Causal Links Planning and the Systematic Approach to Action and Change*

Lars Karlsson
Department of Computer and Information Science
Linköping University
S-581 83 Linköping, Sweden
Email: larka@ida.liu.se

Abstract

This paper presents an analysis of partial-order and causal-links planning based on Sandewall's systematic approach to reasoning about action and change. The partial-order planners TWEAK and SNLP are analysed and reconstructed. A temporal logic, called the fluent logic, is used for representing plans, and the strong connection between causal links and elements of the fluent logic are pointed out.

Introduction

The topic of this paper is the formalization of the planning problem and the analysis and specification of planners and the ways they model a changing world, all within Sandewall's *Features and Fluents* framework (Sandewall 1994). The idea behind Sandewall's framework is that problems of reasoning about action and change should not be approached for all possible kinds of worlds (domains) at once. One should instead identify classes of worlds with certain restrictions on their structure, for instance whether actions can occur concurrently and whether actions can be nondeterministic. Specific logics can then be designed for specific classes of worlds. The classification serves as a frame of reference for studying formal properties of logics of action and change. Sandewall presents a number of logics and also proves their soundness and completeness relative to their specific classes.

Observe that in *Features and Fluents*, the systematic approach is applied to logics of action and change. In this paper, the approach is given a wider application. The approach is in fact relevant also for problems that make use of reasoning about action and change, such as planning. The results described in the paper and in (Karlsson 1995) provide concrete support for this. Partial-order and causal-link planning is analysed in this paper. The emphasis is on representation and basic operations applied to this representation. It is shown how classical partial-order plans are specified using the *fluent logic* (Doherty & Lukaszewicz 1994; Doherty 1994b), a logic of action and change. The truth criterion for partial-order plans as described by Chapman (Chapman 1987) for TWEAK is translated to FL, and then the concept of causal links (McAllester & Rosenblitt 1991) is investigated and expressed within the logic; it turns out that there are strong connections to the concept of *occlusion* that is a central component of FL.

The use of logics can be on several levels. First one should make a distinction between representing plans using a logical formalism, and specifying how the planner is controlled. This paper addresses the first case. Second, logics can be used for specification and analysis, and they can be used for implementation. This paper is about specification.

Classical planning

In the TWEAK formalism and in other classical planners such as STRIPS (Fikes & Nilsson 1971) and SNLP (McAllester & Rosenblitt 1991), a state is represented syntactically as a set of literals. The aim of planning is to find a sequence of actions (operators) that from a given initial state results in a partially specified goal state. Operators are tuples $\alpha = \langle Pre, Post \rangle$ of sets of precondition and postcondition literals. The preconditions specify when the operator is applicable. The postconditions are said to be *asserted* in the state resulting from applying the operator, and their negations are said to be *denied* there. The result of an applicable operator is the input state of the operator plus the asserted literals minus the denied literals. For instance, if an operator $Fire(gun, turkey) = \langle \{loaded(gun)\}, \{\neg alive(turkey), \neg loaded(gun)\} \rangle$ is applied in the state $\{loaded(gun), alive(turkey), hungry(hunter)\}$ then the resulting state would be $\{\neg loaded(gun), \neg alive(turkey), hungry(hunter)\}$. Applying a sequence of operators is defined as functional composition. Classical partial-order planning assumes complete information about actions, and complete information about the initial state is often implicitly assumed. Any completeness results for classical planners rely on complete information concerning the initial state.

*This research was supported in part by the Swedish Research Council for Engineering Sciences (TFR).

Based on the principle of least commitment, a TWEAK plan may be partially instantiated ($Fire(x, turkey)$) and partially ordered (the order between two operators might be unspecified). Further codesignation ($x \approx gun$, $x \not\approx y$) and ordering constraints ($Load(gun) \prec Fire(gun, turkey)$) may be added later during planning. A totally instantiated and totally ordered plan that can be obtained by adding constraints to the plan is called a completion. A partially instantiated and partially ordered plan is interpreted as the set of all its completions.

Efforts towards more expressive planners has been done, for instance the partial-order planner UCPOP (Penberthy & Weld 1992), which can represent actions with quantified preconditions and quantified and context-dependent effects. One of the most interesting issues in present-day planning is planning with incomplete information; the XII planner (Golden, Etzioni, & Weld 1994) is an example of a functioning planner for this case.

A systematic approach

The first motivation of this work is that making formal analyses and descriptions can contribute to a better understanding of how planners work, especially concerning how they represent the world. Logics of action and change can here provide a plan representation with a formal semantics and a solid base for specification and verification of planning systems; this applies both to existing planners and to possible future developments. The basic operations of a planner can be described in logical terms such as inferences. Furthermore, logics of action and change can provide a common basis suitable for comparison and evaluation of different planning systems.

The second motivation is the implicit assumptions about the structure of the world that underlie classical planners. They might eventually turn out to be too restrictive to scale up to the requirements of modeling a complex changing environment. The solution to the frame problem of STRIPS and its descendants, that is syntactically adding and deleting sentences, seems to be hard to extend to handle for instance actions with duration, ramification and concurrency. Incomplete information and knowledge-producing actions (Moore 1985; Golden, Etzioni, & Weld 1994) is another example of a non-trivial extension. Notice that the most serious difficulty is not to find a pragmatic solution to these extensions (although that is far from trivial), but to find a clear semantics.

A third motivation concerns integration. The aim of traditional planning representations have been planning alone. However, planning is but one of the reasoning capabilities required by an agent in a dynamic and complex environment. This puts high demands of expressiveness and flexibility on whatever representation is being used.

