Paper IV

On the Size of Reactive Plans

Peter Jonsson and Christer Bäckström

Abstract

One of the most widespread approaches to reactive planning is Schoppers' universal plans. We propose a stricter definition of universal plans which guarantees a weak notion of soundness not present in the original definition. Furthermore, we isolate three different types of completeness which capture different behaviours exhibited by universal plans. We show that universal plans which run in polynomial time and are of polynomial size cannot satisfy even the weakest type of completeness unless the polynomial hierarchy collapses. However, by relaxing either the polynomial time or the polynomial space requirement, the construction of universal plans satisfying the strongest type of completeness becomes trivial.

1 Introduction

In recent years reactive planning has been proposed as an alternative to classical planning, especially in rapidly changing, dynamic domains. Although this term has been used for a number of more or less related approaches, these have one thing in common: There is usually very little or no planning ahead. Rather the idea is centered around the stimulus-response principle—prompt reaction to the input. One of the most well-known methods for reactive planning is the *universal plans* by Schoppers [1987]. A universal plan is a function from the set of states into the set of operators. Hence, a universal plan does not generate a sequence of operators leading from the current state to the goal state as a classical planner; it decides after each step what to do next based on the current state.

Universal plans have been much discussed in the literature. In a famous debate (Ginsberg [1989b], Schoppers [1989], Ginsberg [1989a], Schoppers [1994]), Ginsberg criticised the approach while Schoppers defended it[1]. Based on a counting argument, Ginsberg claims that almost all (interesting) universal plans takes an infeasibly large amount of space. Schopper's defence has, to a large extent, built on the observation that planning problems are structured. According to Schoppers, this structure can be exploited in order to create small, effective universal plans. We refrain from going into the details of this debate and merely note that both authors have shown great ingenuity in their argumentation. However, from the standpoint of formal rigour, these papers do not settle the question. One of the few papers that treats universal plans from a formal, complexity-theoretic point of view is the paper by Selman [1994]. He shows that the existence of small (polynomially-sized) universal plans with the ability to generate *minimal* plans implies a collapse of the polynomial hierarchy. Since a collapse of the polynomial hierarchy is widely conjectured to be false in the literature [Johnson, 1990; Papadimitriou, 1994], the existence of such universal plans seems highly unlikely. It should be noted that this result holds even for severely restricted problems such as the blocks-world.

In our opinion, one of the problems with universal plans is the over-generality of the definition. This generality makes formal analysis hard or even impossible. Therefore, we begin this paper by giving a stricter definition of universal plans, a definition that embodies the notion of *soundness*.

[1]This list is not exhaustive. Other authors, such as Chapman [1989], have joined the discussion. However, it seems that the main combatants have been Schoppers and Ginsberg.

In addition, we supply three different types of *completeness*. These notions of completeness capture different desirable properties of universal plans. For example, A-completeness states that if the problem has a solution, then the universal plan will find a solution in a finite number of steps. The main result of this paper is that universal plans which run in polynomial time and are of polynomial size cannot satisfy even this weakest type of completeness[2]. However, by relaxing either the polynomial time requirement or the polynomial space requirement, it becomes trivial to construct universal plans that satisfy the strongest type of completeness. Also in this case, the result holds for severely restricted problems, although of another kind than the blocks-world.

The organisation of the paper is as follows: In section 2 we present the basic STRIPS formalism that underlies the rest of the paper. The topic of section 3 is to formally define universal plans and various restrictions on them. Section 4 is devoted to the proof that shows that small, fast universal plans cannot be complete even in a very weak sense. The paper concludes with a brief discussion of the results.

2 Basic Formalism

We base our work in this paper on the propositional STRIPS formalism with negative goals [Bylander, 1994], which is equivalent to most other variants of propositional STRIPS [Bäckström, 1995].

Definition 2.1 An instance of the *PSN planning problem* is a quadruple $\Pi = \langle \mathcal{P}, \mathcal{O}, \mathcal{I}, \mathcal{G} \rangle$ where

- $\mathcal{P}$ is a finite set of *atoms*;
- $\mathcal{O}$ is a finite set of *operators* where $o \in \mathcal{O}$ has the form $Pre \Rightarrow Post$ where
  - Pre is a satisfiable conjunction of positive and negative atoms in $\mathcal{P}$, respectively called the *positive preconditions* ($pre^+(o)$) and the *negative preconditions* ($pre^-(o)$);
  - $Post$ is a satisfiable conjunction of positive and negative atoms in $\mathcal{P}$, respectively called the *positive postconditions* ($add(o)$) and the *negative postconditions* ($del(o)$);

[2] Under the assumption that the polynomial hierarchy does not collapse.

