-door} \} : \langle \{ \text{open} \}, \emptyset, \emptyset \rangle \quad (5)$$

where *push-door* and *pull-door* denote action fluents and 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 the table inside the room (c.f. (2)). Assume further that the door is closed and no water spills out of the glass situated on the table, then this situation can be expressed by the state  $s = \{ \text{push-door}, \text{pull-door}, \text{lift-left} \}$ . Now, aside from  $\ell_1$  in (2) both causal laws in (5) are applicable. However, the first one requires the door to be open in the succeeding state ( $\text{open} \in sf^+$ ) while the second one requires the contrary ( $\text{open} \in sf^-$ ). 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] treat situations like  $s$  as 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 one still intends to draw at least some conclusions about uninvolved fluents, e.g. concluding that the third agent is successful in lifting the table, which causes the water to be spilled out.

The notion of non-determinism 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  $s$ , viz  $\{open, water-spills\}$  and  $\{water-spills\}$ —providing us with the conclusion that *water-spills* is an obligatory effect of  $s$ .

In general, we obtain the causal model in case of non-deterministic systems as follows:

**Definition 11** Let  $\mathcal{F}$  be a set of fluents and  $\mathcal{L}$  a set of (extended) causal laws. For each state  $s$  over  $\mathcal{F}$  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  $\mathcal{L}(s) = \{c_1 : \mathcal{A}_1, \dots, c_k : \mathcal{A}_k\}$  ( $k \geq 0$ ) then let  $Poss(\mathcal{L}(s))$  be the set

$$\{ \{c_1 : a_1, \dots, c_k : a_k\} \mid a_i \in \mathcal{A}_i \ (1 \leq i \leq k) \}$$

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

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

where

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

The set  $Confl(p)$  is intended to contain all disputed fluents (c.f. (3)), and each possible combination of these fluents determines a possible successor state.<sup>6</sup>

Finally, the semantics developed in the previous section is extended to non-deterministic dynamic systems in the following way:

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

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

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

<sup>6</sup>Note that we are supposed to remove the sets  $sf^{\downarrow}$  from  $Trans(p, s)$  due to the fact that all elements in  $Confl(p)$  are first of all added when computing  $Trans(p, s)$ .

## References

- [Baral and Gelfond, 1993] C. Baral and M. Gelfond. Representing Concurrent Actions in Extended Logic Programming. In R. Bajcsy, ed., *Proc. IJCAI*, p. 866–871, Chambéry, France, 1993. Morgan Kaufmann.
- [Bibel and Thielscher, 1994] W. Bibel and M. Thielscher. Deductive Plan Generation. In S. Arikawa and K. P. Jantke, eds., *Algorithmic Learning Theory—Proc. AII and ALT*, volume 872 of *LNAI*, p. 2–5, Reinhardtsbrunn Castle, Germany, 1994. Springer.
- [Bornscheuer and Thielscher, 1994] S.-E. Bornscheuer and M. Thielscher. Representing Concurrent Actions and Solving Conflicts. In B. Nebel and L. Dreschler-Fischer, eds., *Proc. of the German Annual Conference on Artif. Intell. (KI)*, volume 861 of *LNAI*, p. 16–27, Saarbrücken, Germany, 1994. Springer.
- [Ford and Hayes, 1991] K. Ford and P. Hayes, eds. *Reasoning Agents in a Dynamic World: The Frame Problem*. JAI Press, 1991.
- [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, 1994] G. Große. Propositional State-Event Logic. In C. MacNish, D. Peirce, and L. M. Peireira, eds., *Proc. of the European Workshop on Logics in AI (JELIA)*, volume 838 of *LNAI*, p. 316–331, 1994. Springer.
- [Hanks and McDermott, 1987] S. Hanks and D. McDermott. Nonmonotonic logic and temporal projection. *Artif. Intell.*, 33(3):379–412, 1987.
- [Hölldobler and Thielscher, 1993] S. Hölldobler and M. Thielscher. Actions and Specificity. In D. Miller, ed., *Proc. of the International Logic Programming Symposium (ILPS)*, p. 164–180, Vancouver, 1993. MIT Press.
- [Karthan and Lifschitz, 1994] G. N. Karthana and V. Lifschitz. Actions with Indirect Effects. In J. Doyle, E. Sandewall, and P. Torasso, eds., *Proc. of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, p. 341–350, Bonn, Germany, 1994. Morgan Kaufmann.
- [Lifschitz and Rabinov, 1989] V. Lifschitz and A. Rabinov. Things That Change by Themselves. In *Proc. IJCAI*, p. 864–867, 1989.
- [Lin and Shoham, 1992] F. Lin and Y. Shoham. Concurrent Actions in the Situation Calculus. In *Proc. AAAI*, p. 590–595, San Jose, California, 1992. MIT Press.
- [McCarthy and Hayes, 1969] J. McCarthy and P. J. Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence. *Machine Intelligence*, 4:463–502, 1969.
- [McCarthy, 1963] J. McCarthy. Situations and Actions and Causal Laws. Stanford Artificial Intelligence Project, Memo 2, 1963.
- [Pollack, 1992] M. E. Pollack. The uses of plans. *Artif. Intell.*, 57:43–68, 1992.
- [Sandewall, 1992] E. Sandewall. Features and Fluents. Technical Report LiTH-IDA-R-92-30, Institutionen för datavetenskap, Universitetet Linköping, Sweden, 1992.
- [Thielscher, 1994] M. Thielscher. Representing Actions in Equational Logic Programming. In P. Van Hentenryck, ed., *Proc. of the International Conference on Logic Programming (ICLP)*, 207–224, Santa Margherita Ligure, Italy, 1994. MIT Press.