It should be emphasized that the objections above do not imply a rejection of existing planning research. On the contrary; the very purpose with this paper is to securely anchor planning with temporal logics in the advances and results that has been achieved in classical planning. On an algorithmic level, classical planners do well, and the general principles and techniques by which they work (establishment, detecting and resolving conflicts etc.) are most likely possible to lift to more advanced ontologies. The representations used (for instance add-delete-lists) are efficient from an implementational point of view. There is no reason to believe for instance that TWEAK would have become a better planner if it had been implemented employing some kind of theorem prover. However, the point is that the classical planning representation have been designed for implementing specific planners. The keywords are "implementing", "specific" for specific planning techniques, ontologies etc, and "planners", that is to say a specific type of reasoning problem. Logics of action and change can here offer a more formally well-defined and general representation. This paper represents a first step in this direction, by reconstructing and analyzing existing planners in the context of a logic of action and change.

This is well in line with Sandewall's systematic approach. One starts with simple, restricted classes of problems, and then incrementally relaxes these restrictions. In the case of planning, this means that one starts with classical planning (as done in this paper), and then relaxes the assumptions inherent in classic planning (complete knowledge, no context-dependency etc.) step by step. In this process, the first step is a prerequisite of the succeeding ones. In another paper (Karlsson 1996), a step towards more advanced planners has been taken: Chapman's truth criterion has been extended to handle worlds where there are actions with context-dependent and non-deterministic effects.

The Fluent Logic for plan representation

The fundamental tools for representing plans and goals in this paper are scenario descriptions. Intuitively, a scenario description expresses the agent's more or less correct and complete information about action laws, action occurrences and observations. It is a structure of formal objects such as sets of logical sentences, each one with a specific functionality. The scenarios described here belong to the $\mathcal{K}$-**IA** class in Sandewall's notation. A $\mathcal{K}$-**IA** scenario description can be written as a tuple $\langle \mathcal{O}, \text{LAW}, \text{SCD}, \text{OBS} \rangle$ where $\mathcal{O}$ is a description of the object domain including unique names axioms; LAW is a set of action laws; SCD is a schedule; and OBS is a set of observations. $\mathcal{K}$-**IA** denotes scenarios with correct and complete information about action laws and action occurrences and correct observations ($\mathcal{K}$), inertia, integer time and actions with duration (**I**), alternative effect of actions (**A**) in terms of context-dependency (different initial states may give different results for an action) and non-determinism (same initial state may give different results) but without concurrency and ramification.[1] Properties and relations in the world that may vary over time are called features, and inertia is the principle that features do not change values unless explicitly affected by an action.

The language used for scenario descriptions in this work is the circumscriptive *fluent logic* (FL) (Doherty & Lukaszewicz 1994; Doherty 1994b). FL is a reified and typed first-order logic using integer time, based on the PMON logic of Sandewall (Sandewall 1994). The most basic building block in FL is $[t]\delta =_{def} Holds(t, \delta)$, stating that δ holds at time t. The well known Yale shooting scenario is represented in FL as follows.

$$\begin{array}{ll} \mathsf{law1} & [s,t]\,Load(x) \rightsquigarrow [s,t]\,loaded(x){:=}T \\ \mathsf{law2} & [s,t]\,Fire(x,y) \rightsquigarrow [s]\,loaded(x) \Rightarrow \\ & \quad ([s,t]\,loaded(x){:=}F \wedge [s,t]\,alive(y){:=}F) \\ \mathsf{obs1} & [0]\,alive(turkey) \wedge \neg loaded(gun) \\ \mathsf{scd1} & [2,4]\,Load(gun) \\ \mathsf{scd2} & [5,6]\,Fire(gun, turkey) \end{array} \qquad (1)$$

Lines labeled law belong to the set LAW, and so on. Applying the action laws, which are syntactic expansion rules, to the schedule yields the following result.

$$\begin{array}{ll} \mathsf{obs1} & [0]\,alive(turkey) \wedge \neg loaded(gun) \\ \mathsf{scd1} & [2,4]\,loaded(gun){:=}T \\ \mathsf{scd2} & [5]\,loaded(gun) \Rightarrow ([5,6]\,loaded(gun){:=}F \\ & \quad \wedge [5,6]\,alive(turkey){:=}F) \end{array} \qquad (2)$$

Reassignment, $[s,t]\delta{:=}\mathcal{B}$, plays a key role in the logic. It denotes that somewhere in the interval $(s,t]$ the feature δ will be assigned the value $\mathcal{B}$. The definition is as follows:

$$\begin{array}{l} [s,t]\,\delta{:=}T =_{def} \\ \exists t.(s \leq t < t \wedge \forall t'.(t < t' \leq t \Rightarrow Holds(t', \delta))) \\ \wedge \forall t''.(s < t'' \leq t \Rightarrow Occlude(t'', \delta)) \end{array} \qquad (3)$$

and similarly for $[s,t]\delta{:=}F$. The first part describes the result of the reassignment. The $Occlude(t'', \delta)$ expression of the second part denotes that the truth value of feature δ may change at time t''. PMON (Sandewall 1994) minimizes *Occlude* for each timepoint-feature pair in parallel. In FL, the minimization is done using a second-order cirumscription (Lifschitz 1985) schema. For YSS, occlusion after PMON-minimization will be as follows:

$$\begin{array}{l} \forall t, f. Occlude(t, f) \equiv \\ \quad (\langle t, f\rangle = \langle 3, loaded(gun)\rangle \\ \quad \vee \langle t, f\rangle = \langle 4, loaded(gun)\rangle \\ \quad \vee (\ Holds(5, loaded(gun)) \wedge \\ \quad\quad (\langle t, f\rangle = \langle 6, loaded(gun)\rangle \vee \\ \quad\quad \langle t, f\rangle = \langle 6, alive(turkey)\rangle))) \end{array} \qquad (4)$$

[1] Note that although these are quite strong restrictions from the perspective of real-world applications, they are still far less restrictive than the assumptions made in classical planning.

