Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Research
Report
RR-91-34

On the Computational Complexity of Temporal Projection and some Related Problems

Bernhard Nebel, Christer Bäckström

October 1991

Deutsches Forschungszentrum für Künstliche Intelligenz
GmbH

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern und Saarbrücken is a non-profit organization which was founded in 1988 by the shareholder companies ADV/Orga, AEG, IBM, Insiders, Fraunhofer Gesellschaft, GMD, Krupp-Atlas, Mannesmann-Kienzle, Philips, Siemens and Siemens-Nixdorf. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- ❑ Intelligent Engineering Systems
- ❑ Intelligent User Interfaces
- ❑ Intelligent Communication Networks
- ❑ Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

On the Computational Complexity of Temporal Projection and some Related Problems

Bernhard Nebel and Christer Bäckström

DFKI-RR-91-34

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-8901 8).

On the Computational Complexity of Temporal Projection and some Related Problems*†

Bernhard Nebel
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3, D-6600 Saarbrücken 11, Germany
e-mail: nebel@dfki.uni-sb.de

Christer Bäckström
Department of Computer and Information Science,
Linköping University, S-581 83 Linköping, Sweden
e-mail: cba@ida.liu.se

Abstract

One kind of temporal reasoning is *temporal projection*—the computation of the consequences for a set of events. This problem is related to a number of other temporal reasoning tasks such as *story understanding*, *plan validation*, and *planning*. We show that one particular simple case of temporal projection on partially ordered events turns out to be harder than previously conjectured. However, given the restrictions of this problem, planning and story understanding are easy. Additionally, we show that plan validation, one of the intended applications of temporal projection, is tractable for an even larger class of plans. The incomplete decision procedure for the temporal projection problem that has been proposed by other authors, however, fails to be complete in the case where we have shown plan validation to be tractable.

---

*This work was supported by the German Ministry for Research and Technology (BMFT) under contract ITW 8901 8 and the Swedish National Board for Technology Development (STU) under grant STU 90-1675.

†This paper is also published as Research Report LiTH-IDA-R-91-34, Department of Computer and Information Science, Linköping University, Linköping, Sweden.

Contents

1 Introduction

The problem of *temporal projection* is to compute the consequences of a set of events. Dean and Boddy [1988] analyze this problem for sets of partially ordered events assuming a propositional STRIPS-like [Fikes and Nilsson, 1971] representation of events. They investigate the computational complexity of a number of restricted problems and conclude that even for severely restricted cases the problem is NP-hard, which motivate them to develop a tractable and sound but incomplete decision procedure for the temporal projection problem.

Among the restricted problems they analyze, there is one they conjecture to be solvable in polynomial time. As it turns out, however, even in this case temporal projection is NP-hard, as is shown in Section 3. This result does not undermine the arguments of Dean and Boddy [1988], but rather confirms that the problem of temporal projection as they define it is very difficult—even in its simplest form. The result is somewhat surprising, however, because *planning*, *plan validation*, and *story understanding* seem to be easily solvable given the restriction of this temporal projection problem.

The problem of *planning* is, given a current world state, a desired world state, and a set of possible actions that can be executed, find a sequence of actions that, if executed in the current world state, will bring about the desired world state. Planning is a very difficult problem [Chapman, 1987; Bylander, 1991; Chenoweth, 1991; Gupta and Nau, 1991]. However, if we apply the restrictions of the simple temporal projection problem to the formulation of the planning problem, planning turns out to be trivial. Plans of minimal length are derivable in polynomial time. Further, this result can be strengthened to a less restricted problem—the SAS-PUS planning problem [Bäckström and Klein, 1991a].

This observation casts some doubts on whether temporal projection is indeed the problem underlying planning, plan validation, and story understanding, as suggested by Dean and Boddy [1988]. It seems natural to assume that the *validation of plans* is not harder than planning. Thus, one would expect that plan validation is easy for SAS-PUS plans and perhaps for an even larger class of plans. Our NP-hardness result for the simple temporal projection problem seems to suggest the contrary, though. One of the most problematical points in the definition of the temporal projection problem by Dean and Boddy seems to be that event sequences are permitted to contain events that do not affect the world because their preconditions are not satisfied. If we define the plan validation problem in a way such that all possible event sequences have to contain only events that affect the world, plan validation is tractable for the class of plans containing only unconditional events, a point already suggested by Chapman [1987]. In fact, deciding a conjunction of temporal projection problems that is equivalent to the plan validation problem appears to be easier than deciding each conjunct in isolation.

These reflections lead to the question of whether the above mentioned incomplete decision procedure gives useful results. One would expect that the procedure is complete in important special cases that are tractable. However, the procedure sometimes fails on unconditional nonlinear plans, where plan validation is tractable.

Finally, we will discuss the relationships between temporal projection and story understanding. If we add two reasonable assumptions, namely, that a story is *coherent* and *non-repeating*, then it is trivial to reconstruct the underlying course of events in polynomial time in the case where we have proven (contrary to previous conjecture) that temporal projection is NP-hard. This positive complexity result does not carry over to slight generalizations, however. The modified temporal projection problem for coherent and non-repeating stories becomes NP-hard, if general unconditional events are permitted. However, it seems to be the case in story understanding that more constraints are exploited than can be captured by the original formalization of temporal projection.

The remainder of the paper is structured as follows. Section 2 contains the definition of the general temporal projection problem for partially ordered events. In Section 3, the computational complexity of a *simple* form of temporal projection that was conjectured to be tractable by Dean and Boddy [1988] is shown to be NP-hard. The corresponding planning problem permits a polynomial planning algorithm, however, as is shown in Section 4. This positive result is shown to hold for even less restricted planning problems in Section 5. In Section 6 we use this result to show that optimal planning for blocks world is tractable under certain restrictions. In Section 7, we show that plan validation is tractable if all events are unconditional. The degree of completeness of the incomplete decision procedure mentioned above is analyzed in Section 8. Finally, we sketch some ideas concerning the relationship between temporal projection and story understanding in Section 9.

2 Temporal Projection

Given a description of the state of the world and a description of which events will occur, we are usually able to predict what the world will look like. This kind of reasoning is called *temporal projection.* It seems to be the easiest and most basic kind of temporal reasoning. Depending on the representation, however, there are subtle difficulties hidden in this reasoning task.

The formalization of the temporal projection problem for partially ordered events given below closely follows the presentation by Dean and Boddy [1988, Sect. 2]. We start with the definition of what a *causal structure* is, which fixes our vocabulary to talk about states, event types, and laws of change. We confine ourselves to a particular simple form of causal structures, where world states are represented by sets of propositional atoms and laws of change are described as propositional STRIPS-like operators. As a second step, we introduce *sets of partially ordered events* over causal structures that denote all event sequences that satisfy the partial order over the event set. Finally, the notion of *event systems* will be introduced that consist of an initial state and a partially ordered event set. The problem of temporal projection is to decide whether a given propositional atom holds, possibly or necessarily, after a given event in an event system.

Definition 1 *A* **causal structure** *is given by a tuple* $\Phi = \langle \mathcal{P}, \mathcal{E}, \mathcal{R} \rangle$*, where*

- $\mathcal{P} = \{p_1, \ldots, p_n\}$ *is a set of propositional atoms, the* **conditions***,*
- $\mathcal{E} = \{\epsilon_1, \ldots, \epsilon_m\}$ *is a set of* **event types***,*
- $\mathcal{R} = \{r_1, \ldots, r_o\}$ *is a set of* **causal rules** *of the form* $r_i = \langle \epsilon_i, \varphi_i, \alpha_i, \delta_i \rangle$*, where*
  - $\epsilon_i \in \mathcal{E}$ *is the* **triggering event type***,*
  - $\varphi_i \subseteq \mathcal{P}$ *is a set of* **preconditions***,*
  - $\alpha_i \subseteq \mathcal{P}$ *is the* **add list***,*
  - *and* $\delta_i \subseteq \mathcal{P}$ *is the* **delete list***.*

In order to give an example, assume a toy scenario with a hall, a room A, and another room B. Room A contains a public phone, and room B contains an electric outlet. The robot Robby can be in the hall (denoted by the atom `h`), in room A (`a`), or in room B (`b`). Robby can have a phone card (`p`) or coins (`c`). Additionally, when Robby uses the phone, he can inform his master on the phone that everything is in order (`i`). Robby can be fully charged (`f`), almost empty (`e`), or, in unlucky circumstances, his batteries can be damaged (`d`). Summarizing, the set of conditions for our tiny causal structure is the following:

$$\mathcal{P} = \{\mathtt{a}, \mathtt{b}, \mathtt{h}, \mathtt{p}, \mathtt{c}, \mathtt{i}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}.$$

Robby can do the following. He can move from the hall to either room ($\epsilon_{h \to a}$, $\epsilon_{h \to b}$) and *vice versa* ($\epsilon_{a \to h}$, $\epsilon_{b \to h}$). Provided he is in room `a` and he has a phone card or coins, he can call his master (ϵ_{call}). Additionally, if Robby is in room `b`, he can recharge himself (ϵ_{charge}). However, if Robby is already fully charged, this results in damaging his batteries. Summarizing, we have the following set of event types:

$$\mathcal{E} = \{\epsilon_{h \to a}, \epsilon_{h \to b}, \epsilon_{a \to h}, \epsilon_{b \to h}, \epsilon_{call}, \epsilon_{charge}\},$$

and the following set of causal rules:

$$
\begin{aligned}
\mathcal{R} = \{ & \langle \epsilon_{h\to a}, & \{\mathtt{h}\}, & \{\mathtt{a}\}, & \{\mathtt{h}\}\rangle, \\
& \langle \epsilon_{h\to b}, & \{\mathtt{h}\}, & \{\mathtt{b}\}, & \{\mathtt{h}\}\rangle, \\
& \langle \epsilon_{a\to h}, & \{\mathtt{a}\}, & \{\mathtt{h}\}, & \{\mathtt{a}\}\rangle, \\
& \langle \epsilon_{b\to h}, & \{\mathtt{b}\}, & \{\mathtt{h}\}, & \{\mathtt{b}\}\rangle, \\
& \langle \epsilon_{call}, & \{\mathtt{a},\mathtt{p}\}, & \{\mathtt{i}\}, & \emptyset\rangle, \\
& \langle \epsilon_{call}, & \{\mathtt{a},\mathtt{c}\}, & \{\mathtt{i}\}, & \{\mathtt{c}\}\rangle, \\
& \langle \epsilon_{charge}, & \{\mathtt{b},\mathtt{e}\}, & \{\mathtt{f}\}, & \{\mathtt{e}\}\rangle, \\
& \langle \epsilon_{charge}, & \{\mathtt{b},\mathtt{f}\}, & \{\mathtt{d}\}, & \{\mathtt{f}\}\rangle\}.
\end{aligned}
$$

In order to talk about sets of concrete events and temporal constraints over them, the notion of a *partially ordered event set* is introduced.[1]

Definition 2 *Assuming a causal structure $\Phi = \langle \mathcal{P}, \mathcal{E}, \mathcal{R} \rangle$, a* **partially ordered event set** *(POE) over Φ is a pair $\Delta_\Phi = \langle \mathcal{A}_\Phi, \prec \rangle$ consisting of a set of* **actual events** *$\mathcal{A}_\Phi = \{e_1, \ldots, e_p\}$ such that $type(e_i) \in \mathcal{E}$, and a strict partial order[2] $\prec$ over $\mathcal{A}_\Phi$.*

In the following, we will often drop the subscript Φ in Δ_Φ and $\mathcal{A}_\Phi$ if it is clear from the context which causal structure we mean. Continuing our example, we assume a set of six actual events $\mathcal{A} = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}\}$, such that

$$
\begin{aligned}
type(\mathtt{A}) &= \epsilon_{h\to a} \\
type(\mathtt{B}) &= \epsilon_{call} \\
type(\mathtt{C}) &= \epsilon_{a\to h} \\
type(\mathtt{D}) &= \epsilon_{h\to b} \\
type(\mathtt{E}) &= \epsilon_{charge} \\
type(\mathtt{F}) &= \epsilon_{b\to h},
\end{aligned}
$$

and

$$
\begin{aligned}
\mathtt{A} \prec \mathtt{B} \prec \mathtt{C} \\
\mathtt{D} \prec \mathtt{E} \prec \mathtt{F}.
\end{aligned}
$$

POEs denote sets of possible sequences of events satisfying the partial order. A **partial event sequence** of length m over such a POE $\langle \mathcal{A}, \prec \rangle$ is a sequence $\mathbf{f} = \langle f_1, \ldots, f_m \rangle$ such that (1) $\{f_1, \ldots, f_m\} \subseteq \mathcal{A}$, (2) $f_i \neq f_j$ if $i \neq j$, and (3) for each pair f_i, f_j of events appearing in $\mathbf{f}$, if $f_i \prec f_j$ then $i < j$. For instance, $\langle \mathtt{A}, \mathtt{B}, \mathtt{C} \rangle$ is a partial event sequence of length three over the POE given above, while $\langle \mathtt{A}, \mathtt{C}, \mathtt{B} \rangle$ is not. If the event sequence is of length $|\mathcal{A}|$, it is called a **complete event sequence** over the POE. The sequences

[1] This notion is similar to the notion of a *nonlinear plan*.

[2] A strict partial order is a transitive and irreflexive relation.

$\langle$A, B, C, D, E, F$\rangle$ and $\langle$A, D, B, E, C, F$\rangle$ are complete event sequences, for instance. The set of all complete event sequences over a POE Δ is denoted by $CS(\Delta)$.

We say that a partial event sequence $\mathbf{f}$ can be **extended** to an event sequence $\mathbf{g}$ if $|\mathbf{f}| < |\mathbf{g}|$ and for all f_i, f_j with $i < j$ there exists $g_k = f_i$ and $g_l = f_j$ such that $k < l$. If $\mathbf{f} = \langle f_1, \ldots, f_k, \ldots, f_m \rangle$ is an event sequence, then $\langle f_1, \ldots, f_k \rangle$ is the initial sequence of $\mathbf{f}$ up to f_k, written $\mathbf{f}/f_k$. Similarly, $\mathbf{f} \backslash f_k$ denotes the initial sequence $\langle f_1, \ldots, f_{k-1} \rangle$ consisting of all events before f_k. Further, we write $g; \mathbf{f}$ to denote the sequence $\langle g, f_1, \ldots, f_m \rangle$ and $\mathbf{f}; g$ to denote $\langle f_1, \ldots, f_m, g \rangle$.

Each event maps states (subsets of $\mathcal{P}$) to states. Let $S \subseteq \mathcal{P}$ denote a state and let e be an event. Then we say that the causal rule r is **applicable** in state S iff $r = \langle type(e), \varphi, \alpha, \delta \rangle$ and $\varphi \subseteq S$. Given e and S, $app(S, e)$ denotes the set of all **applicable rules** for e in state S. An event e is said to **affect** the world in a state S iff $app(S, e) \neq \emptyset$. In order to simplify notation, we write $\varphi(r)$, $\alpha(r)$, $\delta(r)$ to denote the sets φ, α, and δ, respectively, appearing in the rule $r = \langle \epsilon, \varphi, \alpha, \delta \rangle$. If there is only one causal rule associated with the event type $type(e)$, we will also use the notation $\varphi(e)$, $\alpha(e)$, and $\delta(e)$. Based on this notation, we define what we mean by the *result* of a sequence of events relative to a state S.