- $\mathcal{I} \subseteq \mathcal{P}$ denotes the *initial state*;
- and $\mathcal{G} = \langle \mathcal{G}^+, \mathcal{G}^- \rangle$ denote the *positive* and *negative goal*, respectively, satisfying $\mathcal{G}^+, \mathcal{G}^- \subseteq \mathcal{P}$ and $\mathcal{G}^+ \cap \mathcal{G}^- = \varnothing$.

A *PSN structure* is a tuple $\Phi = \langle \mathcal{P}, \mathcal{O} \rangle$ where $\mathcal{P}$ is a set of atoms and $\mathcal{O}$ is a set of operators over $\mathcal{P}$.

We denote the negation of an atom by overlining it. As an example, the operator o defined as $\overline{p} \Rightarrow q, \overline{r}$ satisfies $pre^+(o) = \varnothing$, $pre^-(o) = \{p\}$, $add(o) = \{q\}$ and $del(o) = \{r\}$.

Definition 2.2 Given a set of operators $\mathcal{O}$, we define the set of all operator sequences over $\mathcal{O}$ as $Seqs(\mathcal{O}) = \{\langle\rangle\} \cup \{\langle o \rangle; \omega | o \in \mathcal{O} \text{ and } \omega \in Seqs(\mathcal{O})\}$, where ; is the sequence concatenation operator.

A sequence $\langle o_1, \ldots, o_n \rangle \in Seqs(\mathcal{O})$ of operators is called a *PSN plan* (or simply plan) over Π. We can now define when a plan solves a planning instance.

Definition 2.3 The ternary relation $Valid \subseteq Seqs(\mathcal{O}) \times 2^{\mathcal{P}} \times (2^{\mathcal{P}} \times 2^{\mathcal{P}})$ is defined s.t. for arbitrary $\langle o_1, \ldots, o_n \rangle \in Seqs(\mathcal{O})$ and $S, T^+, T^- \subseteq \mathcal{P}$, $Valid(\langle o_1, \ldots, o_n \rangle, S, \langle T^+, T^- \rangle)$ iff either

1. $n = 0$, $T^+ \subseteq S$ and $T^- \cap S = \varnothing$ or
2. $n > 0$, $pre^+(o_1) \subseteq S$, $pre^-(o_1) \cap S = \varnothing$ and $Valid(\langle o_2, \ldots, o_n \rangle, (S - del(o_1)) \cup add(o_1), \langle T^+, T^- \rangle)$.

A plan $\langle o_1, \ldots, o_n \rangle \in Seqs(\mathcal{O})$ is a *solution* to Π iff $Valid(\langle o_1, \ldots, o_n \rangle, \mathcal{I}, \langle \mathcal{G}^+, \mathcal{G}^- \rangle)$.

We define the planning problems that we will consider as follows.

Definition 2.4 Let $\Pi = \langle \mathcal{P}, \mathcal{O}, \mathcal{I}, \langle \mathcal{G}^+, \mathcal{G}^- \rangle \rangle$ be a given PSN instance. The *plan generation problem* (**PG**) is to find some $\omega \in Seqs(\mathcal{O})$ s.t. ω is a solution to Π or answer that no such ω exists. The *bounded plan generation problem* (**BPG**) takes an integer $K \geq 0$ as additional parameter and the object is to find some $\omega \in Seqs(\mathcal{O})$ s.t. ω is a solution to Π of length $\leq K$ or answer that no such ω exists.

3 Universal Plans

Universal plans are defined as follows in the literature [Ginsberg, 1989b].

> A universal plan is an arbitrary function from the set of possible situations S into the set of primitive actions A.

Using the terminology we have adopted in this paper results in the following equivalent definition.

Definition 3.1 Given a PSN structure $\Phi = \langle \mathcal{P}, \mathcal{O} \rangle$, a universal plan is a function from the set of states $2^{\mathcal{P}}$ into the set of operators $\mathcal{O}$

This very general notion of universal plans is difficult to use as a basis for formal analyses. We would like, for example, to discuss the issuses of correctness and resource consumption. In the sequel, we will try to classify universal plans in greater detail. For a given PSN structure $\Phi = \langle \mathcal{P}, \mathcal{O} \rangle$ let $\mathcal{S} = 2^{\mathcal{P}}$, $\mathcal{S}_\perp = 2^{\mathcal{P}} \cup \{\perp\}$ and $\mathcal{O}^+ = \mathcal{O} \cup \{o_\perp, o_\top\}$. Here $\perp$ is a new state denoting undefinedness and $o_\perp, o_\top$ are two "special" operators. These operators are not to be considered as operators in the sense of Definition 2.1 but rather as two completely new symbols without internal structure. The special operators will be used by the universal plans for communication with the environment. The following definition is needed for defining soundness of universal plans.