Then a nochange axiom is applied, stating that features can change from the previous time point only when occluded ($\oplus$ denotes 'exclusive or'):

$$\begin{array}{l} \forall f, t.\ Holds(t, f) \oplus Holds(t+1, f) \Rightarrow \\ \quad Occlude(t+1, f). \end{array} \qquad (5)$$

Deduction under PMON will be written $\mid\!\sim$. The conclusions concerning *Holds* in YSS are as follows.

$$\begin{array}{l} [0,2]\,alive(turkey) \wedge \neg loaded(gun) \\ [3]\,alive(turkey) \\ [4,5]\,alive(turkey) \wedge loaded(gun) \\ [6,\infty)\neg alive(turkey) \wedge \neg loaded(gun) \end{array} \qquad (6)$$

The object domain may be finite or infinite. There are no object domain functions. PMON has been proved to be correctly applicable for the $\mathcal{K}$-**IA** class (Sandewall 1994).

Scenarios corresponding to classical monotonic theories (denoted $\mathcal{N}$) can be written as a two-tuple, for instance $\langle \mathcal{O}, \text{GOAL}\rangle$ for a set of goals. Finally, the expansion of the schedule according to the action laws is denoted LAW(SCD).

As shown by Doherty (Doherty 1994a), FL scenario descriptions have the *restricted monotonicity property* (Lifschitz 1993). Some classes of statements of a non-monotonic formalism monotonically increases the set of valid conclusions when added to a set of premises. Such classes can be identified for FL scenario descriptions, namely observations and temporal ($t_1 = t_2$, $t_1 \leq t_2$, and $t_1 < t_2$) and atemporal ($e_1 = e_2$, $e_1 \neq e_2$) constraints. Adding statements of these classes to a scenario description will not invalidate any conclusions of the form $\langle \mathcal{O}, \text{LAW}, \text{SCD}, \text{OBS}\rangle \mid\!\sim \alpha$. Scenarios extended with these classes of sentences are called monotonic extensions.

Planning with FL

Plan synthesis can be described as the reasoning problem that given the initial state of the world

$$\langle \mathcal{O}, \text{LAW}, \emptyset, \text{OBS}\rangle \qquad (7)$$

and a set of goals

$$\langle \mathcal{O}, \text{GOAL}\rangle \qquad (8)$$

one is to find a consistent plan

$$\langle \mathcal{O}, \emptyset, \text{SCD}, \emptyset\rangle \qquad (9)$$

such that

$$\langle \mathcal{O}, \text{LAW}, \text{SCD}, \text{OBS}\rangle \mid\!\sim \langle \mathcal{O}, \text{GOAL}\rangle \qquad (10)$$

where SCD contains only actions that the agent is allowed to perform. Thus, planning is an abductive problem.

A problem which has to be taken into consideration is the case where the agent is to reason about partial plans where actions are still missing. For instance, assume that an agent has just planned to shoot the turkey but not yet planned to load the gun. Using the YSS (1) with scd2 but without scd1 the agent would not be able to derive anything about the effects and conditions of the shooting action. As $[5]\,loaded(gun)$ does not hold in the scenario, $[5,6]\,alive(turkey){:=}F$ is not a consequence. A plan where some preconditions do not hold is called invalid.

A solution is to split the action laws into two parts. The action qualification[2] laws state the preconditions of an action. The action effect laws define the result of the action (Karlsson 1995). The separated version of the YSS with a goal added is as follows.

$$\begin{array}{ll}
\text{qlaw1} & [s,t]\,Load(x) \leadsto T \\
\text{elaw1} & [s,t]\,Load(x) \leadsto [s,t]\,loaded(x){:=}T \\
\text{qlaw2} & [s,t]\,Fire(x,y) \leadsto [s]\,loaded(x) \\
\text{elaw2} & [s,t]\,Fire(x,y) \leadsto \\
 & \quad [s,t]\,loaded(x){:=}F \wedge [s,t]\,alive(y){:=}F \\
\text{obs1} & [0]\,alive(turkey) \wedge \neg loaded(gun) \\
\text{scd1} & [2,4]\,Load(gun) \\
\text{scd2} & [5,6]\,Fire(gun, turkey) \\
\text{goal1} & [8]\neg alive(turkey)
\end{array} \tag{11}$$

In the scenario description $\langle \mathcal{O}, \text{ELAW}, \text{SCD}, \text{OBS}\rangle$ above, all actions always have effects as if their preconditions were true. The YSS after expansion is as follows:

$$\begin{array}{ll}
\text{obs1} & [0]\,alive(turkey) \wedge \neg loaded(gun) \\
\text{scd1} & [2,4]\,loaded(gun){:=}T \\
\text{scd2} & [5,6]\,loaded(gun){:=}F \wedge \\
 & \quad [5,6]\,alive(turkey){:=}F
\end{array} \tag{12}$$

and the goal and preconditions $\langle \mathcal{O}, \text{QLAW}(\text{SCD}) \cup \text{GOAL}\rangle$:

$$\begin{array}{ll}
\text{scd1} & T \\
\text{scd2} & [5]\,loaded(gun) \\
\text{goal1} & [8]\neg alive(turkey)
\end{array} \tag{13}$$

From $\langle \mathcal{O}, \text{ELAW}, \text{SCD}, \text{OBS}\rangle$ the agent can decide that the turkey will die, and from $\langle \mathcal{O}, \text{QLAW}(\text{SCD}) \cup \text{GOAL}\rangle$ he can tell that the gun should be loaded first. Observe that ELAW does not require $[5]\,loaded(gun)$ to hold. ELAW always produces the intended effects of an action, even if the preconditions do not hold.

