COMPLEXITY RESULTS FOR STATE-VARIABLE PLANNING UNDER MIXED SYNTACTICAL AND STRUCTURAL RESTRICTIONS

PETER JONSSON
Department of Computer and Information Science
Linköping University, S-581 83 Linköping, Sweden
email: petej@ida.liu.se

and

CHRISTER BÄCKSTRÖM
Department of Computer and Information Science
Linköping University, S-581 83 Linköping, Sweden
email: cba@ida.liu.se

ABSTRACT

Most tractable planning problems reported in the literature have been defined by syntactical restrictions. To better exploit the inherent structure of problems, however, it is probably necessary to study also structural restrictions on the state-transition graph. We present an almost exhaustive map of complexity results for state-variable planning under all combinations of our previously analysed syntactical (P, U, B, S) and structural (I, A, O) restrictions, considering both optimal and non-optimal plan generation.

1 Introduction

Many planning problems in manufacturing and process industry are believed to be highly structured, thus allowing for efficient planning if exploiting this structure. However, a 'blind' domain-independent planner will most likely go on tour in an exponential search space even for tractable problems. Although heuristics may help a lot, they are often not based on a sufficiently thorough understanding of the underlying problem structure to guarantee efficiency and correctness. Further, we believe that if having such a deep understanding of the problem structure, it is better to use other methods than heuristics.

Some tractability results have been reported in the literature for restrictions on the propositional STRIPS formalism[7,9] and for restrictions on the related state-variable formalism SAS$^+$.[5,6] These results are all based on essentially syntactic restrictions on the set of operators. Syntactic restrictions are very appealing to study, since, typically, they are easy to define and not very costly to test. However, to gain any deeper insight into what makes planning problems hard and easy respectively probably require that we study the structure of the problem, in particular the state-transition graph induced by the operators. Putting explicit restrictions on the state-transition graph must be done with great care, however, since this graph is typically of size exponential in the size of the planning problem instance, making it extremely costly to test arbitrary properties.

In a recent paper[13] we took an intermediate approach. Using the SAS$^+$ formalism we defined restrictions not on the whole state-transition graph, but on the domain-transition graph for each state variable in isolation. These can, hence, be tested in polynomial time. Although not being a substitute for restrictions on the whole state-transition graph, many interesting and useful properties of this graph can be indirectly exploited. In particular, we identified three structural restrictions (I, A and O) which together make planning tractable and properly generalize the tractable problems we have previously defined using syntactical restrictions. We also presented a polynomial-time, sound and complete algorithm for generating optimal plans under the new structural restrictions. Despite being structural, our restrictions can be tested in polynomial time. Further, this approach would not be very useful for a planning formalism based on propositional atoms, since the resulting two-vertex domain-transition graphs would not allow for much structure to exploit.

We have previously[6] presented a map over the complexity of planning for all combinations of the previously considered syntactical restrictions on SAS$^+$ planning. In this paper, we repeat this endeavour, taking also the new structural restrictions into account. We provide a map over the complexity of both optimal and non-optimal plan generation for all combinations of the restrictions, considering also mixed structural and syntactical restrictions. Hence, we augment our previous tractability result for the SAS*-IAO problem by also showing that it is a maximally tractable problem under the restrictions considered. (For reasons explained later in the paper, we actually study a restricted version of the SAS$^+$ formalism.)

2 The SAS$^+$ and SAS* formalisms

The SAS$^+$ formalism,[4,6] is a variant of propositional STRIPS, generalizing the atoms to multi-valued state variables. Furthermore, what is called a precondition in STRIPS is here divided into two conditions, the precondition and the prevailcondition. Variables which are required and changed by an operator go into the precondition and those which remain unchanged, but are required, go into the prevailcondition. We briefly recapitulate the SAS$^+$ formalism below, referring to Bäckström and Nebel[6] for further explanation.