Definition 3 *The function "Result" from states and event sequences to states is defined recursively by:*

$$\begin{aligned} Result\big(S, \langle\rangle\big) &= S \\ Result\big(S, (\mathbf{f}; g)\big) &= Result(S, \mathbf{f}) - \{\delta(r) |\, r \in app(Result(S, \mathbf{f}), g)\} \cup \\ & \qquad \{\alpha(r) |\, r \in app(Result(S, \mathbf{f}), g)\}. \end{aligned}$$

It is easy to verify that the following equation holds for our example scenario:

$$Result(\{\mathtt{h}, \mathtt{e}, \mathtt{c}\}, \langle \mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F} \rangle) = \{\mathtt{h}, \mathtt{f}, \mathtt{i}\}.$$

The definition of the function *Result* permits sequences of events where events occur that do not affect the world. For instance, it is possible to ask what the result of $\langle$A, D, B, E, C, F$\rangle$ in state $\{$h, e, c$\}$ will be:

$$Result(\{\mathtt{h}, \mathtt{e}, \mathtt{c}\}, \langle \mathtt{A}, \mathtt{D}, \mathtt{B}, \mathtt{E}, \mathtt{C}, \mathtt{F} \rangle) = \{\mathtt{h}, \mathtt{e}, \mathtt{i}\}.$$

Although perfectly well-defined, this result seems to be strange because the events D, E, and F occurred without having any effect on the state of the world. Given a state S, we will often restrict our attention to event sequences such that all events affect the world. These sequences are called **admissible event sequences** relative to the state S. The set of all complete event sequences over Δ that are admissible relative to S are denoted by $ACS(\Delta, S)$. If $CS(\Delta) = ACS(\Delta, S)$, we will say that Δ is **coherent** relative to S.

In the following, we will often talk about which consequences a POE will have on some initial state. For this purpose, the notion of an *event system* is introduced.

Definition 4 *An* **event system** Θ *is a pair* $\langle \Delta_\Phi, \mathcal{I} \rangle$*, where* Δ_Φ *is a POE over the causal structure* $\Phi = \langle \mathcal{P}, \mathcal{E}, \mathcal{R} \rangle$*, and* $\mathcal{I} \subseteq \mathcal{P}$ *is the* **initial state.**

In order to simplify notation, the functions *CS* and *ACS* are extended to event systems with the obvious meaning, i.e., $CS(\langle \Delta, S \rangle) = CS(\Delta)$ and $ACS(\langle \Delta, S \rangle) = ACS(\Delta, S)$. Further, if $CS(\Theta) = ACS(\Theta)$, Θ is called coherent.

The problem of temporal projection as formulated by Dean and Boddy [1988] is to determine whether some condition holds, *possibly* or *necessarily*, after a particular event of an event system.

Definition 5 *Given an event system* Θ*, an event* $e \in \mathcal{A}$*, and a condition* $p \in \mathcal{P}$*:*

$$\begin{aligned} p \in Poss(e, \Theta) \quad &\textit{iff} \quad \exists \mathbf{f} \in CS(\Theta)\colon p \in Result(\mathcal{I}, \mathbf{f}/e) \\ p \in Nec(e, \Theta) \quad &\textit{iff} \quad \forall \mathbf{f} \in CS(\Theta)\colon p \in Result(\mathcal{I}, \mathbf{f}/e). \end{aligned}$$

Continuing our example, let us assume the initial state $\mathcal{I} = \{h, e, c\}$. Then the following can be easily verified:

$$\begin{array}{llll} \mathtt{i} \in Poss(\mathtt{B}, \Theta) & \qquad & \mathtt{i} \notin Nec(\mathtt{B}, \Theta) \\ \mathtt{d} \notin Poss(\mathtt{E}, \Theta) & \qquad & \mathtt{d} \notin Nec(\mathtt{E}, \Theta). \end{array}$$

In plain words, Robby is only possibly but not necessarily successful in calling his master. On the positive side, however, we know that Robby's batteries will not be damaged, regardless of in which order the events happen.

Given a set of conditions S and a sequence $\mathbf{f}$, $Result(S, \mathbf{f})$ can easily be computed in polynomial time. Since the set $CS(\Theta)$ may contain exponentially many sequences, however, it is not obvious whether $p \in Poss(e, \Theta)$ and $p \in Nec(e, \Theta)$ can be decided in polynomial time.

3 A "Simple" Temporal Projection Problem

In the general case, temporal projection is quite difficult. Dean and Boddy [1988] show that the decision problems $p \in Poss(e, \Theta)$ and $p \in Nec(e, \Theta)$ are NP-complete and co-NP-complete respectively even under some severe restrictions, such as restricting α or δ to be empty for all rules, or requiring that there is only one causal rule associated with each event type.

Definition 6 *An event system is called* **unconditional** *iff for each $\epsilon \in \mathcal{E}$, there exists only one causal rule with the triggering event type ϵ. An event system is called* **almost simple** *iff it is unconditional and for each causal rule $r = \langle \epsilon, \varphi, \alpha, \delta \rangle$, the sets α and δ are singletons and $\delta \subseteq \varphi$. An event system is called* **simple** *iff it is unconditional, $\mathcal{I}$ is a singleton, and for each causal rule $r = \langle \epsilon, \varphi, \alpha, \delta \rangle$, the sets φ, α, and δ are singletons and $\varphi = \delta$.*

Dean and Boddy [1988, Theorem 2.4] prove that temporal projection for almost simple event systems is NP-hard and conjecture that it is a polynomial-time problem for simple event systems [Dean and Boddy, 1988, p. 379]. As it turns out, however, also this problem is computationally difficult.

Theorem 1 *Deciding $p \in Poss(e, \Theta)$ for simple event systems Θ is* NP-*complete.*

Proof. Membership in NP is obvious. Guess an event sequence $\mathbf{f}$ and verify in polynomial time that $\mathbf{f} \in CS(\Theta)$ and $p \in Result(\mathcal{I}, \mathbf{f}/e)$.

In order to prove NP-hardness, we give a polynomial transformation from **path with forbidden pairs** to the temporal projection problem. The former problem is defined as follows:

> Given a directed graph $G = (V, A)$, two vertices $s, t \in V$, and a collection $C = \big\{\{a_1, b_1\}, \ldots, \{a_n, b_n\}\big\}$ of pairs of arcs from A, is there a directed path from s to t in G that contains at most one arc from each pair in C?

This problem is NP-complete, even if the graph is acyclic and all pairs are disjoint [Garey and Johnson, 1979, p. 203].

First of all, we specify a transformation from *directed acyclic graphs* (DAG) to simple event systems. Let $G = (V, A)$ be a DAG, where $V = \{v_1, \ldots, v_k\}$. Then define

$$
\begin{aligned}
\mathcal{P} &= \{v_1, \ldots, v_k\} \cup \{*\} \\
\mathcal{E} &= \{\epsilon_{i,j} |\, (v_i, v_j) \in A\} \cup \{\epsilon_*\} \\
\mathcal{R} &= \{\langle \epsilon_{i,j}, \{v_i\}, \{v_j\}, \{v_i\} \rangle |\, (v_i, v_j) \in A\} \cup \\
&\quad\ \{\langle \epsilon_*, \{*\}, \{*\}, \{*\} \rangle\} \\
\mathcal{A} &= \{e_{i,j} |\, \epsilon_{i,j} \in \mathcal{E}\} \cup \{e_*\} \\
type(e_{i,j}) &= \epsilon_{i,j} \text{ for all } e_{i,j} \in \mathcal{A} - \{e_*\} \\
type(e_*) &= \epsilon_* \\
e \prec e_* &\quad\ \text{for all } e \in \mathcal{A} - \{e_*\}.
\end{aligned}
$$

Note that such event systems, which we will call DAG event systems, are simple, provided $|\mathcal{I}| = 1$.

Let $G = (V, A)$ be a DAG, let $C = \{\{a_1, b_1\}, \ldots, \{a_n, b_n\}\}$ be a collection of "forbidden pairs" of arcs from A such that each pair consists of different arcs and the pairs are pairwise disjoint. Further, let s and t be two vertices from V and assume without loss of generality that there is no arc $(t, v_i) \in A$.

Let Θ be the corresponding DAG event system with $\mathcal{I} = \{s\}$. For each pair of arcs $\{(v_i, v_j), (v_k, v_l)\} \in C$,

1. if there is a (possibly empty) path from v_j to v_k in G add $e_{k,l} \prec e_{i,j}$ as a temporal constraint to Θ,

2. if there is a (possibly empty) path from v_l to v_i in G, add $e_{i,j} \prec e_{k,l}$ as a temporal constraint to Θ.

Note that this addition of temporal constraints can be done in polynomial time. Further note that it is impossible that (1) and (2) applies simultaneously to a pair of arcs. Finally note that since the forbidden pairs are pairwise disjoint, there is no set of events $\{f_1, f_2, f_3\} \subseteq \mathcal{A}$ such that $f_1 \prec f_2 \prec f_3$.

For the resulting event system, we claim that there is a path from s to t in G that contains at most one arc from each pair in C iff $t \in \mathit{Poss}(e_*, \Theta)$.

"$\Rightarrow$": Let $v_1, \ldots, v_m$, $1 \leq m \leq |V|$, be a path in G, where $v_1 = s$ and $v_m = t$, without forbidden pairs from C. Then by construction of Θ, there exists a sequence of events $\mathbf{g} = \langle g_1, \ldots, g_{m-1} \rangle$ such that $\langle \mathit{type}(g_i), \{v_i\}, \{v_{i+1}\}, \{v_i\} \rangle \in \mathcal{R}$. Note that this sequence is indeed a partial event sequence over Θ because the path does not contain forbidden pairs, and, hence there are no temporal constraints for the events appearing in $\mathbf{g}$. Furthermore, we have for $\alpha(g_{m-1}) = \{t\}$. By the construction of Θ, it holds that

$$\mathit{Result}(\mathcal{I}, (\mathbf{g}; e_*)/e_*) = \{t\}.$$

The sequence $\mathbf{g}; e_*$ can be extended to a complete event sequence $\mathbf{h}$ over Θ in the following way:

1. add all events f that are not temporally constrained and do not appear in $\mathbf{g}$ immediately before e_*,

2. add all pairs of events f, f' such that $f \prec f'$ and that do not appear in $\mathbf{g}$ immediately before e_* respecting $\prec$,

3. add all events f that do not appear in $\mathbf{g}$ and $f \prec g_i$ for some g_i appearing in $\mathbf{g}$ immediately before g_i,

4. add all events f that do not appear in $\mathbf{g}$ and $g_i \prec f$ for some g_i appearing in $\mathbf{g}$ immediately after g_i.

Note that for extensions of the forms (1) and (2) it holds trivially that

$$\mathit{Result}(\mathcal{I}, \mathbf{h}/e_*) = \{t\} \text{ iff } \mathit{Result}(\mathcal{I}, (\mathbf{g}; e_*)/e_*) = \{t\}$$

since no precondition of any rule contains t by assumption. For extensions of the form (3) it holds that $e_{i,j} \prec e_{k,l}$ only if there is path from v_l to v_i in G. Hence, if $e_{i,j}$ is placed immediately before $e_{k,l}$, the precondition of the causal rule associated with $e_{i,j}$ cannot be satisfied. Thus, the above equivalence also holds for case (3). Since (4) is the converse case, the equivalence also holds.

Summarizing, we have for the complete event sequence **h**

$$Result(\mathcal{I}, \mathbf{h}/e_*) = \{t\}.$$

Thus, $t \in Poss(e_*, \Theta)$.

"$\Leftarrow$": Assume $t \in Poss(e_*, \Theta)$. Then there exists a complete event sequence **g** such that

$$Result(\mathcal{I}, \mathbf{g}/e_*) = \{t\}.$$

Consider the subsequence **h** containing only events that affect the world:

$$\mathbf{h} = \langle h_1, h_2, \ldots, h_{m-1} \rangle.$$

By the construction of the causal rules in Θ and the structure of the initial set it is evident that each event in the subsequence **h** has an add list that is identical to the preconditions of the immediately following event. Since the initial conditions are $\mathcal{I} = \{s\}$ and $Result(\mathcal{I}, \mathbf{h}) = \{t\}$, there must be a path $s = v_1, v_2, \ldots, v_m = t$ in G.

Finally, this path cannot contain any forbidden pair. Assume the contrary, i.e., the path is of the form $s, \ldots, v_i, v_j, \ldots, v_k, v_l, \ldots, t$ and $\{(v_i, v_j), (v_k, v_l)\} \in C$. Thus, there is a path from v_j to v_k. In this case, however, we have $e_{k,l} \prec e_{i,j}$ by the construction of $\prec$ in Θ. This means, however, that **h** cannot be a possible event sequence over Θ. Hence, there cannot be any event sequences leading to t that contain forbidden pairs. ■

It is easy to show that $p \in Nec(e, \Theta)$ is computationally equivalent to $p \notin Poss(e, \Theta)$, i.e., co-NP-complete.

Corollary 2 *Deciding $p \in Nec(e, \Theta)$ for simple event systems Θ is co-NP-complete.*

Proof. We show that $p \notin Nec(e, \Theta)$ is NP-complete. Membership in NP is obvious. For the NP-hardness part, we start with the same transformation as in the proof of Theorem 1. We add to Θ a new condition p and a number of events f with associated causal rules of the form:

$$\langle type(f), \{v\}, \{p\}, \{v\} \rangle,$$

for all $v \in V - \{t\}$. These events are constrained to happen before e_* and after all other events constructed in the above reduction.

Now, it follows by the same arguments as in the proof of Theorem 1 that $p \notin Nec(e_*, \Theta)$ iff there is a path from s to t without forbidden pairs. ■

These results are somewhat surprising because one might suspect that planning and story understanding are easy under the restrictions imposed on the structure of event systems. We will analyze this point more thoroughly in the following sections.

4 A Simple Planning Problem

One reason for analyzing the temporal projection problem is that it seems to constitute the heart of plan validation [Dean and Boddy, 1988, p. 378]. If we now consider the restrictions placed on the simple problem, it turns out that planning itself—a problem one would expect to be harder than validation—is quite easy.

In the context of planning, events as introduced above are usually called **actions** and POEs are called **nonlinear plans**, or simply **plans**. In the following, we use these terms interchangeably.