A solution obtained by reasoning with ELAW and QLAW is also a solution for LAW.

Theorem 1 *(Karlsson 1995) Let* LAW *be a set of action laws, each one with only one implication from preconditions to effects, and let* QLAW *and* ELAW *be the corresponding action qualification and effect laws. If*

$$\langle \mathcal{O}, \text{ELAW}, \text{SCD}, \text{OBS}\rangle \mathrel{|\!\!\sim} \langle \mathcal{O}, \text{QLAW}(\text{SCD}) \cup \text{GOAL}\rangle$$

then

$$\langle \mathcal{O}, \text{LAW}, \text{SCD}, \text{OBS}\rangle \mathrel{|\!\!\sim} \langle \mathcal{O}, \text{GOAL}\rangle.$$

Partial-order plans and TWEAK

It is now possible to define an FL version of TWEAK. The FL version of a TWEAK plan consists of action effect laws, action occurrences, ordering constraints and codesignation constrains and observations relating to the first time point, following the schemas below.

$$\begin{array}{ll}
\text{elaw} & [t,t']\,A_j(\overline{x}) \leadsto \\
 & \quad [t,t']\,p_1(\overline{x}_1){:=}\mathcal{B}_1 \wedge \ldots [t,t']\,p_n(\overline{x}_n){:=}\mathcal{B}_n \\
\text{scd} & [S_i, S_i+1]\,A_j(\overline{e}_i) \\
\text{scd} & S_i < S_j \\
\text{scd} & e_i = e_j \\
\text{scd} & e_i \neq e_j \\
\text{obs} & [0][\neg]p(\overline{e})
\end{array} \tag{14}$$

All e_i, e_j denote atemporal constants, $\overline{e}_i$ is zero or more constants, and $\mathcal{B}_i \in \{T,\ F\}$. A set of temporal constants $\{S_i\}_i$ represents the time-points of partially ordered actions. Each action occurrence is associated with a unique S_i. As actions are sequential, all these have to be disjoint:

$$\text{scd} \quad S_i \neq S_j \text{ for each } i,\, j \text{ such that } i \neq j \tag{15}$$

A special set of atemporal constants $\{V_i\}_i$ is introduced to denote arbitrary objects in the domain and are used to represent partial instantiation. The S_i and V_i can be seen as global variables.

The goal and preconditions constitute a scenario description $\langle \mathcal{O}, \text{GOAL} \cup \text{QLAW}(\text{SCD})\rangle$, where GOAL is the goal description, a conjunction of the form:

$$\text{goal} \quad [S_0][\neg]q_1(\overline{e}_1) \wedge \ldots \wedge [S_0][\neg]q_n(\overline{e}_n) \tag{16}$$

where S_0 is a constant representing the unspecified time-point of the goal. QLAW represents the action qualification laws of the form:

$$\begin{array}{ll}
\text{qlaw} & [t,t']\,A_i(\overline{x}) \leadsto \\
 & \quad [t][\neg]q_1(\overline{x}_1) \wedge \ldots \wedge [t][\neg]q_n(\overline{x}_n).
\end{array} \tag{17}$$

The next issue is how to synthesize the plan. The central part of TWEAK is the modal truth criterion (MTC), stating necessary and sufficient conditions for necessary and (erroneously, see (Karlsson 1995)) possible truth. The former is defined as truth in all completions, the latter in some completion. The MTC can be used to decide whether a literal (Chapman uses the word "proposition") holds in a specific situation in the plan. It can also be used to decide what should be added to the plan in order to make a literal hold in a situation. This is the idea behind Chapman's nondeterministic goal achievement procedure.

Theorem 2 (Modal Truth Criterion) *(Chapman 1987) A proposition p is necessarily true in a situation s iff two conditions hold: there is a situation t equal or necessarily previous to s in which p is necessarily asserted; and for every step C possibly before s and every proposition q possibly codesignating with p which C denies, there is a step W necessarily between C and s which asserts r, a proposition such that r and p codesignate whenever p and q codesignate. The criterion for possible truth is exactly analogous, with all the modalities switched (read "necessary" for "possible" and vice versa).*

The symbols p, q and r denote literals, s and t denote situations and C and W denote steps. The s situation is called an establisher of the proposition. This

[2]The term "qualification" as used here should not be confused with the *qualification problem* (McCarthy 1977).

is the situation, either the initial situation or the output situation of some step, that make the proposition p become true. When planning, each goal and precondition should be given an establisher, either by relating to the initial situation or an existing step, or by adding a new step. Besides establishing goals, a planner also has to resolve conflicts. The C steps are called clobberers, and the W steps are called white knights. A clobberer is a step that might make a precondition or goal become false, and the white knight is used to repair the damage of the clobberer. A second alternative is to add codesignation constraints to separate the clobberer from the goal. For instance, if the goal is $p(x)$ and the clobberer denies $p(y)$ then the two can be separated by constraining $x \not\approx y$. A third alternative is to move the clobberer out of the way. The goal achievement procedure operates by giving a proposition an establisher and resolving conflicts using white knights, separation, and ordering constraints.