Definition 2.1 *An instance of the SAS$^+$ planning problem is given by a tuple* $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ *with components defined as follows:*

- $\mathcal{V} = \{v_1, \ldots, v_m\}$ *is a set of* **state variables**. *Each variable* $v \in \mathcal{V}$ *has an associated* **domain** $\mathcal{D}_v$, *which implicitly defines an* **extended domain** $\mathcal{D}_v^+ = \mathcal{D}_v \cup \{\mathsf{u}\}$, *where* u *denotes the* **undefined value**. *Further, the* **total state space** $\mathcal{S} = \mathcal{D}_{v_1} \times \ldots \times \mathcal{D}_{v_m}$ *and the* **partial state space** $\mathcal{S}^+ = \mathcal{D}_{v_1}^+ \times \ldots \times \mathcal{D}_{v_m}^+$ *are implicitly defined. We write* $s[v]$ *to denote the value of the variable* v *in a state* s.

- *$\mathcal{O}$ is a set of* **operators** *of the form $\langle \mathsf{b}, \mathsf{e}, \mathsf{f} \rangle$, where $\mathsf{b}, \mathsf{e}, \mathsf{f} \in \mathcal{S}^+$ denote the* **pre-, post-** *and* **prevail-condition** *respectively. Each operator $\langle \mathsf{b}, \mathsf{e}, \mathsf{f} \rangle \in \mathcal{O}$ is subject to the following two restrictions*

  (R1) *for all $v \in \mathcal{V}$ if $\mathsf{b}[v] \neq \mathsf{u}$, then $\mathsf{b}[v] \neq \mathsf{e}[v] \neq \mathsf{u}$,*

  (R2) *for all $v \in \mathcal{V}$, $\mathsf{e}[v] = \mathsf{u}$ or $\mathsf{f}[v] = \mathsf{u}$.*

- *$s_0 \in \mathcal{S}^+$ and $s_* \in \mathcal{S}^+$ denote the* **initial state** *and* **goal state** *respectively.*

Restriction R1 essentially says that a state variable can never be made undefined, once made defined by some operator. Restriction R2 says that the pre- and prevail-conditions of an operator must never define the same variable.

We write $s \sqsubseteq t$ if the state s is subsumed (or satisfied) by state t, ie. if $s[v] = \mathsf{u}$ or $s[v] = t[v]$. We extend this notion to whole states, defining

$$s \sqsubseteq t \text{ iff } \quad \text{for all } v \in \mathcal{V}, s[v] = \mathsf{u} \text{ or } s[v] = t[v].$$

If $o = \langle \mathsf{b}, \mathsf{e}, \mathsf{f} \rangle$ is a SAS$^+$ operator, we write $\mathsf{b}(o)$, $\mathsf{e}(o)$ and $\mathsf{f}(o)$ to denote b, e and f respectively. $\mathcal{O}^*$ denotes the set of operator sequences over $\mathcal{O}$ and the members of $\mathcal{O}^*$ are called **plans**. *Given two states $s, t \in \mathcal{S}^+$, we define for all $v \in \mathcal{V}$,*

$$(s \oplus t)[v] = \begin{cases} t[v] & \text{if } t[v] \neq \mathsf{u}, \\ s[v] & \text{otherwise.} \end{cases}$$

The ternary relation Valid $\subseteq \mathcal{O}^* \times \mathcal{S}^+ \times \mathcal{S}^+$ is defined recursively s.t. for arbitrary operator sequence $\langle o_1, \ldots, o_n \rangle \in \mathcal{O}^*$ and arbitrary states $s, t \in \mathcal{S}^+$, Valid$(\langle o_1, \ldots, o_n \rangle, s, t)$ iff either

1. *$n = 0$ and $t \sqsubseteq s$ or*
2. *$n > 0$, $\mathsf{b}(o_1) \sqsubseteq s$, $\mathsf{f}(o_1) \sqsubseteq s$ and Valid$(\langle o_2, \ldots, o_n \rangle, (s \oplus \mathsf{e}(o_1)), t)$.*

A plan $\langle o_1, \ldots, o_n \rangle \in \mathcal{O}^*$ **solves** *Π iff Valid$(\langle o_1, \ldots, o_n \rangle, s_0, s_*)$.*

Finally, we define a restricted variant of the SAS$^+$ formalism.