Definition 3.2 Let $\Phi = \langle \mathcal{P}, \mathcal{O} \rangle$ be a PSN structure. The *update* operator $\oplus : \mathcal{S}_\perp \times \mathcal{O}^+ \to \mathcal{S}_\perp$ is defined as follows: $\perp \oplus o = \perp$ for all $o \in \mathcal{O}^+$. If o is a standard operator then

$$S \oplus o = \begin{cases} (S - del(o)) \cup add(o) & \text{if } pre^+(o) \subseteq S \wedge pre^-(o) \cap S = \varnothing \\ \perp & \text{otherwise.} \end{cases}$$

Otherwise, $S \oplus o_\perp = \perp$ and $S \oplus o_\top = S$. An operator $o \in \mathcal{O}^+$ is *admissible* in a state $S \in \mathcal{S}_\perp$ iff $S \oplus o \neq \perp$.

We can now refine our notion of universal plans.

Definition 3.3 Let $\Phi = \langle \mathcal{P}, \mathcal{O} \rangle$ be a PSN structure and let $\mathcal{G}$ be a goal over $\mathcal{P}$. A *sound universal plan* $U_{\mathcal{G}}$ for the goal $\mathcal{G}$ is a function that maps $\mathcal{S}_\perp$ to $\mathcal{O}^+$ such that

1. for every $S \in \mathcal{S}_\perp$, if $U_\mathcal{G}(S) = o \in \mathcal{O}$ then o is admissible in S;
2. for every $S \in \mathcal{S}_\perp$, $U_\mathcal{G}(S) = o_\top$ iff S satisfies $\mathcal{G}$;

The first point in the definition says that if the universal plan generates an operator, then it is executable in the current state. This restriction seems to have been tacitly assumed in the literature. The second point tells us that the special operator $o_\top$ is generated if and only if the universal plan is applied to a state satisfying the goal state. Thus, $o_\top$ is used by $U_\mathcal{G}$ to report success. The reason for introducing the operator $o_\top$ is to avoid the generation of new operators when the current state satisfies the goal state. The special operator $o_\perp$, on the other hand, indicates that the universal plan cannot handle the current state. This can, for instance, be due to the fact that the goal state is not reachable from the current state. Observe that no operator is admissible in $\perp$ so $U_\mathcal{G}$ must generate $o_\perp$ whenever applied to $\perp$. Henceforth, we will use the term universal plan as an abbreviation for sound universal plan.

We continue by defining four properties of universal plans. For a universal plan $U_\mathcal{G}$ we use the notation $U_\mathcal{G}^K(S)$ to denote the operator $U_\mathcal{G}(S_K)$ where $S_1 = S$ and $S_{K+1} = S_K \oplus U_\mathcal{G}(S_K)$.

Definition 3.4 A universal plan $U_\mathcal{G}$ for a PSN structure $\Phi = \langle \mathcal{P}, \mathcal{O} \rangle$ is

P$_T$ *poly-time* iff $U_\mathcal{G}$ can be implemented as a deterministic algorithm that runs in polynomial time in the size of Φ;

P$_S$ *poly-space* iff $U_\mathcal{G}$ can be implemented as a deterministic algorithm A satisfying

1. the size of A is polynomially bounded by the size Φ; and
2. the size of the space used by A is polynomially bounded by the size of Φ;

A *acceptance-complete* iff for every $S \in \mathcal{S}$ such that $\langle \mathcal{P}, \mathcal{O}, S, \mathcal{G} \rangle$ is solvable there exists an integer K such that $U_\mathcal{G}^K(S) = o_\top$;

R *rejection-complete* iff for every $S \in \mathcal{S}$ such that $\langle \mathcal{P}, \mathcal{O}, S, \mathcal{G} \rangle$ is not solvable there exists an integer K such that $U_\mathcal{G}^K(S) = o_\perp$.

Universal plans satisfying some subset of the restrictions P$_T$, P$_S$, A and R are named by combining the corresponding letters. For example, a P$_T$AR universal plan is poly-time, acceptance-complete and rejection-complete. The definition of poly-time should be quite clear while the definition of poly-space may need further explanation. The first part of the definition ensures that $U_{\mathcal{G}}$ can be stored in a polynomially-bounded memory. The second part guarantees that any computation will use only a polynomially-bounded amount of auxiliary memory. Hence, we can both store and run the algorithm in a memory whose size is bounded by a polynomial in the size of Φ. This restriction excludes algorithms using extremely large fixed data structures as well as algorithms building such structures during run-time.