Definition 7 *A* **planning task** Π *is given by* $\langle \Phi, \mathcal{I}, \mathcal{G} \rangle$, *where* $\Phi = \langle \mathcal{P}, \mathcal{E}, \mathcal{R} \rangle$ *is a causal structure as defined above, and* $\mathcal{I} \subseteq \mathcal{P}$ *and* $\mathcal{G} \subseteq \mathcal{P}$ *are the* **initial state** *and* **goal state** *respectively. A* **plan** Δ_Φ **solves** Π *iff (1)* $\mathcal{G} \subseteq Result(\mathcal{I}, \mathbf{f})$ *for all* $\mathbf{f} \in CS(\Delta_\Phi)$, *and (2)* $ACS(\Delta_\Phi, \mathcal{I}) = CS(\Delta_\Phi)$. *A solution* $\Delta = \langle \mathcal{A}, \prec \rangle$ *for* Π *is* **minimal** *iff for all other solutions* $\Delta' = \langle \mathcal{A}', \prec' \rangle$, *it holds that* $|\mathcal{A}| \leq |\mathcal{A}'|$.

The computational complexity of planning has been investigated only recently. Bylander [1991] analyzed the general problem of deciding the existence of a solution for a planning task in the context of propositional STRIPS-like representations and showed that the general problem is PSPACE-complete.[3] A number of restricted problems turn out to be tractable, however. For instance, planning with unconditional causal structures and causal rules restricted by $|(\alpha(r) \cup \delta(r))| = 1$ is tractable [Bylander, 1991, Theorem 7]. Similarly, planning with causal rules such that the preconditions are always empty [Bylander, 1991, Theorem 9] and planning with unconditional causal structures such that the goal state is restricted in size and all rules contain only one precondition [Bylander, 1991, Theorem 8] are tractable. It should be noted, however, that Bylander considers only the *existence* problem and not the associated *optimization* problem of finding minimal plans,

[3] The representation of causal rules in [Bylander, 1991] is a little more powerful. Preconditions can also be negative, i.e., refer to the *absence* of atoms. However, the hardness result applies to our case as well [Bylander, 1991, Corollary 2]. Moreover, all results about positive preconditions can be easily adapted to our formalism.

which is often harder. For example, his Theorem 9 does not apply to the corresponding optimization problem.

Proposition 3 *Deriving minimal plans for planning tasks such that the preconditions of all rules are empty is* NP*-equivalent.*

Proof. The corresponding decision problem of deciding the existence of solutions of a given length is obviously in NP. A straightforward reduction from *minimum cover* [Garey and Johnson, 1979, p. 222] shows NP-completeness of the decision problem. From that the proposition follows immediately. ■

Returning to the problem we analyzed in the previous section, similarly to *simple event systems* we define **simple planning tasks** to be planning tasks that meet the following restrictions: (1) there is only one causal rule associated with each event type, (2) for all causal rules $|\varphi| = |\alpha| = |\delta| = 1$ and $\varphi = \delta$, and (3) $|\mathcal{I}| = 1$. Using Bylander's [1991] Theorem 8, the tractability of the solution existence problem follows immediately. In this case, also plan derivation is tractable, however.

Proposition 4 *For simple planning tasks, it can be decided in polynomial time whether there exists a solution. Further, a minimal valid plan can be derived in polynomial time.*

Proof. Given a simple planning task $\Pi = \big\langle \langle \mathcal{P}, \mathcal{E}, \mathcal{R} \rangle, \{s\}, \{t\} \big\rangle$, construct a directed graph $G = (V, A)$ as follows. Let

$$\begin{aligned} V &= \mathcal{P}, \\ A &= \{(v, w) | \langle \epsilon, \{v\}, \{w\}, \{v\} \rangle \in \mathcal{R}\}. \end{aligned}$$

Then the derivation of a minimal solution for Π reduces to finding a shortest path from s to t in G, which can be done in polynomial time. ■

This result leads to the question why temporal projection, which is supposed to be the underlying problem in plan validation, is more difficult than planning itself in some cases. One explanation could be that a planner could create the complicated structure we used in the proof of Theorem 1, but it never would do so. Hence, the theoretical complexity never shows up in reality. This explanation is unsatisfying, however. If this would be really the case, we should be able to characterize the structure of the nonlinear plans planning systems create and validate. As is shown in Section 7, the problem is more subtle. Before we investigate the plan validation problem, however, we will analyze a different planning problem that turns out to be tractable, as well.

5 Polynomial-Time Planning in Two Different Formalisms

As we have seen, the simple planning problem defined in Section 4 is not the only planning problem known to be tractable. The results on tractable planning by Bylander [1991] have already been revised in the previous section. Bäckström and Klein [1991a; 1991b] have also presented results on tractable planning, which will be analyzed in this section. Furthermore, there are also results on average case tractability of planning using macro-operators or action hierarchies under certain assumptions [Korf, 1987], but it is out of the scope of this paper to discuss such approaches.

Bäckström and Klein presents two tractable planning problems: The SAS-PUBS problem [Bäckström and Klein, 1991b] and the SAS-PUS problem [Bäckström and Klein, 1991a]. Both problems properly subsume the simple planning problem defined in Section 4. In the following, we will only consider the SAS-PUS problem since it properly includes the SAS-PUBS problem. A direct comparison with the simple problem or Bylander's results is, however, not possible, since the SAS-PUS problem is defined in another formalism called *simplified actions structures* (SAS). Although the restrictions defining the SAS-PUS problem are possible to express in the formalism used in the rest of this paper, they are hardly obvious to come up with from the viewpoint of that formalism. On the other hand, they appear quite natural in the SAS formalism. This indicates that the choice of modelling formalism can strongly influence how one defines problems. The rest of this section is devoted to redefining the restrictions of the SAS-PUS problem in the formalism used in the rest of this paper, yielding the SAS-PUS equivalent problem, and prove that this new problem is tractable. Since the proof of tractability is based on transformation to the SAS-PUS problem, it is unavoidable to first present the SAS formalism. This presentation, however, will, be very brief and conform as closely as possible to the other formalism. The main differences are that the planning world is modelled in a somewhat more structured way than just a set of propositions, and that actions are modelled somewhat differently.

In analogy with the concepts causal structure and planning task, the corresponding concepts *causal SAS-structure* and *SAS planning task* are introduced.

Definition 8 *A* **causal SAS-structure** $\Phi = \langle \mathcal{M}, \mathcal{S}_1, \ldots, \mathcal{S}_{|\mathcal{M}|}, \mathcal{E}, \mathcal{C} \rangle$ *is given by:*

- *a set of* **state variable indices**, $\mathcal{M} = \{1, \ldots, m\}$;
- *for each* $i \in \mathcal{M}$, *a* **domain** $\mathcal{S}_i$ *of mutually exclusive values for the ith state variable, implicitly defining*

- *for each* $i \in \mathcal{M}$, *an* **extended domain** $\mathcal{S}_i^+ = \mathcal{S}_i \cup \{u\}$ *where* u *denotes the* **undefined value**,
- *a set of* **total states** $\mathcal{S} = \mathcal{S}_1 \times \ldots \times \mathcal{S}_m$, *and*
- *a set of* **partial states** $\mathcal{S}^+ = \mathcal{S}_1^+ \times \ldots \times \mathcal{S}_m^+$;

- *a set of* **action types** $\mathcal{E} = \{\epsilon_1, \ldots, \epsilon_n\}$;
- *a set of* **causal SAS-rules** $\mathcal{C} = \{c_1, \ldots, c_n\}$ *of the form* $c_i = \langle \epsilon_i, \mathrm{b}(\epsilon_i), \mathrm{e}(\epsilon_i), \mathrm{f}(\epsilon_i) \rangle$ *where*
  - $\epsilon_i \in \mathcal{E}$ *is the* **triggering event type**,
  - $\mathrm{b}(\epsilon_i) \in \mathcal{S}^+$ *is the* **precondition**,
  - $\mathrm{e}(\epsilon_i) \in \mathcal{S}^+$ *is the* **postcondition**, *and*
  - $\mathrm{f}(\epsilon_i) \in \mathcal{S}^+$ *is the* **prevailcondition**;

  $\mathcal{C}$ *must also satisfy the restrictions:*

  S1. for all $\epsilon \in \mathcal{E}$ *and for all* $i \in \mathcal{M}$, *either* $\mathrm{b}(\epsilon)[i] = \mathrm{e}(\epsilon)[i] = u$ *or* $u \neq \mathrm{b}(\epsilon)[i] \neq \mathrm{e}(\epsilon)[i] \neq u$;

  S2. for all $\epsilon \in \mathcal{E}$ *and for all* $i \in \mathcal{M}$, *either* $\mathrm{b}(\epsilon)[i] = u$ *or* $\mathrm{f}(\epsilon)[i] = u$; *and*

  S3. for all $\epsilon, \epsilon' \in \mathcal{E}$, *if* $\mathrm{b}(\epsilon) = \mathrm{b}(\epsilon')$, $\mathrm{e}(\epsilon) = \mathrm{e}(\epsilon')$, *and* $\mathrm{f}(\epsilon) = \mathrm{f}(\epsilon')$ *then* $\epsilon = \epsilon'$.

  where $s[i]$ *denotes the value of the* i*th state variable in* s.

A **SAS planning task** $\Pi = \langle \Phi, s_{\mathcal{I}}, s_{\mathcal{G}} \rangle$ *is given by:*

- *A causal* SAS*-structure* Φ,
- *an* **initial state** $s_{\mathcal{I}} \in \mathcal{S}$, *and*
- *a* **goal state** $s_{\mathcal{G}} \in \mathcal{S}$.

For any action, the conditions of the causal SAS-rule triggered by the corresponding action type are interpreted as follows: The pre- and post-conditions express which state variables are changed by the action, and what values these state variables *must* have at the beginning of the action and *will* have at the end of the action respectively. The prevailcondition expresses which state variables *must* have a certain value during the whole execution of the action but which are not changed by the action. The restrictions express that an action can only change a state variable from a defined value to another defined value (S1), a state variable cannot both be changed and required to have a constant value (S2), and two distinct action types must differ in at least one of their conditions (S3). The interested reader is referred to the original papers [Bäckström and Klein, 1991a; Bäckström and Klein, 1991b] for further details and intuition regarding the SAS formalism.

If $s[i] = u$ for some state $s \in \mathcal{S}^+$, then the value of the ith state variable in s is treated as irrelevant or unknown. For $s, s' \in \mathcal{S}^+$, $s \sqsubseteq s'$ denotes that for all $i \in \mathcal{M}$, either $s[i] = u$ or $s[i] = s'[i]$. As a convention, we also write $\mathrm{b}(e)$, $\mathrm{e}(e)$, and $\mathrm{f}(e)$ meaning $\mathrm{b}(\epsilon)$, $\mathrm{e}(\epsilon)$, and $\mathrm{f}(\epsilon)$ respectively, where $\epsilon = type(e)$. Given an action e and a state $s \in \mathcal{S}^+$, $app_{\mathrm{SAS}}(s, e)$ denotes the set of applicable causal SAS-rules for e in s, that is, all rules $\langle \epsilon, \mathrm{b}(\epsilon), \mathrm{e}(\epsilon), \mathrm{f}(\epsilon) \rangle$ s.t. $type(e) = \epsilon$, $\mathrm{b}(\epsilon) \sqsubseteq s$, and $\mathrm{f}(\epsilon) \sqsubseteq s$. Note that by the definition of causal SAS-structures, $app_{\mathrm{SAS}}(s, e)$ is either empty or a singleton. An action e is said to **SAS-affect** the world in a state s iff $app_{\mathrm{SAS}}(s, e) \neq \emptyset$. A state s is **updated** by another state s', written $s \oplus s'$ and defined as follows:[4]

$$(s \oplus s')[i] = \begin{cases} s'[i] & \text{if } s'[i] \neq u \\ s[i] & \text{otherwise} \end{cases} \quad \text{for all } i \in \mathcal{M}.$$

Event sequences are defined as previously, and the function $Result_{\mathrm{SAS}}$ is defined recursively as:

$$Result_{\mathrm{SAS}}\big(s, \langle\rangle\big) = s$$

$$Result_{\mathrm{SAS}}\big(s, (\mathbf{f}; g)\big) =$$

$$\begin{cases} Result_{\mathrm{SAS}}(s, \mathbf{f}) \oplus \mathrm{e}(g) & \text{if } app_{\mathrm{SAS}}(Result_{\mathrm{SAS}}(s, \mathbf{f}), g) \neq \emptyset \\ Result_{\mathrm{SAS}}(s, \mathbf{f}) & \text{otherwise.} \end{cases}$$

An event sequence $\mathbf{f}$ is **SAS-admissible** relative to a state s if all actions in $\mathbf{f}$ SAS-affect the world when $\mathbf{f}$ is applied in s. Analogously to the definition of ACS, given a state $s \in \mathcal{S}^+$, $ACS_{\mathrm{SAS}}(\langle \mathcal{A}, \prec \rangle, s)$ denotes the set of all $\mathbf{f} \in CS(\langle \mathcal{A}, \prec \rangle)$ s.t. $\mathbf{f}$ is SAS-admissible relative to s.

Nonlinear SAS plans are defined analogously with nonlinear plans, and the *SAS-PUS planning problem* is defined as a more restricted version of the SAS planning problem.

Definition 9 *A tuple* $\Delta = \langle \mathcal{A}, \prec \rangle$ *is a* **nonlinear SAS-plan** *for a* SAS *planning task* $\Pi = \langle \langle \mathcal{M}, \mathcal{S}_1, \ldots, \mathcal{S}_{|\mathcal{M}|}, \mathcal{E}, \mathcal{C} \rangle, s_{\mathcal{I}}, s_{\mathcal{G}} \rangle$ *iff* $type(e) \in \mathcal{E}$ *for all* $e \in \mathcal{A}$, $\prec$ *is a strict partial order on* $\mathcal{A}$, $ACS_{\mathrm{SAS}}(\Delta, s_{\mathcal{I}}) = CS(\Delta)$, *and* $Result_{\mathrm{SAS}}(s_{\mathcal{I}}, \mathbf{f}) = s_{\mathcal{G}}$ *for all* $\mathbf{f} \in CS(\Delta)$.

Definition 10 *A* SAS *planning task* $\Pi = \langle \mathcal{M}, \mathcal{S}_1, \ldots, \mathcal{S}_{|\mathcal{M}|}, \mathcal{E}, \mathcal{C}, s_{\mathcal{I}}, s_{\mathcal{G}} \rangle$ *is a* **SAS-PUS planning task** *iff it satisfies the restrictions:*

SU. for all $\epsilon \in \mathcal{E}$, *there is exactly one* $i \in \mathcal{M}$ *s.t.* $\mathrm{b}(\epsilon)[i] \neq u$;

SP. for all $\epsilon, \epsilon' \in \mathcal{E}$, *if there is some* $i \in \mathcal{M}$ *s.t.* $\mathrm{e}(\epsilon)[i] = \mathrm{e}(\epsilon')[i] \neq u$ *then* $\epsilon = \epsilon'$; *and*

[4]Note that in SAS worlds the notions of *update* and *revision* as defined by Katsuno and Mendelzon [1991] coincide.