Definition 2.2 *The SAS* problem is the SAS$^+$ problem restricted to instances $\langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ satisfying (1) $s_* \in \mathcal{S}$ and (2) for every operator $o \in \mathcal{O}$ and variable $v \in \mathcal{V}$, if $\mathsf{b}(o)[v] = \mathsf{u}$, then $\mathsf{e}(o)[v] = \mathsf{u}$*

3 Restrictions

In this section we define the various restrictions on SAS$^+$ planning to be analysed in the next section. We have previously presented both the syntactical ones (P, U, B and S)[5,6] and three of the structural ones (I, A and O).[13] In addition we present a new structural restriction, A$^+$. Before presenting the restrictions we must define some other concepts, however. Assume below that $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ is a SAS$^+$ instance.

Definition 3.1 *An operator $o \in \mathcal{O}$ is* **unary** *iff there is exactly one $v \in \mathcal{V}$ s.t.* $\mathsf{e}(o)[v] \neq \mathsf{u}$.

Definition 3.2
For each $v \in \mathcal{V}$ and $\mathcal{O}' \subseteq \mathcal{O}$, the set $\mathcal{R}_v^{\mathcal{O}'}$ of **requestable values** *for $\mathcal{O}'$ is defined as* $\mathcal{R}_v^{\mathcal{O}'} = \{\mathsf{f}(o)[v] \mid o \in \mathcal{O}'\} \cup \{\mathsf{b}(o)[v], \mathsf{e}(o)[v] \mid o \in \mathcal{O}' \text{ and } o \text{ is non-unary}\} - \{\mathsf{u}\}$.

Obviously, $\mathcal{R}_v^{\mathcal{O}} \subseteq \mathcal{D}_v$ for all $v \in \mathcal{V}$. For each state variable domain, we further define the graph of possible transitions for this domain, without taking the other domains into account, and the reachability graph for arbitrary subsets of the domain.

Definition 3.3 *For each $v \in \mathcal{V}$, we define the corresponding* **domain transition graph** *G_v as a directed labelled graph $G_v = \langle \mathcal{D}_v^+, \mathcal{T}_v \rangle$ with vertex set $\mathcal{D}_v^+$ and arc set $\mathcal{T}_v$ s.t. for all $x, y \in \mathcal{D}_v^+$ and $o \in \mathcal{O}$, $\langle x, o, y \rangle \in \mathcal{T}_v$ iff $\mathsf{b}(o)[v] = x$ and $\mathsf{e}(o)[v] = y \neq \mathsf{u}$. Further, for each $X \subseteq \mathcal{D}_v^+$ we define the* **reachability graph** *for X as a directed graph $G_v^X = \langle X, \mathcal{T}_X \rangle$ with vertex set X and arc set $\mathcal{T}_X$ s.t. for all $x, y \in X$ $\langle x, y \rangle \in \mathcal{T}_X$ iff there is a path from x to y in G_v.*

Alternatively, G_v^X can be viewed as the restriction to $X \subseteq \mathcal{D}_v^+$ of the transitive closure of G_v, but with unlabelled arcs. When speaking about a path in a domain-transition graph below, we will typically mean the sequence of labels, *ie.* operators, along this path. We say that a path in G_v is *via* a set $X \subseteq \mathcal{D}_v^+$ iff each member of X is visited along the path, possibly as the initial or final vertex.

Definition 3.4 *An operator $o \in \mathcal{O}$ is* **irreplaceable** *wrt. a variable $v \in \mathcal{V}$ iff removing an arc labelled with o in G_v splits some component of G_v into two components.*

In the remainder of this paper we will only be interested in SAS$^+$ instances satisfying combinations of the following restrictions.