For the sake of brevity, we use the terms A- and R-completeness for acceptance- and rejection-completeness, respectively. A minimal requirement on universal plans is that they are A-complete so we are guaranteed to find a solution within a finite number of steps if there is one. Observe that if an A-complete universal plan is not R-complete then $U^K_{\mathcal{G}}(S)$ can differ from $o_\perp$ for all K if $\mathcal{G}$ is not reachable from S. R-completeness is, thus, desirable but not always necessary. In domains such as the blocks-world, where we know that a solution exists in advance, R-completeness is of minor interest. To have R-completeness without A-completeness is useless since we can trivially construct universal plans satisfying $\mathrm{P}_{T,S}\mathrm{R}$ for all problems. Simply let $U_{\mathcal{G}}(S) = o_\perp$ for all $S \in \mathcal{S}_\perp$. This R-complete universal plan can trivially be implemented as a poly-time and poly-space deterministic algorithm.

In certain applications, we need a stronger form of R-completeness.

Definition 3.5 A universal plan $U_{\mathcal{G}}$ for a PSN structure $\langle \mathcal{P}, \mathcal{O} \rangle$ is *strongly rejection-complete* (R^+) iff for every $S \in \mathcal{S}$ such that $\langle \mathcal{P}, \mathcal{O}, S, \mathcal{G} \rangle$ is not solvable, $U_{\mathcal{G}}(S) = o_\perp$.

The motivation for introducing strong R-completeness is simple. If the universal plan outputs operators, we cannot know whether they will lead to a solution or not. Executing such operators is not advisable, since we may wish to try planning for some alternative goal if there is no solution for the first one. However, executing the "invalid" operators may prevent us from reaching the alternative goal.

From a complexity-theoretic point of view, it can be argued that universal plans have to be both poly-time and poly-space to be feasible in practice. This is a hard restriction since by dropping any of the polynomiality requirements, constructing universal plans become easy.

Theorem 3.6 For every PSN structure $\Phi = \langle \mathcal{P}, \mathcal{O} \rangle$ and goal state $\mathcal{G}$ over $\mathcal{P}$ there exist universal plans $U_{\mathcal{G}}$ and $U'_{\mathcal{G}}$ satisfying $\mathrm{P}_T\mathrm{AR}^+$ and $\mathrm{P}_S\mathrm{AR}^+$, respectively.

Proof: *Construction of $U_{\mathcal{G}}$*: We define a function $f : \mathcal{S}_{\perp} \rightarrow \mathcal{O}^{+}$ as follows. For each $K \geq 1$ and $S \in \mathcal{S}$ such that $\langle \mathcal{P}, \mathcal{O}, S, \mathcal{G} \rangle$ has a shortest solution of length K, choose an $o \in \mathcal{O}$ such that $\langle \mathcal{P}, \mathcal{O}, S \oplus o, \mathcal{G} \rangle$ has a shortest solution of length $K - 1$. Denote this operator o_S. Now, for all states $S \in \mathcal{S}_{\perp}$, let

$$f(S) = \begin{cases} o_{\perp} & \text{if } \langle \mathcal{P}, \mathcal{O}, S, \mathcal{G} \rangle \text{ is not solvable} \\ o_{\top} & S \text{ satisfies } \mathcal{G} \\ o_S & \text{otherwise} \end{cases}$$

Clearly, for every $S \in \mathcal{S}$ there exists an integer K such that if $\langle \mathcal{P}, \mathcal{O}, S, \mathcal{G} \rangle$ is solvable then $U_{\mathcal{G}}^{K}(S) = o_{\top}$. Otherwise, $U_{\mathcal{G}}(S) = o_{\perp}$. Consequently, f is both A-complete and strongly R-complete. The proposed construction of the function f is obviously of exponential size. However, it can be arranged as a balanced decision tree of depth $|\mathcal{P}|$ and, hence, be accessed in polynomial time. Consequently, we have constructed $U_{\mathcal{G}}$.

Construction of $U'_{\mathcal{G}}$: Consider a forward-chaining PSN planning algorithm P that is sound, complete and generates shortest plans. We modify the algorithm to output only the first operator of the plan that leads from S to $\mathcal{G}$. Since a plan might be of exponential size this cannot necessarily be implemented in polynomial space. However, we can guess the plan one operator at a time and compute the resulting state after each action, using only polynomial space. Hence, this modified planner can be represented by a non-deterministic algorithm using polynomial space. Thus, by Savitch's theorem [Savitch, 1970], it can also be represented by a deterministic algorithm that uses polynomial space. This modified planner can be the same for all problems simply by giving the PSN structure Φ and the goal state $\mathcal{G}$ as additional inputs. Hence, it is of constant size, *i.e* its size does not depend on the size of the given PSN structure. Consequently, we can disregard the size of the planner and we have constructed a poly-space universal plan. (Observe that the soundness of P implies soundness of $U'_{\mathcal{G}}$ if we modify $U'_{\mathcal{G}}$ to generate $o_{\top}$ whenever the current state satisfies the goal state.)