SS. for all $\epsilon, \epsilon' \in \mathcal{E}$ *and for all* $i \in \mathcal{M}$, *if* $\mathrm{f}(\epsilon)[i] \neq u$ *and* $\mathrm{f}(\epsilon')[i] \neq u$ *then* $\mathrm{f}(\epsilon)[i] = \mathrm{f}(\epsilon')[i]$

The restrictions SU, SP, and SS express that the set of action types must be *unary*, *post-unique*, and *single-valued* respectively.[5] Unariness means that each action changes exactly one state variable, i.e., an action cannot have multiple effects. Post-uniqueness means that there must not be two distinct action types changing the same state variable to the same value, i.e., no two distinct action types have the same effect. Single-valuedness means that if two distinct action types require the same state variable to have some constant, defined value during their executions, then they must require the same constant value for this variable. For example, if one action type requires the light to be on in a room during its execution, no other action type may require the light to be off during its execution.

We will now re-express the restrictions for the SAS-PUS planning problem in the formalism used in the rest of this paper. The resulting problem is called the *SAS-PUS equivalent problem.* We finally prove that minimal plans for the SAS-PUS equivalent problem can be derived in polynomial time.

Definition 11 *A planning task* $\Pi = \langle\langle\mathcal{P}, \mathcal{E}, \mathcal{R}\rangle, \mathcal{I}, \mathcal{G}\rangle$ *is* **SAS-PUS equivalent** *iff it satisfies the following restrictions:*

1. *There is exactly one causal rule for each event;*
2. $\mathcal{P}$ *can be partitioned into* m *disjoint subsets* $P_1, \ldots, P_m$ *s.t.* $|P_i| > 1$ *for* $1 \leq i \leq m$ *and for all causal rules* $\langle\epsilon, \varphi, \alpha, \delta\rangle \in \mathcal{R}$
   (a) $\delta \subseteq \varphi$,
   (b) $|\delta| = 1$;
   (c) $|\varphi \cap P_i| \leq 1$ *for all* i,
   (d) $|\alpha \cap P_i| = |\delta \cap P_i| \leq 1$ *for all* i,
   (e) $\alpha \cap \delta = \emptyset$, *and*
   (f) $|\mathcal{I} \cap P_i| = |\mathcal{G} \cap P_i| = 1$ *for all* i.
3. *for all pairs of causal rules* $\langle\epsilon, \varphi, \alpha, \delta\rangle, \langle\epsilon', \varphi', \alpha', \delta'\rangle \in \mathcal{R}$
   (a) *if* $\varphi = \varphi'$, $\alpha = \alpha'$, *and* $\delta = \delta'$ *then* $\epsilon = \epsilon'$;
   (b) *if* $\epsilon \neq \epsilon'$ *then* $\alpha \cap \alpha' = \emptyset$; *and*
   (c) *for all* $i \in \mathcal{M}$, *if* $(\varphi - \delta) \cap P_i \neq \emptyset$ *and* $(\varphi' - \delta') \cap P_i \neq \emptyset$ *then* $(\varphi - \delta) \cap P_i = (\varphi' - \delta') \cap P_i$.

[5]The acronym PUS is derived from the words post-unique, unary, and single-valued. The B in the acronym SAS-PUBS stands for *binary*, which means that all state variable domains must have exactly two defined values.

The restrictions 2b, 3b, and 3c correspond to unariness, post-uniqueness, and single-valuedness respectively. The requirement that $|P_i| > 1$ is not really a restriction; Suppose $P_i = \{p\}$, then we can extend it to a set $P_i' = \{p, \neg p\}$ and extend the conditions in the causal rules s.t. whenever an event type adds p it also deletes $\neg p$, and vice versa.

Theorem 5 *Minimal nonlinear plans for* SAS-PUS *equivalent planning tasks can be derived in polynomial time.*

Proof. Prove that any SAS-PUS equivalent planning task $\Pi = \langle\langle \mathcal{P}, \mathcal{E}, \mathcal{R}\rangle, \mathcal{I}, \mathcal{G}\rangle$ can be transformed to an equivalent SAS-PUS planning task $\Pi' = \langle\langle \mathcal{M}, \mathcal{S}_1, \ldots, \mathcal{S}_{|\mathcal{M}|}, \mathcal{E}, \mathcal{C}\rangle, s_\mathcal{I}, s_\mathcal{G}\rangle$ in polynomial time. The proof consists of three parts: first prove that there is a transformation from Π into Π' s.t. Π' is a SAS-PUS planning task, then prove that the solutions for Π' are exactly the solutions for Π, and, finally, prove that transforming Π into Π' and solving Π' can both be done in polynomial time.

The transformation from Π to Π' is defined as follows:

- $\mathcal{M} = \{1, \ldots, m\}$ where m is the number of partitions of $\mathcal{P}$;
- $\mathcal{S}_i = P_i$ for $1 \leq i \leq m$;
- the function[6] $\xi : \mathbf{2}^\mathcal{P} \to \mathcal{S}^+$ is defined s.t. for $i \in \mathcal{M}$,

$$\xi(S)[i] = \begin{cases} u, & S \cap P_i = \emptyset \\ x, & S \cap P_i = \{x\} \end{cases}$$

  and $\xi^{-1} : \mathcal{S}^+ \to \mathbf{2}^\mathcal{P}$, the inverse of ξ, is defined s.t. for all i

$$\xi^{-1}(s) \cap P_i = \begin{cases} \emptyset, & s[i] = u \\ \{s[i]\}, & s[i] \neq u; \end{cases}$$

- $|\mathcal{C}| = |\mathcal{R}|$ and for each causal rule $\langle \epsilon, \varphi, \alpha, \delta\rangle \in \mathcal{R}$, the corresponding causal SAS-rule $\langle \epsilon, \mathrm{b}(\epsilon), \mathrm{e}(\epsilon), \mathrm{f}(\epsilon)\rangle \in \mathcal{C}$ is defined as:
  - $\mathrm{b}(\epsilon) = \xi(\delta)$,
  - $\mathrm{e}(\epsilon) = \xi(\alpha)$, and
  - $\mathrm{f}(\epsilon) = \xi(\varphi - \delta)$; and
- $s_\mathcal{I} = \xi(\mathcal{I})$ and $s_\mathcal{G} = \xi(\mathcal{G})$.

[6] In order to make the presentation of the SAS formalism as brief as possible, the inconsistent values have been left out. This does not have any implications for expressiveness, and since 2c, 2d, and 2f guarantees that $s_\mathcal{I}$, $s_\mathcal{G}$, and all action type conditions are consistent, there is no need to define ξ for the case where $|s \cap P_i| > 1$.

Except for the restrictions S1–S3, Π' is obviously a SAS planning task, so prove that S1–S3, SU, SP, and SS are satisfied by Π' in order to prove that it is a SAS-PUS planning task.

S1. For each $\epsilon \in \mathcal{E}$ and $i \in \mathcal{M}$, either $\alpha(\epsilon) \cap P_i = \emptyset$ or not. First suppose $\alpha(\epsilon) \cap P_i = \emptyset$, then 2d gives $\delta(\epsilon) \cap P_i = \emptyset$ so $\mathrm{b}(\epsilon)[i] = \xi(\delta)[i] = u$ and $\mathrm{e}(\epsilon)[i] = \xi(\alpha)[i] = u$. Instead, suppose $\alpha(\epsilon) \cap P_i \neq \emptyset$, then 2e gives $\delta(\epsilon) \cap P_i \neq \alpha(\epsilon) \cap P_i$ so $\mathrm{b}(\epsilon)[i] = \xi(\delta)[i] \neq \xi(\alpha)[i] = \mathrm{e}(\epsilon)[i]$. Since $\delta(\epsilon) \cap P_i \neq \emptyset$ only if $\alpha(\epsilon) \cap P_i \neq \emptyset$, S1 follows trivially.

S2. Suppose $\mathrm{b}(\epsilon)[i] = \mathrm{f}(\epsilon)[i] \neq u$ for some $\epsilon \in \mathcal{E}$ and $i \in \mathcal{M}$, then $\delta(\epsilon) \cap P_i = (\varphi(\epsilon) - \delta(\epsilon)) \cap P_i \neq \emptyset$. This is impossible, so S2 is satisfied.

S3. Immediate from 3a.

SU. For all $\epsilon \in \mathcal{E}$, 2b gives $|\delta(\epsilon)| = 1$ so there is exactly one i s.t. $\delta(\epsilon) \cap P_i \neq \emptyset$. Hence, there is exactly one $i \in \mathcal{M}$ s.t. $\mathrm{b}(\epsilon)[i] = \xi(\delta(\epsilon))[i] \neq u$ and, SU follows.

SP. For arbitrary $\epsilon, \epsilon' \in \mathcal{E}$, suppose there is some i s.t. $\mathrm{e}(\epsilon)[i] = \mathrm{e}(\epsilon')[i] \neq u$. Then $\alpha(\epsilon) \cap P_i = \alpha(\epsilon') \cap P_i \neq \emptyset$ so SP follows from the contrapositive of 3b.

SS. For arbitrary $\epsilon, \epsilon' \in \mathcal{E}$, suppose there is some i s.t. $\mathrm{f}(\epsilon)[i] \neq u$ and $\mathrm{f}(\epsilon')[i] \neq u$. Then $(\varphi(\epsilon) - \delta(\epsilon)) \cap P_i \neq \emptyset$ and $(\varphi(\epsilon') - \delta(\epsilon')) \cap P_i \neq \emptyset$ so 3c gives $(\varphi(\epsilon) - \delta(\epsilon)) \cap P_i = (\varphi(\epsilon') - \delta(\epsilon')) \cap P_i$ which implies $\mathrm{f}(\epsilon)[i] = \mathrm{f}(\epsilon')[i]$, and SS follows.

Proving that the nonlinear plans for Π are exactly the nonlinear SAS-plans for Π' means to prove for every tuple $\Delta = \langle \mathcal{A}, \prec \rangle$, s.t. $\mathcal{A}$ is a set of actions of some type in $\mathcal{E}$ and $\prec$ is a partial order on $\mathcal{A}$, that $ACS(\Delta, \mathcal{I}) = CS(\Delta)$ iff $ACS_{\mathrm{SAS}}(\Delta, \mathcal{I}) = CS(\Delta)$, and for all $\mathbf{f} \in CS(\Delta)$, $Result(\mathcal{I}, \mathbf{f}) = \xi^{-1}(Result_{\mathrm{SAS}}(\xi(\mathcal{I}), \mathbf{f}))$. Proof by induction that for every initial sequence $\mathbf{g}$ of $\mathbf{f}$, $\mathbf{g}$ is admissible relative to $\mathcal{I}$ iff $\mathbf{g}$ is SAS-admissible relative to $\xi(\mathcal{I})$, and if $\mathbf{g}$ is admissible relative to $\mathcal{I}$ then $\xi^{-1}(Result_{\mathrm{SAS}}(\xi(\mathcal{I}), \mathbf{g}) = Result(\mathcal{I}, \mathbf{g})$.

Basis: The empty sequence $\langle\rangle$ is both admissible relative to $\mathcal{I}$ and SAS-admissible relative to $\xi(\mathcal{I})$. Furthermore,

$$\xi^{-1}(Result_{\mathrm{SAS}}(\xi(\mathcal{I}), \langle\rangle)) = \xi^{-1}(\xi(\mathcal{I})) = \mathcal{I} = Result(\mathcal{I}, \langle\rangle).$$

Induction: Suppose that for some f in $\mathbf{f}$, $\mathbf{f}\backslash f$ is admissible relative to $\mathcal{I}$ iff $\mathbf{f}\backslash f$ is SAS-admissible relative to $\xi(\mathcal{I})$, and also suppose that if $\mathbf{f}\backslash f$ is admissible relative to $\mathcal{I}$, then $\xi^{-1}(Result_{\mathrm{SAS}}(\xi(\mathcal{I}), \mathbf{f}\backslash f)) = Result(\mathcal{I}, \mathbf{f}\backslash f)$. If $\mathbf{f}\backslash f$ is not admissible relative to $\mathcal{I}$, then $\mathbf{f}/f$ is trivially neither admissible relative to $\mathcal{I}$ nor SAS-admissible relative to $\xi(\mathcal{I})$ and vice versa. Suppose instead that $\mathbf{f}\backslash f$ is admissible relative to $\mathcal{I}$, then

$\mathbf{f}/f$ is admissible relative to $\mathcal{I}$

iff f is admissible relative to $\mathit{Result}(\mathcal{I},\mathbf{f}\backslash f)$

iff $\varphi(f) \subseteq \mathit{Result}(\mathcal{I},\mathbf{f}\backslash f)$

iff $\delta(f) \subseteq \mathit{Result}(\mathcal{I},\mathbf{f}\backslash f)$ and $\varphi(f) - \delta(f) \subseteq \mathit{Result}(\mathcal{I},\mathbf{f}\backslash f)$

iff $\xi(\delta(f)) \sqsubseteq \xi(\mathit{Result}(\mathcal{I},\mathbf{f}\backslash f))$ and $\xi(\varphi(f) - \delta(f)) \sqsubseteq \xi(\mathit{Result}(\mathcal{I},\mathbf{f}\backslash f))$

iff $\mathrm{b}(f) \sqsubseteq \mathit{Result}_{\mathrm{SAS}}(\xi(\mathcal{I}),\mathbf{f}\backslash f)$ and $\mathrm{f}(f) \sqsubseteq \mathit{Result}_{\mathrm{SAS}}(\xi(\mathcal{I}),\mathbf{f}\backslash f)$

iff f is SAS-admissible relative to $\mathit{Result}_{\mathrm{SAS}}(\xi(\mathcal{I}),\mathbf{f}\backslash f)$

iff $\mathbf{f}/f$ is SAS-admissible relative to $\xi(\mathcal{I})$.

It remains to prove that if $\mathbf{f}/f$ is admissible relative to $\mathcal{I}$ then

$$\xi^{-1}(\mathit{Result}_{\mathrm{SAS}}(\xi(\mathcal{I}),\mathbf{f}/f)) = \mathit{Result}(\mathcal{I},\mathbf{f}/f).$$

We will implicitly make use of the fact that, since $\mathbf{f}/f$ is both admissible relative to $\mathcal{I}$ and SAS-admissible relative to $\xi(\mathcal{I})$, we have $|\mathit{app}(\mathit{Result}(\mathcal{I},\mathbf{f}),f)| = |\mathit{app}_{\mathrm{SAS}}(\mathit{Result}_{\mathrm{SAS}}(\xi(\mathcal{I}),\mathbf{f}),f)| = 1$. Let $S = \mathit{Result}(\mathcal{I},\mathbf{f}\backslash f)$ and, hence also, $S = \xi^{-1}(\mathit{Result}_{\mathrm{SAS}}(\xi(\mathcal{I}),\mathbf{f}\backslash f))$ by the induction hypothesis. Let $S' = \xi^{-1}(\mathit{Result}_{\mathrm{SAS}}(\xi(\mathcal{I}),\mathbf{f}/f)) = \xi^{-1}(\mathit{Result}_{\mathrm{SAS}}(\xi(\mathcal{I}),\mathbf{f}\backslash f) \oplus \mathrm{e}(f)) = \xi^{-1}(\xi(S) \oplus \mathrm{e}(f))$. Hence, for all $i \in \mathcal{M}$,

$$\xi(S')[i] = \xi(S) \oplus \mathrm{e}(f) = \begin{cases} \mathrm{e}(f)[i], & \text{if } \mathrm{e}(f)[i] \neq u \\ \xi(S)[i], & \text{otherwise} \end{cases},$$

that is, for all i,

$$S' \cap P_i = \begin{cases} \{\mathrm{e}(f)[i]\}, & \text{if } \mathrm{e}(f)[i] \neq u \\ S \cap P_i, & \text{otherwise} \end{cases} = \begin{cases} \alpha(f) \cap P_i, & \text{if } \alpha(f) \cap P_i \neq \emptyset \\ S \cap P_i, & \text{otherwise} \end{cases}.$$

Also, let $S'' = \mathit{Result}(\mathcal{I},\mathbf{f}/f) = \mathit{Result}(S,f) = S - \delta(f) \cup \alpha(f)$, then 2d and 2f implies that for all i,

$$S'' \cap P_i = \begin{cases} \alpha(f) \cap P_i, & \text{if } \alpha(f) \cap P_i \neq \emptyset \\ S \cap P_i, & \text{otherwise} \end{cases}.$$

It follows that $S' = S''$ so $\xi^{-1}(\mathit{Result}_{\mathrm{SAS}}(\xi(\mathcal{I}),\mathbf{f}/f)) = \mathit{Result}(\mathcal{I},\mathbf{f}/f)$. This ends the induction proof and, hence, the nonlinear SAS-plans for Π' are exactly the nonlinear plans for Π.