Definition 3.5 *The SAS$^+$ instance Π is:*

(P) Post-unique *iff for all $o, o' \in \mathcal{O}$, if $\mathsf{e}(o)[v] = \mathsf{e}(o')[v] \neq \mathsf{u}$ for some $v \in \mathcal{V}$, then $o = o'$;*

(U) Unary *iff for all $o \in \mathcal{O}$, o is unary;*

(B) Binary *iff $|\mathcal{D}_v| = 2$ for all $v \in \mathcal{V}$,*

(S) Single-valued *iff there exists some state $s \in \mathcal{S}^+$ s.t. $\mathsf{f}(o) \sqsubseteq s$ for all $o \in \mathcal{O}$.*

(I) Interference-safe *iff every operator $o \in \mathcal{O}$ is either unary or irreplaceable wrt. every $v \in \mathcal{V}$ it affects.*

(A) Acyclic wrt. $\mathcal{R}^{\mathcal{O}}$ *iff $G_v^{\mathcal{R}_v^{\mathcal{O}}}$ is acyclic for each $v \in \mathcal{V}$.*

(A$^+$) Acyclic *iff G_v is acyclic for each $v \in \mathcal{V}$.*

(O) Prevail-order-preserving *iff for each* $v \in \mathcal{V}$*, whenever there are two* $x, y \in \mathcal{D}_v^+$ *s.t.* G_v *has a shortest path* $\langle o_1, \ldots, o_m \rangle$ *from* x *to* y *via some set* $X \subseteq \mathcal{R}_v^{\mathcal{O}}$ *and it has any path* $\langle o'_1, \ldots, o'_n \rangle$ *from* x *to* y *via some set* $Y \subseteq \mathcal{R}_v^{\mathcal{O}}$ *s.t.* $X \subseteq Y$*, there exists some subsequence* $\langle \ldots, o'_{i_1}, \ldots, o'_{i_m}, \ldots \rangle$ *s.t.* $\mathsf{f}(o_k) \sqsubseteq \mathsf{f}(o'_{i_k})$ *for* $1 \leq k \leq m$.

From now on, we will consider the SAS* formalism instead of the SAS$^+$ formalism. One of the reasons for doing so, is that SAS$^+$ and SAS* are equally expressive formalisms under polynomial reductions. (See Bäckström[3,2] for a discussion of expressiveness equivalences.) Another reason is that all previously described polynomial-time SAS$^+$ planners require that the problem instance satisfy both restriction I and A. This mixes badly with actions having u as precondition. For example, restriction I prevents the existence of a path in G_v from u to any state that is the precondition of a non-unary action and restriction A prevents the existence of two requestable values which are both reachable from u.

Many of the complexity results to be presented in the next section will carry over by inheritance, using the following subproblem relationships. (The proofs of theorems are omitted or only sketched. All proofs can be found in Jonsson and Bäckström.[12])

Lemma 3.6 *The following subproblem relations hold:*

1. *SAS*-$A^+ \subseteq$ SAS*-A*
2. *SAS*-U $\subseteq$ SAS*-I*
3. *SAS*-US $\subseteq$ SAS*-A*
4. *SAS*-P $\subseteq$ SAS*-O*
5. *SAS*-PA $\subseteq$ SAS*-I*

Proof sketch: 1 and 2 are trivial. 3 follows from the fact that S and U in combination implies $|\mathcal{R}_v^{\mathcal{O}}| \leq 1$ for all $v \in \mathcal{V}$. Both 4 and 5 follow from analysing how post-uniqueness restricts the domain-transition graphs. □

4 Complexity of plan generation

In this paper we will only discuss the plan generation problem (finding a solution). We will not consider the plan existence problem (deciding whether a solution exists), since we are ultimately interested in actually generating a solution.

Definition 4.1 *Given a SAS* instance* Π*, we have the following planning problems: The* **plan generation problem (PG)** *finds a solution for* Π *or answer that no solution exists. The* **bounded plan generation problem (BPG)** *takes an integer* $k \geq 0$ *as an additional parameter and finds a solution for* Π *of length* k *or shorter or answer that no plan of length* k *or shorter exists for* Π.

The complexity of the plan generation problems follows from the theorems in this section, Lemma 3.6 and inheritance. The tractability results appear in previous publications, as indicated below.

Theorem 4.2 BPG *is polynomial for SAS*-IAO.*[13]

Theorem 4.3 PG *is polynomial for SAS*-US.*[6]

For the intractability results we have to distinguish those problems that are inherently intractable, *ie.* can be proven to take exponential time, and those which are NP-equivalent, *ie.* intractable unless $P = NP$. Observe that we cannot use the term NP-complete since we consider the search problem (generating a solution) and not the decision problem (whether a solution exists). A search problem is NP-easy if it can be Turing reduced to some NP-complete problem, NP-hard if some NP-complete problem can be Turing reduced to it and NP-equivalent if it is both NP-easy and NP-hard. Loosely speaking, NP-equivalence is to search problems what NP-completeness is to decision problems. See Johnson[11] for formal details.