Chapman's modal truth criterion can be reconstructed in a fairly straight-forward manner. Necessary truth is expressed as logical consequence under PMON circumscription $\mid\!\sim$. The indexes of the steps are chosen according to the letters in the modal truth criterion above.

Theorem 3 (Truth criterion for partial-order plans (TWEAK))
Let $\Gamma = \langle \mathcal{O}, \text{ELAW}, \text{SCD}, \text{OBS}\rangle$ be a TWEAK plan.

$$\Gamma \mid\!\sim \ulcorner [S_s] p(\overline{e}_s) \urcorner \text{ iff}$$

$$\begin{aligned}
&\exists S_e \; [\; (\Gamma \mid\!\sim \ulcorner S_e < S_s \urcorner \wedge \\
&\qquad \Gamma \mid\!\sim \ulcorner [S_e, S_e + 1] p(\overline{e}_s) := T \urcorner) \vee \\
&\qquad \Gamma \mid\!\sim \ulcorner [0] p(\overline{e}_s) \urcorner \;] \wedge \\
&\forall S_c \; [\; \Gamma \mid\!\sim \ulcorner S_s \leq S_c \urcorner \vee \\
&\qquad \forall \overline{e}_c \; [\; \Gamma \mid\!\sim \ulcorner [S_c, S_c + 1] p(\overline{e}_c) := F \urcorner \Rightarrow \\
&\qquad\qquad \Gamma \mid\!\sim \ulcorner \vee_i (e_{si} \neq e_{ci}) \urcorner \vee \\
&\qquad\qquad \exists S_w, \; \overline{e}_w \; [\Gamma \mid\!\sim \ulcorner S_c < S_w \urcorner \wedge \\
&\qquad\qquad\qquad \Gamma \mid\!\sim \ulcorner S_w < S_s \urcorner \wedge \\
&\qquad\qquad\qquad \Gamma \mid\!\sim \ulcorner [S_w, S_w + 1] p(\overline{e}_w) := T \urcorner \wedge \\
&\qquad\qquad\qquad \Gamma \mid\!\sim \ulcorner \; \wedge_i (e_{ci} = e_{si}) \Rightarrow \\
&\qquad\qquad\qquad\qquad \wedge_i (e_{wi} = e_{si}) \urcorner \;]\;]\;]
\end{aligned}$$

For negated feature statements, substitute $\ulcorner[..]\neg p(..)\urcorner$ for $\ulcorner[..]p(..)\urcorner$ and F for T (and vice versa) above.

The truth criterion depends on an infinite domain; thus a domain closure assumption is not possible. Observe that the criterion is about plans with ELAW. However, it applies also to complete plans with LAW. A full proof appears in (Karlsson 1995). A few points should be made here. First, the truth criterion is independent of the application, in this case planning. It holds for any scenario that satisfies the restrictions in the previous section. Second, the criterion is applicable to both valid and invalid plans. This is obtained by using ELAW to represent the effects. Third, reassignment plays a key role in the criterion. The nature of the reassignment statement makes it possible to easily identify the points of assertion. As ELAW does not include any conditionals, a sentence such as $\ulcorner [S_e, S_e + 1] p(\overline{e}_s) := T \urcorner$ is either true or false in all models.

The goal achievement procedure (Chapman 1987) retains its structure; only the basic operations need be altered to correspond to the FL representation. These operations consists of adding sentences to the schedule and making derivations.

procedure **goal-achievement**($[S_s] p(\overline{e}_s)$)

1. Establishment — choose:

(a) Infer that $\Gamma \mid\!\sim \ulcorner [0] p(\overline{e}_e) \urcorner$, and infer or add:

$$\textsf{scd} \quad e_{si} = e_{ei} \text{ for each } i \tag{18}$$

(b) Select an existing step S_e such that $\Gamma \mid\!\sim \ulcorner [S_e, S_e + 1] p(\overline{e}_e) := T \urcorner$, and infer or add:

$$\begin{aligned} &\textsf{scd} \quad e_{si} = e_{ei} \text{ for each } i \\ &\textsf{scd} \quad S_e < S_s \end{aligned} \tag{19}$$

(c) Add a new step to Γ:

$$\begin{aligned} &\textsf{scd} \quad [S_e, S_e + 1] A(\overline{e}') \\ &\textsf{scd} \quad S_e < S_s \end{aligned} \tag{20}$$

such that $\Gamma \mid\!\sim \ulcorner [S_e, S_e + 1] p(\overline{e}_s) := T \urcorner$.

2. Declobbering — for each existing S_c such that $\Gamma \not\mid\!\sim \ulcorner S_s \leq S_c \urcorner$ and for each $\overline{e}_c$ such that $\Gamma \mid\!\sim \ulcorner [S_c, S_c + 1] p(\overline{e}_c) := F \urcorner$ choose:

(a) Promote the clobberer: $\textsf{scd}$ $S_s < S_c$

(b) Separate the steps: $\textsf{scd}$ $e_{si} \neq e_{ci}$ for some i.