Finally, it remains to prove that a plan for Π can be found in polynomial time. The major difficulty with the transformation of Π into Π' is finding the partitioning of $\mathcal{P}$. However, this can be done in polynomial time as follows (details left to the reader). First ascertain that α and δ are singletons for all causal rules. Create $m = |\mathcal{I}|$ singleton sets $P_1, \ldots, P_m$ s.t. each $p \in \mathcal{I}$ belongs to exactly one of the P_is. For each causal rule $\langle \epsilon, \varphi, \alpha, \delta \rangle \in \mathcal{R}$, if $\delta \subseteq P_i$ for some i, then add the proposition in α to P_i. This process must be repeated until no more propositions can be added to any P_i. Any propositions in $\mathcal{P}$ that are not in any of the P_i:s cannot appear in any situation reachable from $\mathcal{I}$. Hence, they can either be discarded or put in arbitrary P_i. Finally, the remaining restrictions must be tested to see if the partitioning is consistent. If not, then Π cannot be transformed into a SAS-PUS planning task. The rest of the transformation is obviously polynomial and plans for Π' can be found in polynomial time using the algorithm in Bäckström and Klein [1991a]. This algorithm is sound and complete so Π can be solved in polynomial time by transforming it into an equivalent SAS-PUS planning task Π', as described above, and applying the algorithm to Π'. ■

6 Planning in a Simple Blocks-World Scenario

The *elementary blocks world* problem (EBW) [Gupta and Nau, 1991] is as follows. There are n distinctly labelled blocks and a table which is large enough to hold at least n blocks. Blocks can be stacked onto each other to any height, but no block is allowed to be immediately supported by more than one block and no block is allowed to immediately support more than one block. There is no metric, so a block cannot be at a specific position on the table; it can only be on the table or on some other block. There are three types of actions that can be performed on the blocks. A block can be moved from a position on some other block onto the table, it can be moved from the table to a position on some other block, and it can be moved from a position on some block to a position on some other block. The obvious restrictions apply. For example, a block cannot be moved if there is some other block on it, and a block cannot be moved onto a block on which there is already some other block. The EBW planning problem is, given an initial configuration and a desired (goal) configuration, find a plan that, if applied in the initial configuration, moves around the blocks so that the desired configuration will hold after executing the plan. The *primitive blocks world* problem (PBW) is the EBW problem with the extra restriction that the goal state is completely specified. Gupta and Nau [1991] have shown that finding a minimal plan for PBW, and thus implicitly also for EBW, is NP-hard.

Let the *restricted* EBW problem (EBW$^-$) denote the same problem but with the restriction that blocks are not allowed to be moved immediately from one block to another block, i.e., they must first be moved to the table and then moved onto the new block. Similarly, define the *restricted* PBW problem (PBW$^-$) in the same way. Bylander [1991, Theorem 10] has shown that the *plan existence* problem for EBW^-, and hence also for PBW^-, can be solved in polynomial time. We will show below that PBW^- can be encoded in the SAS-PUBS problem [Bäckström and Klein, 1991b], which implies that derivation of minimal plans is also a polynomial time problem. Before we show how to encode PBW^- in SAS-PUBS we will present a small example to illustrate the principle.

Suppose we have three blocks `A`, `B` , and `C`. Then there are twelve action types, for example, block `A` can be moved from block `B` onto the table, from block `C` onto the table, from the table onto block `B`, and from the table onto block `C`. Analogously, there are four action types for each of the other blocks. We need six state variables, as follows: `AonB`, `AonC`, `BonA`, `BonC`, `ConA`, and `ConB`. Each of these can have the values *true* (`t`), *false* (`f`), and *undefined* (`u`). The state variable `AonB` is true iff block `A` is immediately on block `B`; i.e., it is not true if block `A` is on block `C` which is on block `B`. The action types for moving block `A` can be encoded as shown in table 6, where states are encoded as tuples ⟨`AonB`, `AonC`, `BonA`, `BonC`, `ConA`, `ConB`⟩.

Action type	Pre-condition	Post-condition	Prevail-condition
AfromB	⟨t, u, u, u, u, u⟩	⟨f, u, u, u, u, u⟩	⟨u, f, f, u, f, f⟩
AfromC	⟨u, t, u, u, u, u⟩	⟨u, f, u, u, u, u⟩	⟨f, u, f, f, f, u⟩
AtoB	⟨f, u, u, u, u, u⟩	⟨t, u, u, u, u, u⟩	⟨u, f, f, u, f, f⟩
AtoC	⟨u, f, u, u, u, u⟩	⟨u, t, u, u, u, u⟩	⟨f, u, f, f, f, u⟩

Table 1: Encoding of action types for moving block `A`.

Action type `AfromB` moves block `A` from block `B` onto the table, i.e., it changes the state variable `AonB` from true to false. The prevail-condition expresses that we must also require that block `A` is not also on some other block[7] (`AonC` false), there is no block on block `A` (`BonA` and `ConA` false), and there is no block on block `B` (`ConB` false). The encodings of the action types `AfromC` (move block `A` from block `C` to the table), `AtoB` (move block `A` from the table to block `B`), and `AtoC` (move block `A` from the table to block `C`) are motivated analogously. Action types for moving blocks `B` and `C` can be encoded in the same way.

The general case with n blocks labelled $\mathtt{B}_1, \ldots, \mathtt{B}_n$ can be encoded as follows. There are $n^2 - n$ state variable indices[8] x_{ij} for all i and j s.t. $1 \leq i \leq n$, $1 \leq j \leq n$, and $i \neq j$, and each state variable can take on the values true (`t`), false (`f`), and undefined (`u`). For each pair of blocks $\mathtt{B}_i$, $\mathtt{B}_j$

[7]Strictly speaking, this is not necessary to test here if we make sure that no block can be on two other blocks simultaneously in the initial state and that no action type can bring about this situation.

[8]Note that the symbols x and y, usually with subscripts, denote *state variable indices*, not state variables.

s.t. $i \neq j$, there are two action types: $\mathtt{B}_i\mathtt{fromB}_j$ and $\mathtt{B}_i\mathtt{toB}_j$. The conditions of $\mathtt{B}_i\mathtt{fromB}_j$ are encoded s.t.

- $\mathrm{b}(\mathtt{B}_i\mathtt{fromB}_j)[x_{ij}] = \mathtt{t}$ and $\mathrm{b}(\mathtt{B}_i\mathtt{fromB}_j)[y] = \mathtt{u}$ for all $y \neq x_{ij}$;
- $\mathrm{e}(\mathtt{B}_i\mathtt{fromB}_j)[x_{ij}] = \mathtt{f}$ and $\mathrm{b}(\mathtt{B}_i\mathtt{fromB}_j)[y] = \mathtt{u}$ for all $y \neq x_{ij}$; and
- $\mathrm{f}(\mathtt{B}_i\mathtt{fromB}_j)[x_{ik}] = \mathtt{f}$ for all k s.t. $i \neq k \neq j$ (block $\mathtt{B}_i$ is not on any other block than $\mathtt{B}_j$), $\mathrm{f}(\mathtt{B}_i\mathtt{fromB}_j)[x_{ki}] = \mathtt{f}$ for all $k \neq i$ (no block is on block $\mathtt{B}_i$), $\mathrm{f}(\mathtt{B}_i\mathtt{fromB}_j)[x_{kj}] = \mathtt{f}$ for all k s.t. $i \neq k \neq j$ (no block is on block $\mathtt{B}_j$, except block $\mathtt{B}_i$), and $\mathrm{f}(\mathtt{B}_i\mathtt{fromB}_j)[y] = \mathtt{u}$ for all other state variable indices y.

The conditions of $\mathtt{B}_i\mathtt{toB}_j$ are identical except that $\mathrm{b}(\mathtt{B}_i\mathtt{fromB}_j)[x_{ij}] = \mathtt{f}$ and $\mathrm{e}(\mathtt{B}_i\mathtt{fromB}_j)[x_{ij}] = \mathtt{t}$. The initial state $s_{\mathcal{I}}$ must also satisfy that for each i, there is at most one j s.t. $s_{\mathcal{I}}[x_{ij}] = \mathtt{t}$, and similarly for the goal state $s_{\mathcal{G}}$. This encoding of EBW$^-$ is essentially the same as in Bylander [1991], and it obviously satisfies the restrictions for the SAS-PUBS problem[9]. As a consequence, optimal planning for PBW$^-$ is a polynomial time problem.

Proposition 6 *The encoding of* PBW$^-$ *shown above satisfies the restrictions for the* SAS-PUBS *problem.*

Proposition 6 together with the results in Bäckström and Klein [1991b] immediately lead to the following corollary.

Corollary 7 *Minimal plans for the* PBW$^-$ *problem can be found in polynomial time.*

Furthermore, the planning algorithm in Bäckström and Klein [1991b] can easily be modified to handle incompletely specified goal states, so corollary 7 also holds for EBW$^-$. Although this result strengthens Bylander's [1991] Theorem 10, it does not extend to the general EBW problem, as has been shown by Gupta and Nau [1991]. On the other hand, any plan for an EBW$^-$ planning task is also a plan for the corresponding EBW planning task, and a minimal plan for an EBW$^-$ planning task is at most twice as long as a minimal plan for the corresponding EBW planning task. It is also likely that one could derive near-minimal plans for an EBW planning task in polynomial time from a minimal plan for the corresponding EBW$^-$ planning task.

It is an interesting observation that the PBW$^-$ problem can be encoded to satisfy the restrictions both for the SAS-PUBS problem and for one of the problems which Bylander has proven tractable. However, neither of these problems appear to properly include the other. The exact relationship between these two problems remains to be investigated, however.

[9]The SAS-PUBS problem has the same restrictions as the SAS-PUS problem plus the restriction that each state variable domain has only two defined values, i.e., $|\mathcal{S}_i| = 2$.

7 Temporal Projection and Plan Validation

Dean and Boddy [1988, p. 378] suggest that temporal projection is the basic underlying problem in plan validation:

> A nonlinear plan is represented as a set of actions $\{e_1, \ldots, e_n\}$ partially ordered by $\prec$. Each action has some set of *intended effects:* Intended$(e_i) \subseteq \mathcal{P}$. A nonlinear plan is said to be *valid* just in case Intended$(e_i) \subseteq$ Necessary(e_i), for $1 \leq i \leq n$.

Although this definition sounds reasonable, there are some points which are arguable. As we have seen in Definition 7, a plan is a solution iff (1) it achieves its goal, and (2) it is coherent relative to the initial state, i.e., all preconditions are necessarily satisfied.[10] If a plan achieves its overall goals (ignoring its coherence), it is called **partially valid**. If it is partially valid and coherent relative to the initial state, it is called **valid**. Note that in contrast to Dean and Boddy's formulation, we do not refer to the *intended effects of particular events* but to the effects of the *overall plan* and to the state *before* particular events.

Deciding whether a plan is partially valid can be straightforwardly reduced to temporal projection in linear time. Given a planning task $\Pi = \langle \Phi, \mathcal{I}, \mathcal{G} \rangle$, and a plan Δ_Φ, we extend the plan by an event e_* that is not associated with any causal rule and occurs after all other events. The resulting plan is called Δ'_Φ. Now it is easy to see that Δ_Φ is partially valid if, and only if, $\mathcal{G} \subseteq \mathit{Nec}(e_*, \langle \Delta'_\Phi, \mathcal{I} \rangle)$.

Coherence, however, is a property that cannot be easily reduced to temporal projection as defined by Dean and Boddy. If we restrict ourselves to *unconditional* causal structures, however, we can define a variant of the temporal projection problem that refers to the state *before* an event occurs and that can be used to decide coherence. More importantly, the restriction to unconditional causal structures will enable us to prove tractability of plan validation. Although the restriction may sound severe, it shows that plan validation is tractable for a considerable larger class of problems than temporal projection. Furthermore, we will, at the end of this section, argue that this restriction is not very severe at all.

Definition 12 *Given an event system Θ, an event $e \in \mathcal{A}$, and a condition $p \in \mathcal{P}$:*

$$p \in \mathit{Poss}_b(e, \Theta) \quad \mathit{iff} \quad \exists \mathbf{f} \in \mathit{CS}(\Theta)\colon p \in \mathit{Result}(\mathcal{I}, \mathbf{f}\backslash e)$$
$$p \in \mathit{Nec}_b(e, \Theta) \quad \mathit{iff} \quad \forall \mathbf{f} \in \mathit{CS}(\Theta)\colon p \in \mathit{Result}(\mathcal{I}, \mathbf{f}\backslash e).$$

[10] Note that our definition coincides with Chapman's [1987, p. 340] definition of when a plan *solves* a problem.

Proposition 8 *An unconditional event system* Θ *is coherent iff*

$$\forall e \in \mathcal{A}\colon \varphi(e) \subseteq Nec_b(e, \Theta).$$

Deciding $p \in Nec_b(e, \Theta)$ instead of $p \in Nec(e, \Theta)$ does not simplify the problem. All the NP-hardness proofs for *Nec* can be easily used to show NP-hardness for Nec_b. For instance, the following corollary is an immediate consequence of Corollary 2.

Corollary 9 *Deciding* $p \in Nec_b(e, \Theta)$ *is co-NP-complete for simple event systems.*

In order to simplify the following discussion, we will restrict ourselves to **consistent** unconditional event systems, which have to meet the restrictions that $\alpha(e) \cap \delta(e) = \emptyset$, for all $e \in \mathcal{A}$. Note that any unconditional event system Θ can be transformed into a consistent unconditional event system Θ' in polynomial time by setting

$$\begin{aligned} \varphi'(e) &= \varphi(e) \\ \alpha'(e) &= \alpha(e) \\ \delta'(e) &= \delta(e) - \alpha(e), \end{aligned}$$

for all $e \in \mathcal{A}$. Consulting the definition of *Result*, it is obvious that this modification does not change the outcome of $Result(S, \mathbf{f})$ for all $S \subseteq \mathcal{P}$ and all partial event sequences $\mathbf{f}$ over Θ.