The planner P is complete and generates minimal plans. Hence, if the shortest plan from the current state S to the goal state $\mathcal{G}$ is of length L, the length of the shortest plan from $S \oplus U'_{\mathcal{G}}(S)$ to $\mathcal{G}$ is $L-1$. By this observation and the fact that P is complete, A-completeness of $U'_{\mathcal{G}}$ follows.

Finally, if there is no plan from the current state to the goal state, the planner will fail to generate even the first operator. In this case we simply output $o_{\perp}$ and strong R-completeness follows. $\square$

It is crucial that the planner used in the previous theorem generates shortest plans. Otherwise, we cannot guarantee A-completeness. We illustrate this with a small, contrived example.

Example 3.1 Consider the following PSN structure $\Phi = \langle \mathcal{P}, \mathcal{O} \rangle = \langle \{p, q\}, \{p^+, q^+, q^-\} \rangle$ where the operators are defined as follows.

- $p^+ : \overline{p} \Rightarrow p$;
- $q^- : q \Rightarrow \overline{q}$;
- $q^+ : \overline{q} \Rightarrow q$.

Let $\mathcal{I}_1 = \{q\}$, $\mathcal{I}_2 = \varnothing$, $\mathcal{G} = \langle \{p\}, \varnothing \rangle$, $\Pi_1 = \langle \mathcal{P}, \mathcal{O}, \mathcal{I}_1, \mathcal{G} \rangle$ and $\Pi_2 = \langle \mathcal{P}, \mathcal{O}, \mathcal{I}_2, \mathcal{G} \rangle$. The shortest plan for both Π_1 and Π_2 is $\langle p^+ \rangle$. Assume a planning algorithm A that generates the plan $\omega_1 = \langle q^-, p^+ \rangle$ for Π_1 and $\omega_2 = \langle q^+, p^+ \rangle$ for Π_2. A universal plan $U_{\mathcal{G}}$ based on A would then satisfy $U_{\mathcal{G}}(\mathcal{I}_1) = q^+$ and $U_{\mathcal{G}}(\mathcal{I}_2) = q^-$. Consequently, $U_{\mathcal{G}}^K(\mathcal{I}_1) = q^+$ for K odd and $U_{\mathcal{G}}(\mathcal{I}_1) = q^-$ for K even. In other words, the universal plan will toggle q forever. Hence, $U_{\mathcal{G}}$ is not A-complete.

For planning problems such that **BPG**[3] can be solved in polynomial-time, we can construct universal plans satisfying $\mathrm{P}_{T,S}\mathrm{AR}^+$ by Theorem 3.6. For planning problems such that **PG** is polynomial but **BPG** is not, the theorem does not apply. This method for constructing universal plans is pointed out in Selman [1994] but he does not explicitly state that generating the shortest plan is necessary. The question whether we can construct $\mathrm{P}_{T,S}\mathrm{AR}^+$ universal plans for problems where **PG** is polynomial but **BPG** is not remains open.

4 Non-Existence of $\mathbf{P}_{T,S}\mathbf{A}$ Universal Plans

In order to show that $\mathrm{P}_{T,S}\mathrm{A}$ universal plans do not exist for all PSN planning problems, we will use *advice-taking* Turing machines [Johnson, 1990]. Advice-taking TMs are an alternative way of describing non-uniform circuits, which is the approach adopted by Selman [1994].

Definition 4.1 An *advice-taking* Turing machine is a TM T that has associated with it a special "advice oracle", A, one that is a (not necessarily

[3] Recall that **BPG** and **PG** denote the bounded and unbounded plan generation problem respectively.

computable) function. Let x be an arbitrary input string and let $|x|$ denote the size of x. When T is applied to x, a special "advice tape" is automatically loaded with $A(|x|)$ and from then on the computation proceeds as normal, based on the two inputs, x and $A(|x|)$. An advice-taking Turing machine uses *polynomial advice* iff its advice oracle satisfies $|A(n)| \leq p(n)$ for some fixed polynomial p and all nonnegative integers n. The class P/poly is the set of languages defined by polynomial-time advice-taking TMs.

Advice-taking TMs are very powerful. They can, for instance, compute certain undecidable functions. Despite their apparent power, it is highly unlikely that NP problems can be solved by P/poly TMs.

Theorem 4.2 (Karp and Lipton [1982]) If NP $\subseteq$ P/poly then the polynomial hierarchy collapses into Σ_2^p.