(c) White Knight — find an old step or add a new step $[S_w, S_w + 1] A(\overline{e}')$ to Γ such that $\Gamma \mid\!\sim \ulcorner [S_w, S_w + 1] p(\overline{e}_w) := T \urcorner$ and add or infer:

$$\begin{aligned} &\textsf{scd} \quad S_c < S_w \\ &\textsf{scd} \quad S_w < S_s \end{aligned} \tag{21}$$

plus add atemporal constraints such that: $\Gamma \mid\!\sim \ulcorner \overline{e}_c = \overline{e}_s \Rightarrow \overline{e}_w = \overline{e}_s \urcorner$

Assume an FL version of TWEAK given the initial state $[0] alive(turkey) \wedge [0] \neg loaded(gun)$ is to achieve the goal $[S_0] \neg alive(turkey)$. The actions from (11) are available. The schedule, representing the actual plan, is initially empty. The planner would first add $[S_1, S_1 + 1] Fire(V_1, turkey)$ and $S_1 < S_0$, as this according to ELAW will yield $[S_1, S_1 + 1] alive(turkey) := F$ (establishment). The $Fire$ action has a precondition $[S_1] loaded(V_1)$. That can in turn be achieved by adding the action $[S_2, S_2 + 1] Load(V_1)$ and $S_2 < S_1$, as this according to ELAW will yield $[S_2, S_2 + 1] loaded(V_1) := T$. Now no subgoals remains, and the plan can be completed with $V_1 = gun$.

Causal links and SNLP

McAllester's and Rosenblitt's systematic nonlinear planner (SNLP) (McAllester & Rosenblitt 1991) basically has the same plan representation as TWEAK, with one addition: causal links. A causal link is a triple $\langle s, P, w\rangle$ where P is a literal, w is a step name that has P as a precondition, and s is a step name that has P in its add list. A causal link states an ordering between two steps; the link above constrains s to precede w. It also indicates a dependency between the two steps; s supplies P for w. Thus, causal links encode teleological information. A causal link on condition p between step s and step w states that one of the purposes of s is to supply p for w. A step name v is called a threat to a causal link $\langle s, P, w\rangle$ if v is not identical to s or w and v either adds or deletes P. For some other planners (Tate 1977), this condition is weakened to only cover steps that delete P. The stronger version is used in SNLP in order to make the search of the plan space systematic by giving each goal or subgoal a unique supplier.

The aim of the planning process is to connect to each goal[3] and precondition P of a step w in the plan a causal link $\langle s, P, w\rangle$, and that no causal link is threatened by some step. A plan that satisfies these two conditions is called a complete plan, and any completion of a complete plan is a solution. Completion here means the same thing as for TWEAK.

There is a strong connection between the *Occlude* predicate and the concept of a causal link. The former explicitly represents that a feature is influenced by some action, whereas the latter forbids any influence within a specific interval. Thus, causal links can be represented in FL as follows. First, an ordering constraint appearing in the plan:

$$\mathsf{scd} \quad S_s < S_w \tag{22}$$

Second, an unoccluded interval which prevents the feature to be reassigned:

$$\mathsf{csl} \quad (S_s + 1, S_w]\neg p^*(\overline{e}) \tag{23}$$

where $(\boldsymbol{s}, \boldsymbol{t}]\neg\delta^* =_{def} \forall t.(\boldsymbol{s} < t \leq \boldsymbol{t} \Rightarrow \neg Occlude(t, \delta))$. All the csl components constitute a scenario description:

$$\langle \mathcal{O}, \textsc{Csl}\rangle \tag{24}$$

The concept of a complete SNLP plan being a solution is straight-forwardly transferred to the FL-SNLP representation.

Theorem 4 (SNLP) *(Karlsson 1995)*
Given a plan $\langle \mathcal{O}, \textsc{Elaw}, \textsc{Scd}, \textsc{Obs}\rangle$, goals and preconditions $\langle \mathcal{O}, \textsc{Goal} \cup \textsc{Qlaw}(\textsc{Scd})\rangle$, and a causal link description $\langle \mathcal{O}, \textsc{Csl}\rangle$, if the following conditions hold:

1. *For each goal and precondition $[S_w]p(\overline{e})$ in $\langle \mathcal{O}, \textsc{Goal} \cup \textsc{Qlaw}(\textsc{Scd})\rangle$, there is a causal link*

$$\begin{array}{ll}\mathsf{scd} & S_s < S_w \\ \mathsf{csl} & (S_s + 1, S_w]\neg p^*(\overline{e})\end{array} \tag{25}$$

such that

$$\langle \mathcal{O}, \textsc{Elaw}, \textsc{Scd}, \textsc{Obs}\rangle \mathrel{|\!\sim} \ulcorner[S_s, S_s + 1]p(\overline{e}) := T\urcorner,$$

or a causal link

$$\mathsf{csl} \quad (0, S_w]\neg p^*(\overline{e}) \tag{26}$$

such that

$$\langle \mathcal{O}, \textsc{Elaw}, \textsc{Scd}, \textsc{Obs}\rangle \mathrel{|\!\sim} \ulcorner[0]p(\overline{e})\urcorner;$$

and the corresponding condition holds for each goal and precondition $[S_w]\neg p(\overline{e})$ (substitute F for T and $[0]\neg p(\overline{e})$ for $[0]p(\overline{e})$ above).

2. *For each causal link $(S_s + 1, S_w]\neg p^*(\overline{e})$ in $\langle \mathcal{O}, \textsc{Csl}\rangle$ and each effect*

$$\langle \mathcal{O}, \textsc{Elaw}, \textsc{Scd}, \textsc{Obs}\rangle \mathrel{|\!\sim} \ulcorner[S_v, S_v + 1]p(\overline{e}) := \mathcal{B}\urcorner$$

where $\mathcal{B} \in \{T, F\}$, one of the following conditions holds:

   (a) $\langle \mathcal{O}, \textsc{Elaw}, \textsc{Scd}, \textsc{Obs}\rangle \mathrel{|\!\sim} \ulcorner S_v \leq S_s\urcorner$

   (b) $\langle \mathcal{O}, \textsc{Elaw}, \textsc{Scd}, \textsc{Obs}\rangle \mathrel{|\!\sim} \ulcorner S_w \leq S_v\urcorner$,

then:

$$\langle \mathcal{O}, \textsc{Elaw}, \textsc{Scd}, \textsc{Obs}\rangle \mathrel{|\!\sim} \langle \mathcal{O}, \textsc{Goal} \cup \textsc{Qlaw}(\textsc{Scd})\rangle;$$

that is to say, the plan is a solution.