As a first step to specifying a polynomial algorithm that decides coherence for unconditional event systems, we define a simple syntactic criterion, written $Maybe_b(e, \Theta)$, that approximates $Nec_b(e, \Theta)$.

Definition 13 *Given a consistent, unconditional event system* Θ*, an atom* $p \in \mathcal{P}$*, and an event* $e \in \mathcal{A}$*,* $Maybe_b(e, \Theta)$ *is defined as follows:*

$$\begin{aligned} p \in Maybe_b(e, \Theta) \text{ iff } & (1)\; p \in \mathcal{I} \vee \exists e' \in \mathcal{A}\colon (e' \prec e \wedge p \in \alpha(e')) \wedge \\ & (2)\; \neg\exists e' \in \mathcal{A} - \{e\}\colon (e' \not\prec e \wedge e \not\prec e' \wedge p \in \delta(e')) \wedge \\ & (3)\; \forall e' \in \mathcal{A}\colon \big((e' \prec e \wedge p \in \delta(e')) \rightarrow \\ & \qquad \exists e'' \in \mathcal{A}\colon (e' \prec e'' \prec e \wedge p \in \alpha(e''))\big). \end{aligned}$$

This definition resembles Chapman's [1987] *modal truth criterion.* The first condition states that p has to be established before e. The second condition makes sure that there is no event unordered w.r.t. e that could delete p, and the third condition enforces that for all events that could delete p and that occur before e, some other event will reestablish p. It is obvious that this criterion can be checked in polynomial time.

Proposition 10 $Maybe_b(e, \Theta)$ *can be decided in polynomial time.*

Note that $Maybe_b$ is neither sound nor complete w.r.t. Nec_b in the general case because we do not know whether the events referred to in the definition actually affect the world. However, $Maybe_b$ coincides with Nec_b in the important special case that the event system is consistent and coherent.

Lemma 11 *Let Θ be an consistent unconditional event system. If Θ is coherent, then*

$$\forall e \in \mathcal{A}\colon Nec_b(e,\Theta) = Maybe_b(e,\Theta).$$

Proof. "$\subseteq$": We will show that all three conditions of $p \in Maybe_b(e,\Theta)$ in Definition 13 are true for all $e \in \mathcal{A}$ and all $p \in Nec_b(e,\Theta)$.

Assume that the first condition does not hold for some event e and atom $p \in Nec_b(e,\Theta)$, i.e., $p \notin \mathcal{I}$ and $\neg\exists e'\colon e' \prec e \wedge p \in \alpha(e')$. Since Θ is coherent, we can construct an admissible complete event sequence $\mathbf{f} = \langle f_1,\ldots,e,\ldots\rangle$ such that $\mathbf{g} = \mathbf{f}\backslash e$ contains only events g_i such that $g_i \prec e$. By induction over the length of the length of $\mathbf{f}\backslash e$, we get $p \notin Result(\mathcal{I},\mathbf{f}\backslash e)$, hence $p \notin Nec_b(e,\Theta)$, which is a contradiction.

Assume that the second condition does not hold for some event e and atom $p \in Nec_b(e,\Theta)$, i.e., there exists an event e' unordered with respect to e such that $p \in \delta(e')$. Since e' is unordered with respect to e, there exists a complete event sequence $\mathbf{f} = \langle f_1,\ldots,e',e,\ldots\rangle$. Since Θ is coherent, and thus e' affects the world, it is obvious that $p \notin Result(\mathcal{I},\mathbf{f}/e') = Result(\mathcal{I},\mathbf{f}\backslash e) \supseteq Nec_b(e,\Theta)$, which is a contradiction.

Assume the third condition is not satisfied, i.e., there exists $p \in Nec_b(e,\Theta)$ and an event $e' \prec e$ such that $p \in \delta(e')$, but there is no e'' such that $e' \prec e'' \prec e$ and $p \in \alpha(e'')$. Consider a complete event sequence $\mathbf{f} = \langle f_1,\ldots,e',\ldots,e,\ldots\rangle$ such that there are only events f_i between e' and e that have to occur between them. Because $p \notin Result(\mathcal{I},\mathbf{f}/e')$ and there are no events after e' that have p in the add list, using induction on the length of $\mathbf{f}\backslash e$, we can infer $p \notin Result(\mathcal{I},\mathbf{f}\backslash e) \supseteq Nec_b(e,\Theta)$, which is again a contradiction.

"$\supseteq$": Assume $p \in Maybe_b(e,\Theta)$. We will show that also $p \in Nec_b(e,\Theta)$. Consider any complete event sequence $\mathbf{g} \in CS(\Theta)$. We want to show that $p \in Result(\mathcal{I},\mathbf{g}\backslash e)$. By condition (1) of the definition of $Maybe_b$ and the fact that all complete event sequences are admissible, we know that there exists $g_i \in \mathcal{A}$ such that $|\mathbf{g}\backslash g_i| \leq |\mathbf{g}\backslash e|$ and $p \in Result(\mathcal{I},\mathbf{g}\backslash g_i)$. Consider the latest such event, i.e., g_i with a maximal i. Since all event sequences are finite, such an event must exist. If $g_i = e$, we are ready. Otherwise, we will show that i cannot be maximal.

Since g_i is the latest event in $\mathbf{g}$ such that $p \in Result(\mathcal{I},(\mathbf{g}\backslash e)\backslash g_i)$, it must be the case that $p \in \delta(g_i)$. By condition (2) in the definition of $Maybe_b$, we know that g_i cannot be unordered with respect to e. By condition (3), we know that there exists an event g_j such that $g_i \prec g_j \prec e$ and $p \in \alpha(g_j)$.

Because $\varphi(g_j) \subseteq Nec_b(g_j, \Theta)$ it must be the case that $p \in Result(\mathcal{I}, \mathbf{g}/g_j)$ and $|\mathbf{g}\backslash g_i| < |\mathbf{g}/g_j| \leq |\mathbf{g}\backslash e|$. Hence, g_i cannot be the latest event before e such that p holds before the occurrence of g_i. Hence, $p \in Result(\mathcal{I}, \mathbf{g}\backslash e)$. Because $\mathbf{g}$ was an arbitrary element of $CS(\Theta)$, this holds for all complete event sequences. Hence, $p \in Nec_b(e, \Theta)$. ∎

Now we can give a necessary and sufficient condition for coherence of consistent unconditional event systems.

Theorem 12 *A consistent unconditional event system Θ is coherent iff*

$$\forall e \in \mathcal{A}\colon \varphi(e) \subseteq Maybe_b(e, \Theta).$$

Proof. "$\Rightarrow$": Since Θ is coherent, we know that $\forall e \in \mathcal{A}\colon \varphi(e) \subseteq Nec_b(e, \Theta)$. Further, by Lemma 11, $Maybe_b(e, \Theta) = Nec_b(e, \Theta)$, for all $e \in \mathcal{A}$. Hence, $\forall e \in \mathcal{A}\colon \varphi(e) \subseteq Maybe_b(e, \Theta)$.

"$\Leftarrow$": For the converse direction, we use induction on the number of conditions appearing in the preconditions of events over the entire event system: $\sum_{e \in \mathcal{A}} |\varphi(e)|$. As the base step, we assume, that for all events $e \in \mathcal{A}$, $\varphi(e) = \emptyset$. Clearly, $\varphi(e) \subseteq Maybe_b(e, \Theta)$ and $\varphi(e) \subseteq Nec_b(e, \Theta)$, for all $e \in \mathcal{A}$. Hence, the hypothesis holds for $k = 0$.

Now assume that our claim holds for all event systems with k or less preconditions. We will show that it also holds for event systems with $k+1$ preconditions.

Consider an event system Θ with $k+1$ preconditions such that $\varphi(e) \subseteq Maybe_b(e, \Theta)$ for all $e \in \mathcal{A}$. Choose one event f that has a nonempty set of preconditions and replace the associated causal rule $\langle type(f), \varphi, \alpha, \delta\rangle$ by the rule $\langle type(f), \emptyset, \alpha, \delta\rangle$. This new event system is called Θ'. We will write $\varphi'(e)$, $\alpha'(e)$, and $\delta'(e)$ in order to refer to the preconditions, add lists, and delete lists in Θ', respectively. Note that for all $e \in \mathcal{A} - \{f\}$ it still holds that $\varphi'(e) \subseteq Maybe_b(e, \Theta') = Maybe_b(e, \Theta)$ because the $Maybe_b$ conditions do not refer to φ. Further, we have vacuously that $\varphi'(f) \subseteq Maybe_b(f, \Theta')$. Because $k \geq \sum_{e \in \mathcal{A}'} |\varphi'(e)|$, we can apply our induction hypothesis and know that $\varphi'(e) \subseteq Nec_b(e, \Theta')$ for all $e \in \mathcal{A}$, hence Θ' is coherent. Finally note that by Lemma 11, we still have $\varphi(f) \subseteq Maybe_b(f, \Theta) = Maybe_b(f, \Theta') = Nec_b(f, \Theta')$. Hence, any sequence $\mathbf{g} \in CS(\Theta')$ that contains f is an admissible sequence even if $\varphi'(f) = \varphi(f)$. Since we have $CS(\Theta) = CS(\Theta')$, it follows that all sequences $\mathbf{h} \in CS(\Theta)$ are admissible. Hence, Θ is coherent, whence, the induction hypothesis holds for $k+1$ preconditions. ∎

From that it follows straightforwardly that coherence can be decided in polynomial time.

Corollary 13 *Coherence of unconditional event systems can be decided in polynomial time.*

Proof. The claim follows immediately from Theorem 12, the fact that $p \in \mathit{Maybe}_b(e, \Theta)$ can be decided in polynomial time, and the fact that any unconditional event system can be transformed into a consistent one in polynomial time. ■

Plan validation can easily be reduced to coherence, so it is a polynomial time problem if the causal structure is unconditional.

Theorem 14 *Deciding whether a plan Δ_Φ is a solution for a planning task Π with an unconditional causal structure is a polynomial time problem.*

Proof. Follows immediately from Corollary 13 and the fact that plan validation can be reduced to coherence in linear time as follows: Add an extra event e_* s.t. $\varphi(e_*)$ is the intended effects of the plan and e_* is constrained to occur after all other events. ■

One interesting point to note about this result is that it appears to be easier to decide a big conjunction of the form

$$\bigwedge_{e \in \mathcal{A}} \varphi(e) \subseteq \mathit{Nec}_b(e, \Theta)$$

than to decide one of the conjuncts. In other words, the claim by Dean and Boddy [1988] that temporal projection (in some form) is the underlying problem of plan validation is conceptually correct. However, it turns out that solving the subproblems is (most probably) harder than solving the original problem.

Although maybe surprising, the result is not new. Chapman [1987] used a similar technique to prove plan validation to be a polynomial time problem for a slightly different formalism. It should be noted, however, that Chapman's [1987, p. 368] proof of the correctness and soundness of the *modal truth criterion* is correct only if we make the assumption that the plan is already coherent. Alternatively, we could modify the meaning of the term *necessary* as used by Chapman to a notion that is weaker than Nec_b. It seems to be the case that Chapman means by "a proposition is necessarily asserted in a situation" that the postcondition contains a certain proposition (in our simple formalism). However, because we do not know whether the event affects the world, i.e., asserts the particular proposition, we cannot make any claim whether the particular proposition really will get asserted. So it seems to be the case that Chapman actually means Maybe_b instead of Nec_b and misses to prove the second half of our Theorem 12.

We will end this section with a brief analysis of the implications of restricting event systems to be unconditional. There are mainly three motivations for conditional actions: to handle uncertain initial states, context-dependent outcome of actions, and external events, i.e., events out of the control of the planner. An example for the first case are the following two rules associated with the event type ϵ_{call}:

$$\langle \epsilon_{call}, \ \{\mathtt{a},\mathtt{p}\}, \ \{\mathtt{i}\}, \ \emptyset \rangle,$$
$$\langle \epsilon_{call}, \ \{\mathtt{a},\mathtt{c}\}, \ \{\mathtt{i}\}, \ \emptyset \rangle.$$

Regardless of whether the robot has coins (c) or a phone card (p), he can make his call and afterwards the conditions are the same. It seems possible that under reasonable restrictions such cases could actually be handled by a slight extension of the plan validation algorithm. However, a further analysis of such cases is necessary. An example for context-dependent actions is provided by the causal rules describing the effects of the ϵ_{charge} action. After this action Robby's batteries are fully charged or damaged, depending on the state of the batteries before the event. Chapman [1987] has already shown that plan validation becomes NP-hard in this case. However, it seems more reasonable to handle this kind of combinatorics in the planner. The planner may *commit* itself in advance to one of the causal rules associated with the action and make sure that only this rule gets applied. In other words, the task of plan validation is then to check that only the committed rules are actually applied, which again can be reduced to the plan validation problem as defined above.

Coping with external events usually means to undo the effects of some event e whose occurrence is out of our control. This can be done by executing an action e' after e such that e' undoes the effects of e. This can be done only if we know when e will occur or if we can plan to wait for its occurrence. Furthermore, e' need not be a conditional action but can rather be an action that has the inverse effect of e even if e has not occurred. The only case where conditional actions are really needed in order to cope with external events is when there can occur any number of external events and we do not know when they will occur and possibly not even what events may occur. In this case, we need more advanced types of conditional plans (see, for example, Schoppers [1987]), which cannot be modelled in STRIPS-like formalisms. It seems that the formalism suggested by Dean and Boddy is too weak to adequately express those scenarios where conditional actions are needed.

Summarizing, for plan validation purposes in the STRIPS-like formalism as used in this paper, it hardly seems to be a severe restriction to require the event systems to be unconditional.

8 Approximate Temporal Projection

Based on the observation that temporal projection is difficult even for severly restricted cases, Dean and Boddy [1988] develop an incomplete decision procedure that computes its results in polynomial time. Reconsidering the reflections from the previous sections, one may ask whether this procedure is based on the right assumptions and whether it gives useful results. Such a judgement is, of course, difficult.

In the area of reasoning about temporal relations between events [Allen, 1983], it was possible to identify tractable special cases that are natural for uncertain observations and text understanding [Nökel, 1989; Vilain *et al.*, 1989]. Further, the incomplete decision procedure for the full problem turned out to be complete for the tractable special case. Thus, we have a good justification for using the incomplete algorithm in this case.

If we consider the incomplete decision procedure for temporal projection, there is the question what the interesting special cases are where we want the procedure to be complete. Dean and Boddy [1988, Theorem 3.4] prove their procedure to be complete if the events are totally ordered, which gives us one characterization of the behavior of the procedure. Since one of the intended applications is validation of nonlinear plans, one would also expect that the procedure is complete for cases where plan validation is tractable, e.g., if we consider unconditional events only. This is not the case, however. The main reason for this failure is that the procedure considers all events *unordered* with respect to a given event as equally likely to appear. Condition (3) in the definition of $Maybe_b$, however, tells us that sometimes the deletion of an atom can be ignored.