Σ_2^p is a complexity class in the second level of the polynomial hierarchy [Johnson, 1990]. Collapse of the polynomial hierarchy is widely conjectured to be false in the literature [Johnson, 1990; Papadimitriou, 1994]. Our proofs rely on the following construction.

Lemma 4.3 Let $\mathcal{F}_n$ be the set of all 3SAT [Garey and Johnson, 1979] instances with n variables. For every n, there is a PSN structure $\Theta_n = \langle \mathcal{P}, \mathcal{O} \rangle$ and a goal state $\mathcal{G}_n$ such that for every $F \in \mathcal{F}_n$, there exists an $\mathcal{I}_F$ with the following property: $\Pi_F = \langle \mathcal{P}, \mathcal{O}, \mathcal{I}_F, \mathcal{G}_n \rangle$ is a planning instance which is solvable iff F is satisfiable. Furthermore, any solution to Π_F must have a length less than or equal to $8n^3 + 2n$.

Proof: Let $U = \{u_1, \ldots, u_n\}$ be the set of variables used by the formulae in $\mathcal{F}_n$. Observe that there can only be $(2n)^3$ different clauses in any formula in $\mathcal{F}_n$. Let $\mathcal{C} = \{C_1, \ldots, C_{8n^3}\}$ be an enumeration of the possible clauses over the variable set U. Let $\mathcal{P} = \{T(i), F(i), C(j) | 1 \leq i \leq n,\ 1 \leq j \leq 8n^3\}$. The atoms will have the following meanings: $T(i)$ is true iff the variable u_i is true, $F(i)$ is true iff the variable u_i is false and $C(j)$ is true iff the clause C_j is satisfied. For each variable u_i, two operators are needed:

- $\overline{T(i)}, \overline{F(i)} \Rightarrow T(i)$,
- $\overline{T(i)}, \overline{F(i)} \Rightarrow F(i)$.

That is, $T(i)$ can be made true iff $F(i)$ is false and vice versa. In this fashion, only one of $T(i)$ and $F(i)$ can be true. For each case where a clause $C(j) \in \mathcal{C}$ contains a variable u_i, the first operator below is needed: for a negated variable $\neg u_i$, the second operator is needed:

- $T(i), \overline{C(j)} \Rightarrow C(j)$,
- $F(i), \overline{C(j)} \Rightarrow C(j)$.

We specify the goal such that $\mathcal{G}_n = \langle \mathcal{G}_n^+, \mathcal{G}_n^- \rangle = \langle \{C_1, \ldots, C_{8n^3}\}, \varnothing \rangle$. Let $F \in \mathcal{F}$. We want to construct an initial state $\mathcal{I}_F$ such that $\Pi = \langle \mathcal{P}, \mathcal{O}, \mathcal{I}_F, \mathcal{G}_n \rangle$ is solvable iff F is satisfiable. Let $\mathcal{I}_F = \{C(j) | C(j) \notin F\}$. Clearly, every $C(j)$ can be made true iff a satisfying assignment for F can be found. Finally, it is easy to see that any solution to Π_F must be of length $\leq 8n^3 + 2n$ since we have exactly $8n^3 + 2n$ atoms and each atom can be made true at most once. $\square$

Lemma 4.4 If, for every integer $n \geq 1$, there exists a polynomial advice function that allows us to solve Π_F for all $F \in \mathcal{F}_n$ in polynomial time, then the polynomial hierarchy collapses into Σ_2^p.

Proof: Suppose Π_F is solvable iff F has a satisfying truth assignment, then NP $\subseteq$ P/poly so, by Theorem 4.2, the polynomial hierarchy collapses into Σ_2^p. $\square$

We can now prove our main theorem.

Theorem 4.5 If there exists a universal plan $U_{\mathcal{G}_n}$ satisfying $\mathrm{P}_{T,S}\mathrm{A}$ for Θ_n, $n \geq 1$, then the polynomial hierarchy collapses into Σ_2^p.