Observe that for each complete plan, the truth criterion for partial-order plans (theorem 3) holds for all members of $\textsc{Goal} \cup \textsc{Qlaw}(\textsc{Scd})$. As each member has a causal link that is not threatened, the following conditions hold:

1. There is an establisher S_s for each goal or precondition $[S_w]p(\overline{e})$.
2. For any clobberer S_v, either

$$\langle \mathcal{O}, \textsc{Elaw}, \textsc{Scd}, \textsc{Obs}\rangle \mathrel{|\!\sim} \ulcorner S_w \leq S_v\urcorner$$

or

$$\langle \mathcal{O}, \textsc{Elaw}, \textsc{Scd}, \textsc{Obs}\rangle \mathrel{|\!\sim} \ulcorner S_v \leq S_s\urcorner;$$

in the latter case, the establisher serves as a white knight.

Furthermore, condition 2 of theorem 4 assures that no unoccluded interval in Csl is violated: occlusion only occurs as a component of reassignment (:=), and no reassignment is allowed within the interval. Thus

$$\langle \mathcal{O}, \textsc{Elaw}, \textsc{Scd}, \textsc{Obs}\rangle \mathrel{|\!\sim} \langle \mathcal{O}, \textsc{Csl}\rangle.$$

It is also possible to define the weaker kind of causal links in FL. If a reassignment $\ulcorner[S_v, S_v + 1]p(\overline{e}) := \mathcal{B}\urcorner$ is not to be considered a threat when $\mathcal{B} = T$, only when $\mathcal{B} = F$, a causal link can be specified as

$$\begin{array}{ll}\mathsf{scd} & S_s < S_w \\ \mathsf{csl} & (S_s + 1, S_w](\neg p^*(\overline{e}) \vee p(\overline{e}))\end{array} \tag{27}$$

where

$$(\boldsymbol{s}, \boldsymbol{t}](\neg\delta^* \vee \delta) =_{def} \tag{28}$$
$$\forall t.(\boldsymbol{s} < t \leq \boldsymbol{t} \Rightarrow (\neg Occlude(t, \delta) \vee Holds(t, \delta))).$$

For illustrating the operations of FL-SNLP, a slightly altered version of the YSS is used. The initial state is

$$\mathsf{obs} \quad [0]alive(turkey) \wedge [0]\neg loaded(gun) \tag{29}$$

and the (extended) goal is

$$\mathsf{goal} \quad [S_0]loaded(gun) \wedge [S_0]\neg alive(turkey). \tag{30}$$

The planner would first attempt to achieve the goal $[S_0]loaded(gun)$. This can be done by adding the action $[S_1, S_1 + 1]Load(gun)$, $S_1 < S_0$ and a causal link

[3] The goal state is encoded as a step appearing last in the plan.

$(S_1 + 1, S_0]\neg loaded^*(gun)$. Next, the planner would add the action and link $[S_2, S_2 + 1]\, Fire(V_1, turkey)$, $S_2 < S_0$ and $(S_2 + 1, S_0]\neg alive^*(turkey)$. At this point, there is a threat to the causal link $(S_1, S_0]\neg loaded^*(gun)$. By the fire action, $[S_2, S_2 + 1]\, loaded(V_1){:=}F$. This threat is resolved by constraining $S_2 < S_1$. The *Fire* action has a precondition $[S_2]\, loaded(V_1)$. That can in turn be achieved by adding the action $[S_3, S_3 + 1]\, Load(V_1)$ and $S_3 < S_2$, as this according to ELAW will yield $[S_3, S_3 + 1]\, loaded(V_1){:=}T$. Now no subgoals remains, and the plan can be completed with $V_1 = gun$.

Operations of the FL planners

The FL versions of the planners perform two kinds of operations. The first one is to add new statements to the plan. The types of statements added are (a) new actions, (b) temporal and atemporal constraints, and for SNLP (c) causal links. For (a) the circumscription of the plan has to be recomputed. However, (b) results in monotonic extensions as described at p. 3 and thus does not require any recomputation. Finally, (c) is not added to $\langle \mathcal{O}, \text{ELAW}, \text{SCD}, \text{OBS} \rangle$ and thus does not require any recomputation. The second type of operation is to decide if something is a consequence. The different cases are (a) $\Gamma \mathrel{|\!\sim} \ulcorner [S_i, S_i + 1]\, f(\overline{e}){:=}\mathcal{B}\urcorner$, which, as ELAW is used, collapses to check whether there is any explicit $\ulcorner [S_i, S_i + 1]\, f(\overline{e}'){:=}\mathcal{B}\urcorner$ in Γ and then prove $\ulcorner \bigwedge_i (e_k = e'_k)\urcorner$; (b) $\Gamma \mathrel{|\!\sim} \ulcorner S_i < S_j \urcorner$, and (c) $\Gamma \mathrel{|\!\sim} \ulcorner e_{ik} = e_{jk}\urcorner$.