Since we cannot reproduce the entire procedure because of space limitations, the reader is referred to the original article [Dean and Boddy, 1988, p. 380-392]. Here we will only sketch the ideas of the procedure. For every event e, two sets are computed, namely, $Strong(e, \Theta)$ and $Weak(e, \Theta)$, such that

$$Strong(e, \Theta) \subseteq Nec(e, \Theta) \subseteq Poss(e, \Theta) \subseteq Weak(e, \Theta),$$

where $Strong(e, \Theta)$ is intended to contain only conditions that hold after e in *all* complete event sequences, while $Weak(e, \Theta)$ is meant to contain all conditions that might hold after e in *some* complete event sequence.

In addition, the sets $S\text{-}Strong(e, \Theta)$ and $S\text{-}Weak(e, \Theta)$ are computed. The first set contains all of $Strong(e, \Theta)$ except those conditions that could be deleted by an event unordered with respect to e. Similarly, $S\text{-}Weak(e, \Theta)$ contains all of $Weak(e, \Theta)$ plus those conditions that could be added by events unordered with respect to e.

Consider now the following unconditional event system:

$$\begin{aligned}
\mathcal{P} &= \{\mathtt{p}, \mathtt{q}, \mathtt{r}\} \\
\mathcal{E} &= \{\epsilon_a, \epsilon_b, \epsilon_c\} \\
\mathcal{R} &= \{\langle \epsilon_a, \{\mathtt{q}\}, \{\}, \{\mathtt{r}\} \rangle, \\
&\quad\ \langle \epsilon_b, \{\mathtt{q}\}, \{\mathtt{r}\}, \{\} \rangle, \\
&\quad\ \langle \epsilon_c, \{\mathtt{q}, \mathtt{r}\}, \{\mathtt{p}\}, \{\} \rangle \}
\end{aligned}$$

$$\mathcal{A} = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}\}$$
$$\mathcal{I} = \{\mathtt{q}\}$$

The types of the events and the partial order is given in Figure 1. It is easy

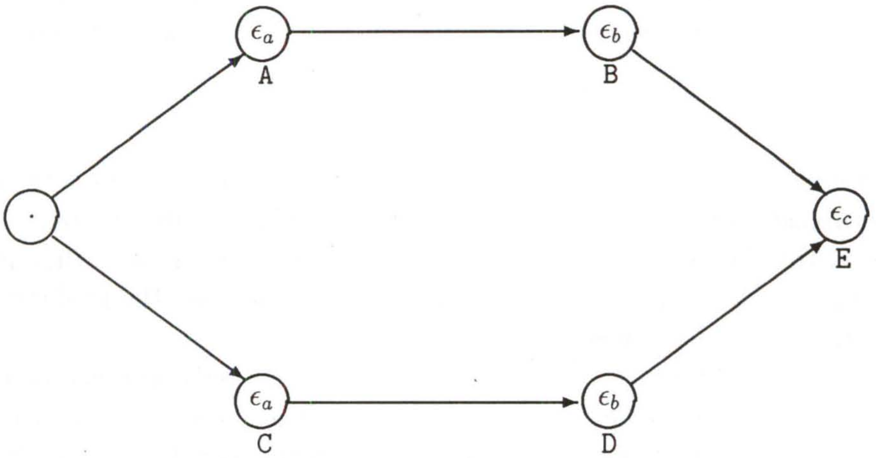

Figure 1: A valid nonlinear plan

to see that this unconditional event system is coherent and achieves $\{\mathtt{p}, \mathtt{q}, \mathtt{r}\}$. Using Theorem 14, this could be checked in polynomial time. However, the incomplete decision procedure is too conservative. It misses to report that r and p are among the necessary consequences, as can be seen from Table 2.

Event	Type	*S-Strong*	*Strong*	*Nec*	*Poss*	*Weak*	*S-Weak*
		{q}	{q}	{q}	{q}	{q}	{q}
A	ϵ_a	{q}	{q}	{q}	{q}	{q}	{q, r}
B	ϵ_b	{q}	{q, r}	{q, r}	{q, r}	{q, r}	{q, r}
C	ϵ_a	{q}	{q}	{q}	{q}	{q}	{q, r}
D	ϵ_b	{q}	{q, r}	{q, r}	{q, r}	{q, r}	{q, r}
E	ϵ_c	{q}	{q}	{p, q, r}	{p, q, r}	{p, q, r}	{p, q, r}

Table 2: Results of the incomplete decision procedure

In the computation of *S-Strong*(B) and *S-Strong*(D), the procedure is overly pessimistic with respect to the occurrence of the events A and C. Since these could delete the condition r, it may be the case that r does not hold before the occurrence of the event E. However, it is easy to see that r is necessarily added before occurrence of E.

In summary, this result shows that in an important tractable special case the incomplete decision procedure fails to provide a complete result.

9 Story Understanding

Besides plan validation, Dean and Boddy [1988, p. 375] also mention story understanding as one domain where temporal projection is important:

> "...an author may not provide the reader with the exact time of all events mentioned in a narrative, knowing that it is not critical that the reader have such information in order to follow the story."

Theorem 1, however, tells us that we are lost, as authors or readers. Even in the simplest case, the author has better to provide complete information or there is the danger that the reader gets lost in figuring out what is the case.[11] However, if we place some reasonable restrictions on the problem, the computational problems vanish.

First of all, it seems reasonable that we consider only admissible event sequences. It simply makes no sense that an author tells a reader that an event takes place that does not have any effect on the world. Conversely, one could argue that an author does not tell the exact time of events if the reader is able to recover the sequential information by other means, for instance, by the *coherence* of the events. If we take, for instance, the event system introduced in Section 2 and assume that the partial ordering over the events is all the author told us about temporal relations, then the natural way to interpret the story is to assume that either $\langle \mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F} \rangle$ or $\langle \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{A}, \mathtt{B}, \mathtt{C} \rangle$ is the course of events because all other possible complete sequences are not admissible. With this assumption, we are able to infer that under the given initial conditions $\{\mathtt{h}, \mathtt{e}, \mathtt{q}\}$ afterwards Robby has informed his master ($\mathtt{i}$), recharged his batteries ($\mathtt{f}$) and returned to the hall ($\mathtt{h}$). Secondly, we will assume that a story is *non-repeating*, i.e., all states are different. Otherwise, the story would contain more than once the same situation—which is rather unlikely. In order to capture this formally, we introduce the notion of **non-repeating sequences** of an event system, written $NCS(\langle \Delta, \mathcal{I} \rangle)$, with the intention that for all events g, h, where $g \neq h$, appearing in an event sequence, we have $Result(\mathcal{I}, \mathbf{f}/g) \neq Result(\mathcal{I}, \mathbf{f}/h)$. Evidently, it is the case that $NCS(\Theta) \subseteq ACS(\Theta)$ because the occurrence of an event e that does not affect the world leads to the same state as before the occurrence of e. Using this formalization of story-understanding, yet another variant of temporal projection is defined.

[11] Note that NP-completeness means that we (most probably) cannot hope to solve the problem effortlessly. Instead, "puzzle mode" reasoning is necessary to arrive at a conclusion [Levesque, 1988].

Definition 14 *Given an event system Θ, an event $e \in \mathcal{A}$, and a condition $p \in \mathcal{P}$:*

$$p \in Poss^{+}(e,\Theta) \quad \textit{iff} \quad \exists \mathbf{f} \in NCS(\Theta)\colon p \in Result(\mathcal{I},\mathbf{f}/e)$$
$$p \in Nec^{+}(e,\Theta) \quad \textit{iff} \quad \forall \mathbf{f} \in NCS(\Theta)\colon p \in Result(\mathcal{I},\mathbf{f}/e).$$

Proposition 15 *For simple event systems Θ, $p \in Nec^{+}(e,\mathcal{I})$ and $p \in Poss^{+}(e,\Theta)$ can be decided in polynomial time.*

Proof. The restriction to non-repeating sequences over simple event systems implies that the effects of all events are unique, i.e., there are no two events with the same add list and the initial state is different from all add lists. The uniqueness of the add lists implies the uniqueness of the preconditions. If the preconditions are not unique, there is no non-repeating event sequence. Thus, we can construct the (unique) event sequence incrementally—provided there exists one—starting with the set of initial conditions. This can be done in polynomial time. $p \in Poss^{+}(e,\Theta)$ iff there exists a non-repeating complete event sequence and the add list of e contains p. $p \in Nec^{+}(e,\Theta)$ iff there exists a non-repeating complete event sequence and $p \in \alpha(e)$ or there is no such sequence. ∎

Thus story understanding (in the highly abstract form as defined here) is easier than temporal projection in the case of simple event systems. The question is, in how far this result can be generalized.

If we remove the restriction that the event sequence is non-repeating and require only that the course of events is admissible, the complexity of story understanding for simple event systems is not obvious. The resulting problem is equivalent to finding an Euler tour in a graph such that the arcs on this tour respect a given partial ordering. It is not obvious whether this problem can be solved in polynomial time. However, as we remarked above, the *non-repeating* restriction seems to be quite reasonable.

Generalizing the problem to general conditional event systems leads immediately to NP-completeness because we can design the causal rules in a way such that all sequences are non-repeating. A more interesting question is, whether we can solve the problem for general unconditional event systems. Because plan-validation is easy in this case, one may suspect that this also holds for temporal projection in an story understanding context. Unfortunately, this is not true, though.[12]

Theorem 16 *For unconditional event systems Θ, deciding $p \in Poss^{+}(e,\Theta)$ is NP-complete.*

[12] Note that instead of *requiring* that all complete event sequences are admissible, here we *quantify* over the non-repeating complete sequences, which is a subset of the admissible sequences.

Proof. Again, membership in NP is obvious. For the hardness part we use the problem of *directed Hamilton path*, which is NP-complete [Garey and Johnson, 1979, p. 199].

Let Let $G = (V, A)$ be a digraph, where $V = \{v_1, \ldots, v_n\}$, and let $v_s, v_t \in V$. Let $\overline{V} = \{\overline{v}_1, \ldots, \overline{v}_n\}$ be a disjoint copy of V. Define the event system Θ as follows:

$$
\begin{aligned}
\mathcal{E} &= \{\epsilon_i | \, v_i \in V\} \\
\mathcal{P} &= V \cup \overline{V} \cup \{p\} \\
\mathcal{R} &= \{\langle \epsilon_i, \{v_i\}, \{\overline{v}_i, v_{j_1}, \ldots, v_{j_m}\}, V\rangle | \, (v_i, v_{j_k}) \in A, v_i \neq v_t\} \cup \\
&\quad\ \{\langle \epsilon_t, \{v_t\}, \{p\}, V\rangle\} \\
\mathcal{A} &= \{e_i | \epsilon_i \in \mathcal{E}\} \\
type(e_i) &= \epsilon_i \text{ for all } e_i \in \mathcal{A},\ 1 \leq i \leq n \\
e_s \prec e & \quad\ \ \text{for all } e \in \mathcal{A} \text{ such that } e \neq e_s \\
\mathcal{I} &= \{v_s\}.
\end{aligned}
$$

Note that Θ is an unconditional event system and that it can be constructed in polynomial time.

Now we claim that there exists a Hamilton path from v_s to v_t in G iff $p \in Poss^{+}(e_t, \Theta)$.

"$\Rightarrow$": Let $w_1, w_2, \ldots, w_n$ be a Hamilton path in G with $w_1 = v_s$ and $w_n = v_t$. By construction of Θ, there exists a non-repeating complete event sequence, $\mathbf{f} = f_1, \ldots, f_n$ such that $f_1 = e_s$ and $f_n = e_t$. Since the add list of e_t is $\{p\}$, we have $p \in Poss^{+}(e_t, \Theta)$.

"$\Leftarrow$": Assume that there exists a non-repeating complete event sequence $\mathbf{f} = f_1, \ldots, f_n$, where $f_n = e_t$. Then there exists by construction of Θ a path from v_s to v_t that contains every vertex exactly once, i.e., there is a Hamilton path from v_s to v_t. ■

Assuming that story understanding is an easy (i.e., tractable) task, this result implies that the formalization of the problem is still too general to account for the structure of the domain. It is desirable to identify restrictions that lead to polynomial algorithms for temporal projections in this domain, but there do not seem to be natural and obvious such conditions.

However, it should be noted that story understanding most probably involves more than can be expressed in our formalism. It seems plausible that plan recognition is one crucial part in story understanding and that abduction in general plays a vital role in such a task. Since we cannot express any of these phenomena, it seems to make not much sense to speculate about the complexity of this task. What seems to be clear, however, is that story understanding is more than temporal projection and that most probably other mechanisms than temporal projection are responsible for inferring the outcome of a story.

10 Conclusions

Reconsidering the problem of temporal projection for sets of partially ordered events as defined by Dean and Boddy [1988], we noted that one special case conjectured to be tractable turned out to be NP-complete. Although this result does not undermine the arguments of Dean and Boddy [1988], it leads to some counter-intuitive results.

Planning is easier than temporal projection in this special case. This positive result can be generalized to a less restricted problem, namely, the so-called SAS-PUS planning problem.

Further, we noticed that plan validation, if defined appropriately, is tractable for an even larger problem, namely validation of unconditional nonlinear plans. This means that the problem of validating a plan as a whole is easier than validating all its actions separately. In other words, what might look like a *divide and conquer* strategy at a first glance is rather the opposite.

These two observations lead to the question of whether the formalization [Dean and Boddy, 1988] really captures one of the intended applications, namely, validation of nonlinear plans. In particular, one may ask whether the incomplete decision procedure for temporal projection developed by Dean and Boddy [1988] is based on the right assumptions.

Dean and Boddy [1988] showed that their incomplete decision procedure is complete if the order of events is total. However, under the assumption that plan validation is one of the intended applications, one would expect that the procedure is also complete for other important tractable special cases, such as validation of nonlinear plans containing only unconditional events, where the plan-validation problem is tractable. It turns out, however, that this is not the case.

Also the task of story understanding, which was another motivation for the formalization of the temporal projection problem and the development of an incomplete decision procedure, does not seem to be particularly well described as a temporal projection problem. Under some reasonable further restrictions, this task is also tractable in the special case where temporal projection is NP-hard. Unfortunately, however, this positive complexity result cannot be straightforwardly generalized. For general unconditional events, the problem can be shown to be NP-complete, while the corresponding plan-validation problem is still tractable. However, it seems to be the case that there is more to story understanding than just temporal projection. Plan recognition seems to play a crucial role which cannot be accounted for in the framework of temporal projection used in this paper.

As a final remark, it should be noted that the criticisms expressed in this paper are possible only because Dean and Boddy [1988] made their ideas and claims very explicit and formal. Although the general direction of isolating and formalizing a problem, followed by the development of an incomplete decision procedure, as exercised by Dean and Boddy, seems a promising way to go, two points should be emphasized. First, sometimes the decomposition of a problem into subproblems can lead to problems that are more difficult than the original problem, as demonstrated by the decomposition of the plan validation problem into temporal projection problems. Second, it is hard to judge the merit of an incomplete decision procedure if there are no well justified criteria for doing this, and such criteria should be given for each proposed incomplete procedure.