Proof: Assume $U_{\mathcal{G}_n}$ to be a $\mathrm{P}_{T,S}\mathrm{A}$ universal plan for Θ_n. Consider the algorithm A in Figure 1. $U_{\mathcal{G}_n}$ is sound so it must generate an operator that is admissible in the given state or generate one of the special operators $o_\perp, o_\top$. Hence, by Lemma 4.3, the **repeat** loop can iterate at most $8n^3 + 2n$ times before o equals either $o_\perp$ or $o_\top$. We have assumed that $U_{\mathcal{G}_n}$ is a polynomial-time algorithm so algorithm A runs in polynomial time. We show that algorithm A accepts iff F has a satisfying truth assignment. The if-part is trivial by noting that if F has a satisfying truth assignment then the algorithm accepts by A-completeness. For the only-if part, assume that the algorithm accepts. Then $U_{\mathcal{G}_n}$ has returned the operator $o_\top$ when applied to some state S. By Definition 3.3, $U_{\mathcal{G}_n}(S) = o_\top$ iff S satisfies $\mathcal{G}_n$. Consequently, F is satisfiable by Lemma 4.3. Hence, the algorithm accepts iff F is satisfiable and rejects iff F is not satisfiable. Furthermore, $U_{\mathcal{G}_n}$ is a polynomial advice function since we have restricted $U_{\mathcal{G}_n}$ to be of polynomial size and the theorem follows by Lemma 4.4. $\square$

```
Algorithm A.
Input: A 3SAT formula F with n variables.
S ← I_F
 repeat
   o ← U_{G_n}(S)
   S ← S ⊕ o
until o ∈ {o_⊥, o_⊤}
if o = o_⊤  then accept
else reject
```

Figure 1: The algorithm used in the proof of Theorem 4.5.

The generality of this theorem has to be emphasized. Recall that an advice is an *arbitrary* function from the size of the input. This function does not even have to be computable. Hence, there does not exist any mechanism whatsoever that is of polynomial size and can be accessed in polynomial time with the ability to solve problems like those exhibited in the previous theorem. Methods that have been proposed to reduce the size of universal plans, such as the *variables* introduced by Schoppers [1994], cannot change this fact.

Moreover, observe that Theorem 4.5 applies even to a class of severely restricted PSN structures. The restrictions are, among others, that all delete-lists are empty and each operator has at most two preconditions. Since the delete-lists are empty, this restricted class is in NP [Bylander, 1994]. Consequently, it is a class with considerably less expressive power than the general PSN planning problem which is PSPACE-complete[4]. Yet, $P_{T,S}A$ universal plans do not exist for this class of planning problems. Note that this is not caused by the existence of exponentially-size minimal plans [Bäckström and Nebel, 1995] since all minimal plans in this class are polynomially bounded.

Finally, we would like to compare Theorem 4.5 with a negative result by Selman [1994].

Theorem 4.6 Unless NP$\subseteq$P/poly, there exists a blocks-world planning goal for which there is no $P_{T,S}A$ universal plan for generating the minimal sequence of operators leading to the goal.

[4] Under the plausible assumption that NP$\neq$PSPACE

It is important to note the difference between this theorem and Theorem 4.5. Where Selman shows that $P_{T,S}A$ universal plans cannot generate minimal plans under certain conditions, we show that there are cases when they cannot generate any plans at all.

5 Discussion

The results in this paper should not be interpreted too negatively. What they tell us is that naïve approaches to universal planning will not work. In particular, we cannot hope for efficient universal plans solving arbitrary planning problems. However, we question only the efficiency of universal plans. We do not claim universal plans to be inferior to classical planners in all aspects. It is, for instance, highly probable that universal planning can offer great advantages over classical planning in rapidly changing, dynamic domains. Thus, one of the challenges for the future is to characterize which planning problems can be efficiently solved by universal plans. We have seen that if a problem can be solved optimally in polynomial time, then there is an efficient universal plan solving it. Almost certainly, there are other interesting classes of planning problems that can be solved by small, fast universal plans.

Another question to be answered in the future is how to make universal planning more powerful. Several approaches are conceivable. One would be to give universal plans access to random sources—thus making universal planning probabilistic. Recent research has shown that probabilistic algorithms can be surprisingly efficient for certain types of problems. To mention one example, the probabilistic GSAT algorithm [Selman *et al.*, 1992] for satisfiability testing of propositional formulae has shown good performance in empirical studies [Selman and Kautz, 1993]. Another extension would be to allow universal plans to have an internal state; that is, the output of the universal plan is not only dependent on the current state, but also on previous states. Universal plans with internal states have been studied briefly by Selman [1994]. The results are unfortunately not encouraging.

Universal planning should also be compared with *incremental planning* [Ambros-Ingerson and Steel, 1988; Jonsson and Bäckström, 1995]. The idea behind incremental planning is to have a planner that can output valid prefixes of the final plan before it has finished planning. It has been argued that this method could considerably bring down the time lost in planning, especially in dynamic domains, where replanning has to occur frequently. This motivation is almost exactly the same as the motivation for introducing universal plans (or reactive planning in general). Here we have a spectrum of different approaches to planning ranging from classical planning which first computes the complete plan and then executes it, via incremental planning, where chunks of the plan are generated and executed in an interleaved fashion, to universal planning, where just one operator at a time is generated and immediately executed.