Theorem 4.4 *Optimal solutions are always polynomially bounded for (1) SAS*-A and (2) SAS*-IS.*

Proof sketch: We observe that for SAS*-A each requestable value is visited at most once in an optimal solution and, hence, (1) holds. For SAS*-IS we observe that cycles in a domain-transition graph can only contain unary operators. Hence, the proof of (2) is a simple extension to the proof that optimal SAS$^+$-US plans are of polynomial size.[6] □

Corollary 4.5 BPG *is NP-easy for SAS*-A and SAS*-IS.*

Theorem 4.6 PG *is NP-hard for SAS*-BSA$^+$O.*

Proof sketch: Proof by reduction from EXACT COVER BY 3-SETS. □

Theorem 4.7 BPG *is NP-hard for SAS*-UBSA$^+$.*

Proof sketch: Proof by reduction from MINIMUM COVER. □

Theorem 4.8 PG *is NP-hard for SAS*-BSIA$^+$ and SAS*-UBA$^+$.*

Proof sketch: Proof by reduction from 3-SATISFIABILITY. □

Theorem 4.9 *Both SAS*-PUB and SAS*-PBS have instances with exponentially sized optimal solutions[6] (and are thus inherently intractable).*

The complexity results are summarized in the lattice in Figure 1 which can be viewed as a three-dimensional cube. The figure is to be interpreted in the following way: The top-element of each diamond-shaped sublattice corresponds to a combination of restrictions on the SAS* problem defined by selecting at most one restriction from each of the sets $\{A^+, A\}$, $\{P, O\}$ and $\{U, I\}$. These restrictions are marked along the three axes in the figure, where "-" denotes that neither of the two restrictions on an axis applies. The other three points in each sublattice further specialize the top element by adding one or both of the restrictions B and S, as shown in the enlarged sublattice. As an example of how to interpret the lattice, the SAS*-SAO problem is indicated explicitly in the figure.