Acknowledgements

We would like to thank Bart Selman, who provided helpful comments on an earlier version of this paper.

References

[Allen, 1983] James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November 1983.

[Bäckström and Klein, 1991a] Christer Bäckström and Inger Klein. Parallel non-binary planning in polynomial time. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 268–273, Sydney, Australia, 1991. An extended version of this paper is available as Research Report LiTH-IDA-R-91-11, Department of Computer and Information Science, Linköping University, Linköping, Sweden.

[Bäckström and Klein, 1991b] Christer Bäckström and Inger Klein. Planning in polynomial time: The SAS-PUBS class. *Computational Intelligence*, 7(4), November 1991. To appear.

[Bylander, 1991] Tom Bylander. Complexity results for planning. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 274–279, Sydney, Australia, 1991.

[Chapman, 1987] David Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32(3):333–377, July 1987.

[Chenoweth, 1991] Stephen V. Chenoweth. On the NP-hardness of blocks world. In *Proceedings of the 9th National Conference of the American Association for Artificial Intelligence*, pages 623–628, Anaheim, CA, 1991.

[Dean and Boddy, 1988] Thomas L. Dean and Mark Boddy. Reasoning about partially ordered events. *Artificial Intelligence*, 36(3):375–400, October 1988.

[Fikes and Nilsson, 1971] Richard E. Fikes and Nils Nilsson. STRIPS: A new approach to the application of theorem proving as problem solving. *Artificial Intelligence*, 2:198–208, 1971.

[Garey and Johnson, 1979] Michael R. Garey and David S. Johnson. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.

[Gupta and Nau, 1991] Naresh Gupta and Dana S. Nau. Complexity results for blocks-world planning. In *Proceedings of the 9th National Conference of the American Association for Artificial Intelligence*, pages 629–633, Anaheim, CA, 1991.

[Katsuno and Mendelzon, 1991] Hirofumi Katsuno and Alberto O. Mendelzon. On the difference between updating a knowledge base and revising it. In J. A. Allen, R. Fikes, and E. Sandewall, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 2nd International Conference*, pages 387–394, Cambridge, MA, April 1991.

[Korf, 1987] Richard E. Korf. Planning as search: A quantitative approach. *Artificial Intelligence*, 33:65–88, 1987.

[Levesque, 1988] Hector J. Levesque. Logic and the complexity of reasoning. *Journal of Philosophical Logic*, 17:355–389, 1988.

[Nökel, 1989] Klaus Nökel. Convex relations between time intervals. In J. Rettie and K. Leidlmair, editors, *Proceedings der 5. Österreichischen Artificial Intelligence-Tagung*, pages 298–302. Springer-Verlag, Berlin, Heidelberg, New York, 1989.

[Schoppers, 1987] Marcel J. Schoppers. Universal plans for reactive robots in unpredictable environments. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 1039–1046, Milan, Italy, 1987.

[Vilain *et al.*, 1989] Marc B. Vilain, Henry A. Kautz, and Peter G. van Beek. Constraint propagation algorithms for temporal reasoning: A revised report. In D. S. Weld and J. de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems*, pages 373–381. Morgan Kaufmann, San Mateo, CA, 1989.

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
6750 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all publisched papers so far can be ordered from the above address.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-90-03
Andreas Dengel, Nelson M. Mattos: Integration of Document Representation, Processing and Management
18 pages

RR-90-04
Bernhard Hollunder, Werner Nutt: Subsumption Algorithms for Concept Languages
34 pages

RR-90-05
Franz Baader: A Formal Definition for the Expressive Power of Knowledge Representation Languages
22 pages

RR-90-06
Bernhard Hollunder: Hybrid Inferences in KL-ONE-based Knowledge Representation Systems
21 pages

RR-90-07
Elisabeth André, Thomas Rist: Wissensbasierte Informationspräsentation:
Zwei Beiträge zum Fachgespräch Graphik und KI:
1. Ein planbasierter Ansatz zur Synthese illustrierter Dokumente
2. Wissensbasierte Perspektivenwahl für die automatische Erzeugung von 3D-Objektdarstellungen

24 Seiten

RR-90-08
Andreas Dengel: A Step Towards Understanding Paper Documents
25 pages

RR-90-09
Susanne Biundo: Plan Generation Using a Method of Deductive Program Synthesis
17 pages

RR-90-10
Franz Baader, Hans-Jürgen Bürckert, Bernhard Hollunder, Werner Nutt, Jörg H. Siekmann: Concept Logics
26 pages

RR-90-11
Elisabeth André, Thomas Rist: Towards a Plan-Based Synthesis of Illustrated Documents
14 pages

RR-90-12
Harold Boley: Declarative Operations on Nets
43 pages

RR-90-13
Franz Baader: Augmenting Concept Languages by Transitive Closure of Roles: An Alternative to Terminological Cycles
40 pages

RR-90-14
Franz Schmalhofer, Otto Kühn, Gabriele Schmidt: Integrated Knowledge Acquisition from Text, Previously Solved Cases, and Expert Memories
20 pages

RR-90-15
Harald Trost: The Application of Two-level Morphology to Non-concatenative German Morphology
13 pages

RR-90-16
Franz Baader, Werner Nutt: Adding Homomorphisms to Commutative/Monoidal Theories, or: How Algebra Can Help in Equational Unification
25 pages

RR-90-17
Stephan Busemann: Generalisierte Phasenstrukturgrammatiken und ihre Verwendung zur maschinellen Sprachverarbeitung
114 Seiten

RR-91-01
Franz Baader, Hans-Jürgen Bürckert, Bernhard Nebel, Werner Nutt, Gert Smolka: On the Expressivity of Feature Logics with Negation, Functional Uncertainty, and Sort Equations
20 pages

RR-91-02
Francesco Donini, Bernhard Hollunder, Maurizio Lenzerini, Alberto Marchetti Spaccamela, Daniele Nardi, Werner Nutt: The Complexity of Existential Quantification in Concept Languages
22 pages

RR-91-03
B.Hollunder, Franz Baader: Qualifying Number Restrictions in Concept Languages
34 pages

RR-91-04
Harald Trost: X2MORF: A Morphological Component Based on Augmented Two-Level Morphology
19 pages

RR-91-05
Wolfgang Wahlster, Elisabeth André, Winfried Graf, Thomas Rist: Designing Illustrated Texts: How Language Production is Influenced by Graphics Generation.
17 pages

RR-91-06
Elisabeth André, Thomas Rist: Synthesizing Illustrated Documents A Plan-Based Approach
11 pages

RR-91-07
Günter Neumann, Wolfgang Finkler: A Head-Driven Approach to Incremental and Parallel Generation of Syntactic Structures
13 pages

RR-91-08
Wolfgang Wahlster, Elisabeth André, Som Bandyopadhyay, Winfried Graf, Thomas Rist: WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation
23 pages

RR-91-09
Hans-Jürgen Bürckert, Jürgen Müller, Achim Schupeta: RATMAN and its Relation to Other Multi-Agent Testbeds
31 pages

RR-91-10
Franz Baader, Philipp Hanschke: A Scheme for Integrating Concrete Domains into Concept Languages
31 pages

RR-91-11
Bernhard Nebel: Belief Revision and Default Reasoning: Syntax-Based Approaches
37 pages

RR-91-12
J.Mark Gawron, John Nerbonne, Stanley Peters: The Absorption Principle and E-Type Anaphora
33 pages

RR-91-13
Gert Smolka: Residuation and Guarded Rules for Constraint Logic Programming
17 pages

RR-91-14
Peter Breuer, Jürgen Müller: A Two Level Representation for Spatial Relations, Part I
27 pages

RR-91-15
Bernhard Nebel, Gert Smolka: Attributive Description Formalisms ... and the Rest of the World
20 pages

RR-91-16
Stephan Busemann: Using Pattern-Action Rules for the Generation of GPSG Structures from Separate Semantic Representations
18 pages

RR-91-17
Andreas Dengel, Nelson M. Mattos: The Use of Abstraction Concepts for Representing and Structuring Documents
17 pages

RR-91-18
John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann, Judith Klein: A Diagnostic Tool for German Syntax
20 pages

RR-91-19
Munindar P. Singh: On the Commitments and Precommitments of Limited Agents
15 pages

RR-91-20
Christoph Klauck, Ansgar Bernardi, Ralf Legleitner FEAT-Rep: Representing Features in CAD/CAM
48 pages

RR-91-21
Klaus Netter: Clause Union and Verb Raising Phenomena in German
38 pages

RR-91-22
Andreas Dengel: Self-Adapting Structuring and Representation of Space
27 pages

RR-91-23
Michael Richter, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Akquisition und Repräsentation von technischem Wissen für Planungsaufgaben im Bereich der Fertigungstechnik
24 Seiten

RR-91-24
Jochen Heinsohn: A Hybrid Approach for Modeling Uncertainty in Terminological Logics
22 pages

RR-91-25
Karin Harbusch, Wolfgang Finkler, Anne Schauder: Incremental Syntax Generation with Tree Adjoining Grammars
16 pages

RR-91-26
M. Bauer, S. Biundo, D. Dengler, M. Hecking, J. Koehler, G. Merziger:
Integrated Plan Generation and Recognition
- A Logic-Based Approach -
17 pages

RR-91-27
A. Bernardi, H. Boley, Ph. Hanschke, K. Hinkelmann, Ch. Klauck, O. Kühn, R. Legleitner, M. Meyer, M. M. Richter, F. Schmalhofer, G. Schmidt, W. Sommer:
ARC-TEC: Acquisition, Representation and Compilation of Technical Knowledge
18 pages

RR-91-28
Rolf Backofen, Harald Trost, Hans Uszkoreit:
Linking Typed Feature Formalisms and Terminological Knowledge Representation Languages in Natural Language Front-Ends
11 pages

RR-91-29
Hans Uszkoreit: Strategies for Adding Control Information to Declarative Grammars
17 pages

RR-91-30
Dan Flickinger, John Nerbonne:
Inheritance and Complementation: A Case Study of *Easy* Adjectives and Related Nouns
39pages

RR-91-31
H.-U. Krieger , J. Nerbonne:
Feature-Based Inheritance Networks for Computational Lexicons
11 pages

RR-91-32
Rolf Backofen, Lutz Euler, Günther Görz:
Towards the Integration of Functions, Relations and Types in an AI Programming Language
14 pages

RR-91-33
Franz Baader , Klaus Schulz:
Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures
33 pages

RR-91-35
Winfried Graf, Wolfgang Maaß: Constraint-basierte Verarbeitung graphischen Wissens
14 Seiten

DFKI Technical Memos

TM-90-03
Franz Baader, Bernhard Hollunder: KRIS: Knowledge Representation and Inference System
-System Description-
15 pages

TM-90-04
Franz Baader, Hans-Jürgen Bürckert, Jochen Heinsohn, Bernhard Hollunder, Jürgen Müller, Bernhard Nebel, Werner Nutt, Hans-Jürgen Profitlich: Terminological Knowledge Representation: A Proposal for a Terminological Logic
7 pages

TM-91-01
Jana Köhler: Approaches to the Reuse of Plan Schemata in Planning Formalisms
52 pages

TM-91-02
Knut Hinkelmann: Bidirectional Reasoning of Horn Clause Programs: Transformation and Compilation
20 pages

TM-91-03
Otto Kühn, Marc Linster, Gabriele Schmidt:
Clamping, COKAM, KADS, and OMOS:
The Construction and Operationalization of a KADS Conceptual Model
20 pages

TM-91-04
Harold Boley (Ed.):
A sampler of Relational/Functional Definitions
12 pages

TM-91-05
Jay C. Weber, Andreas Dengel , Rainer Bleisinger:
Theoretical Consideration of Goal Recognition Aspects for Understanding Information in Business Letters
10 pages

TM-91-06
Johannes Stein: Aspects of Cooperating Agents
22 pages

TM-91-08
Munindar P. Singh: Social and Psychological Commitments in Multiagent Systems
11 pages

TM-91-09
Munindar P. Singh: On the Semantics of Protocols Among Distributed Intelligent Agents
18 pages

TM-91-10
Béla Buschauer, Peter Poller, Anne Schauder, Karin Harbusch: Tree Adjoining Grammars mit Unifikation
149 pages

TM-91-11
Peter Wazinski: Generating Spatial Descriptions for Cross-modal References
21 pages

TM-91-12
Klaus Becker, Christoph Klauck, Johannes Schwagereit: FEAT-PATR: Eine Erweiterung des D-PATR zur Feature-Erkennung in CAD/CAM
33 Seiten

TM-91-13
Knut Hinkelmann:
Forward Logic Evaluation: Developing a Compiler from a Partially Evaluated Meta Interpreter
16 pages

DFKI Documents

D-90-05
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Formalismus zur Repräsentation von Geo-metrie- und Technologieinformationen als Teil eines Wissensbasierten Produktmodells
66 Seiten

D-90-06
Andreas Becker: The Window Tool Kit
66 Seiten

D-91-01
Werner Stein , Michael Sintek: Relfun/X - An Experimental Prolog Implementation of Relfun
48 pages

D-91-03
Harold Boley, Klaus Elsbernd, Hans-Günther Hein, Thomas Krause: RFM Manual: Compiling RELFUN into the Relational/Functional Machine
43 pages

D-91-04
DFKI Wissenschaftlich-Technischer Jahresbericht 1990
93 Seiten

D-91-06
Gerd Kamp: Entwurf, vergleichende Beschreibung und Integration eines Arbeitsplanerstellungssystems für Drehteile
130 Seiten

D-91-07
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner
TEC-REP: Repräsentation von Geometrie- und Technologieinformationen
70 Seiten

D-91-08
Thomas Krause: Globale Datenflußanalyse und horizontale Compilation der relational-funktionalen Sprache RELFUN
137 pages

D-91-09
David Powers and Lary Reeker (Eds):
Proceedings MLNLO'91 - Machine Learning of Natural Language and Ontology
211 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-10
Donald R. Steiner, Jürgen Müller (Eds.):
MAAMAW'91: Pre-Proceedings of the 3rd European Workshop on „Modeling Autonomous Agents and Multi-Agent Worlds"
246 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-11
Thilo C. Horstmann: Distributed Truth Maintenance
61 pages

D-91-12
Bernd Bachmann:
HieraCon - a Knowledge Representation System with Typed Hierarchies and Constraints
75 pages

D-91-13
International Workshop on Terminological Logics
Organizers: Bernhard Nebel, Christof Peltason, Kai von Luck
131 pages

D-91-14
Erich Achilles, Bernhard Hollunder, Armin Laux, Jörg-Peter Mohren: *KRIS* : *K*nowledge *R*epresentation and *I*nference *S*ystem
- Benutzerhandbuch -
28 Seiten

On the Computational Complexity of Temporal Projection and some Related Problems

RR-91-34
Research Report

Bernhard Nebel and Christer Bäckström