6 Conclusions

We have proposed a stricter definition of universal plans which guarantees a weak notion of soundness not present in the original definition. In addition, we have identified three different types of completeness which capture different behaviours exhibited by universal plans. A-completeness guarantees that if there exists a plan from the current state to the goal state, then the universal plan will find a solution in a finite number of steps. R-completeness is the converse of A-completeness, *i.e.* if there does not exist a plan from the current state to the goal state, then the universal plan will report this after a finite number of applications. R^+-completeness is a stronger version of R-completeness, stating that if there does not exist a plan from the current state to the goal state, then the universal plan will report this after one application. We show that universal plans which run in polynomial time and are of polynomial size cannot be A-complete unless the polynomial hierarchy collapses. However, by dropping either the polynomial time or the polynomial space requirement, the construction of A- and R^+-complete universal plans becomes trivial.

References

[Ambros-Ingerson and Steel, 1988] José A. Ambros-Ingerson and Sam Steel. Integrating planning, execution and monitoring. In *Proceedings of the 7th (US) National Conference on Artificial Intelligence (AAAI-88)*, pages 83–88, St. Paul, MN, USA, August 1988. American Association for Artificial Intelligence, Morgan Kaufmann.

[Bäckström and Nebel, 1995] Christer Bäckström and Bernhard Nebel. Complexity results for SAS^+ planning. *Computational Intelligence*, 11(4):625–655, 1995.

[Bäckström, 1995] Christer Bäckström. Expressive equivalence of planning formalisms. *Artificial Intelligence*, 76(1–2):17–34, 1995.

[Bylander, 1994] Tom Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69:165–204, 1994.

[Chapman, 1989] David Chapman. Penguins can make cake. *AI Magazine*, pages 45–50, Winter 1989.

[Garey and Johnson, 1979] Michael Garey and David Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[Ginsberg, 1989a] Matthew L Ginsberg. Ginsberg replies to Chapman and Schoppers. *AI Magazine*, pages 61–62, Winter 1989.

[Ginsberg, 1989b] Matthew L Ginsberg. Universal planning: An (almost) universally bad idea. *AI Magazine*, pages 40–44, Winter 1989.

[Johnson, 1990] David S Johnson. A catalog of complexity classes. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science: Algorithms and Complexity*, volume A, chapter 2, pages 67–161. Elsevier, Amsterdam, 1990.

[Jonsson and Bäckström, 1995] Peter Jonsson and Christer Bäckström. Incremental planning. In Malik Ghallab and Alfredo Milani, editors, *New Directions in AI Planning: EWSP'95—3rd European Workshop on Planning*, Frontiers in AI and Applications, Assisi, Italy, September 1995. IOS Press.

[Karp and Lipton, 1982] R M Karp and R Lipton. Turing machines that take advice. *Enseign. Math*, 28:191–209, 1982.

[Papadimitriou, 1994] Christos H. Papadimitriou. *Computational Complexity*. Addison Wesley, Reading, MA, 1994.

[Savitch, 1970] W J Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

[Schoppers, 1987] M J Schoppers. Universal plans for reactive robots in unpredictable environments. In John McDermott, editor, *Proceedings of the 10th International Joint Conference on Artificial Intelligence (IJCAI-87)*, pages 1039–1046, Milano, Italy, August 1987. Morgan Kaufmann.

[Schoppers, 1989] Marcel J Schoppers. In defense of reaction plans as caches. *AI Magazine*, pages 51–62, Winter 1989.

[Schoppers, 1994] Marcel Schoppers. Estimating reaction plan size. In *Proceedings of the 12th (US) National Conference on Artificial Intelligence (AAAI-94)*, pages 1238–1244, Seattle, WA, USA, July–August 1994. American Association for Artificial Intelligence.

[Selman and Kautz, 1993] Bart Selman and Henry Kautz. An empirical study of greedy local search for satisfiability testing. In *Proceedings of the 11th (US) National Conference on Artificial Intelligence (AAAI-93)*, pages 46–51, Washington DC, USA, July 1993. American Association for Artificial Intelligence.

[Selman *et al.*, 1992] Bart Selman, Hector Levesque, and David Mitchell. A new method for solving hard satisfiability problems. In *Proceedings of the 10th (US) National Conference on Artificial Intelligence (AAAI-92)*, pages 440–446, San José, CA, USA, July 1992. American Association for Artificial Intelligence.

[Selman, 1994] Bart Selman. Near-optimal plans, tractability, and reactivity. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proceedings of the 4th International Conference on Principles on Knowledge Representation and Reasoning (KR-94)*, pages 521–529, Bonn, Germany, May 1994. Morgan Kaufmann.