Related work

Using logics of action and change for planning is in itself not a new idea (Green 1969; Allen *et al.* 1991). CHICA (Missiaen, Bruynooghe, & Denecker 1995) is an interesting planner, which uses event calculus (Kowalski & Sergot 1986) in Horn-clause form as a representation language. The implementation is based on a theorem prover using SLDNFA resolution, which is an abductive extension of SLDNF resolution. A central part is a *domain-independent theory* in Horn-clause form that corresponds to TWEAK's modal truth criterion. This theory states that a property P holds at a point T in time if P is initiated by an event E preceding T and P is maintained between E and T. Maintenance is in turn defined as that there has to be a white knight each time there is an event that terminates P (that is a clobberer that makes P false). However, although the similarities are strong, CHICA is not a logic programming version of TWEAK. It is a distinct planning system with an expressiveness exceeding TWEAK (it can represent context-dependent effects of actions and a limited form of ramification). Furthermore, CHICA is not complete and sometimes it also returns incorrect plans.

In the area of planning with incomplete information, there has been a number of formal studies based on logics of action and change (Moore 1985; Davis 1994; Levesque 1996).

Conclusions

This paper argues for the use of temporal logics for plan representation in order to provide a robust formal foundation for the analysis and development of advanced planners. A first step in this direction has been taken with a study in partial-order planning. The partial-order planners TWEAK and SNLP, the latter using causal links, have been the subjects of an analysis and reconstruction. The tool has been the fluent logic, a logic rooted in Sandewall's systematic approach to action and change (Sandewall 1994) that can represent partially ordered actions and subsumes the classical plan representations by its ability to express actions with duration and context-dependent and nondeterministic effects. An FL representation for TWEAK plans was formulated, and the modal truth criterion was converted to this representation. Then the causal links used in SNLP was reformulated in the fluent logic, and the strong connection to the *Occlude* predicate that represents influence of an action to a feature was pointed out. Reassignment, containing *Occlude*, is important also for positive identification of the points of influence. Finally, FL is a nonmonotonic logic but several of the most common operations result in monotonic extensions.

References

Allen; Kautz; Pelavin; and Tenenberg., eds. 1991. *Reasoning About Plans.* Morgan Kaufmann.

Allen, J.; Hendler, J.; and Tate, A., eds. 1990. *Readings in Planning.* Morgan Kaufmann.

Chapman, D. 1987. Planning for conjunctive goals. *Artificial Intelligence* 32:333–377. Reprinted in (Allen, Hendler, & Tate 1990).

Davis, E. 1994. Knowledge preconditions for plans. *Journal of Logic and Computation* 4(5):721–766.

Doherty, P., and Lukaszewicz, W. 1994. Circumscribing features and fluents. In *Proceedings of the 1st International Conference on Temporal Reasoning*, 82–100. Springer.

Doherty, P. 1994a. Notes on PMON circumscription. Technical Report 94-43, Department of Computer and Information Science, Linköping University. Available on WWW: `http://www.ida.liu.se/labs/rkllab/people/patdo/`.

Doherty, P. 1994b. Reasoning about action and change using occlusion. In *Proceedings of the Eleventh European Conference on Artificial Intelligence.* John Wiley & Sons.

Fikes, R. E., and Nilsson, N. J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208. Reprinted in (Allen, Hendler, & Tate 1990).

Golden, K.; Etzioni, O.; and Weld, D. 1994. Omnipotence without omniscience: Efficient sensor management for planning. In *Proceedings of the Twelveth National Conference on Artificial Intelligence.* AAAI Press, Menlo Park, California.

Green, C. 1969. Applications of theorem proving to problem solving. In *Proceedings of the First International Joint Conference on Artificial Intelligence.* Morgan Kaufmann. Reprinted in (Allen, Hendler, & Tate 1990).

IJCAI 1977. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann.

Karlsson, L. 1995. Specification and synthesis of plans using the features and fluents framework. Licentiate thesis, Department of Computer and Information Science, Linköping University. Available on WWW: `http://www.ida.liu.se/labs/rkllab/people/larka/`.

Karlsson, L. 1996. Planning, truth criteria and the systematic approach to action and change. In *Proceedings of the 9th International Symposium on Methodologies for Intelligent Systems*, Lecture Notes for Artificial Intelligence. Springer Verlag.

Kowalski, R. A., and Sergot, M. 1986. A logic-based calculus of events. *New Generation Computing* 4:67–95.

Levesque, H. J. 1996. What is planning in the presence of sensing? In *Proceedings of the Thirteenth National Conference on Artificial Intelligence.* AAAI Press, Menlo Park, California.

Lifschitz, V. 1985. Computing circumscription. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 121–127. Morgan Kaufmann.

Lifschitz, V. 1993. Restricted monotonicity. In *Proceedings of the Eleventh National Conference on Artificial Intelligence.* AAAI Press, Menlo Park, California.

McAllester, D., and Rosenblitt, D. 1991. Systematic nonlinear planning. In *Proceedings of the Ninth National Conference on Artificial Intelligence.* AAAI Press, Menlo Park, California.

McCarthy, J. 1977. Epistemological problems of artificial intelligence. In IJCAI77 (1977).

Missiaen, L.; Bruynooghe, M.; and Denecker, M. 1995. Chica, an abductive planning system based on event calculus. *Journal of Logic and Computation* 5(5):579–602.

Moore, R. C. 1985. A formal theory of knowledge and action. In Hobbs, J., and Moore, R., eds., *Formal Theories of the Commonsense World.* Ablex, Norwood, New Jersey. chapter 9.

Penberthy, J. S., and Weld, D. S. 1992. UCPOP: A sound, complete, partial order planner for ADL. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference.* Morgan Kaufmann.

Sandewall, E. 1994. *Features and Fluents.* Oxford Press.

Tate, A. 1977. Generating project networks. In IJCAI77 (1977), 888–93. Reprinted in (Allen, Hendler, & Tate 1990).