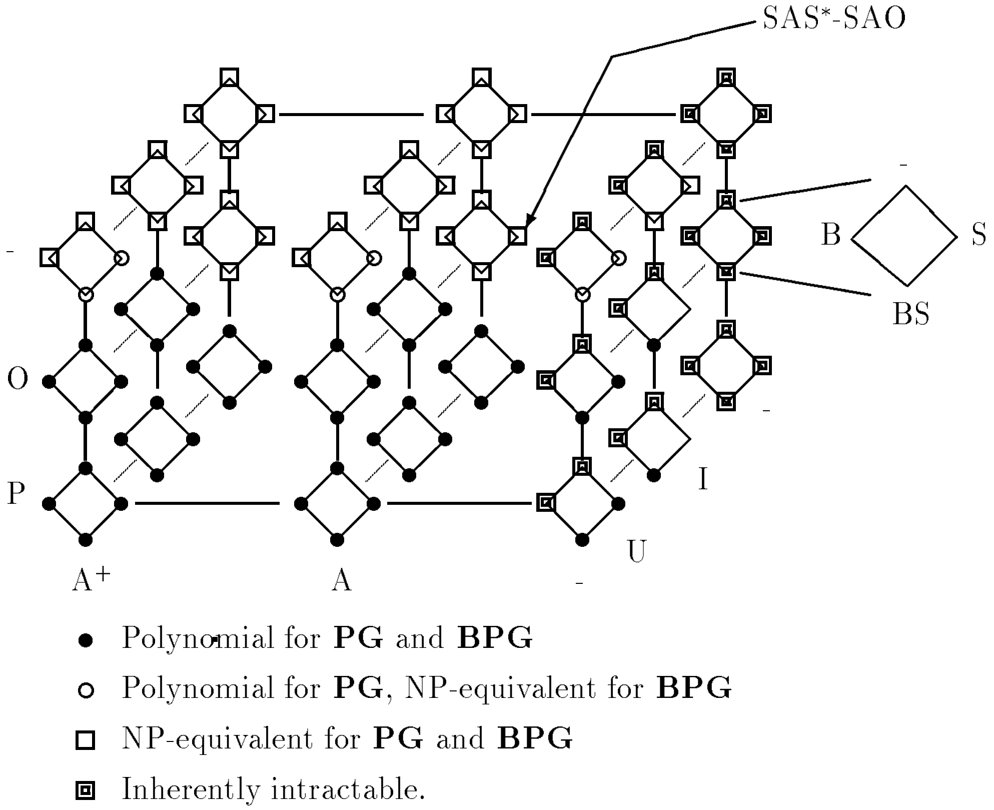

Figure 1: Complexity of SAS* plan generation

The lattice presents complexity results for both bounded and unbounded plan generation as follow. Problems which are polynomial for both **PG** and **BPG** are marked with a filled dot, while those which are polynomial for **PG** but NP-equivalent for **BPG** are marked by an unfilled dot. The problems marked by an unfilled square are NP-equivalent for both **PG** and **BPG** and those marked with a filled square are inherently intractable for both **PG** and **BPG**. Unmarked positions denote problems whose complexity is unknown at present.

5 Discussion and conclusions

Not much research on structural restrictions on planning problems seems to be reported in the literature, although some exceptions can be found. Korf[14] has defined some subgoal dependency properties of planning problems modelled by state-variables, for instance serializability of subgoals. However, this property is PSPACE-complete to test,[8] but does not guarantee tractable planning. Mädler[15] extends Sacerdoti's[16] essentially syntactic state abstraction technique to *structural abstraction*, identifying bottle-neck states (*needle's eyes*) in the state-transition graph for a state-variable formalism. Smith and Peot[17] use an *operator graph* for preprocessing planning problem instances, identifying potential threats that can be safely postponed during planning—thus, pruning the search tree. The operator graph can be viewed as an abstraction of the full state-transition graph, containing all the information relevant to analysing threats. In addition to this, we have previously presented a polynomial-time algorithm for the SAS$^+$-IAO problem[13] and proven that this problem properly generalizes the previously studied syntactically defined tractable planning problems.

In this paper, we have extended this result by analysing the complexity of plan generation under all combinations of restrictions, considering both these new structural restrictions and the previously analysed syntactical ones. By the results in this paper, we can conclude that SAS*-IAO is maximally tractable under the structural restrictions I, A and O, thus being a structural counterpart of the result in Bäckström & Nebel[6] stating that SAS$^+$-PUS is the maximal tractable problem under the syntactical restrictions P, U, B and S. Obviously, this result holds for the SAS$^+$ case as well. Mixing syntactical and structural restrictions yield two problems, SAS*-SIO and SAS*-PSI, whose complexity we have not managed to show. Both problems seems to be of minor practical interest since the S restriction is very severe. Though, it is worth noticing that the SAS*-SIO class is incomparable wrt. expressive power with both the SAS*-PUS and the SAS*-IAO class. We have also shown SAS*-US (and consequently SAS$^+$-US) non-optimal plan generation cannot be further generalized with preserved tractability by replacing US with any combination of our studied syntactical or structural restrictions. By providing some additional hardness results, we have built a map over the complexity of planning for all combinations of both the syntactical and structural restrictions, considering both optimal and non-optimal plan generation.

6 Acknowledgements

This research was sponsored by *the Swedish Research Council for the Engineering Sciences (TFR)* under grants Dnr. 92-143 and Dnr. 93-00291.

References

[1] *Proc. 10th (US) Nat'l Conf. on Artif Intell. (AAAI-92)*, San José, CA, USA, July 1992.

[2] C. Bäckström. *Computational Complexity of Reasoning about Plans.* Doctoral dissertation, Linköping University, Linköping, Sweden, June 1992.

[3] C. Bäckström. Equivalence and tractability results for SAS$^+$ planning. In *Proc. 3rd Int'l Conf. on Principles of Knowledge Repr. and Reasoning (KR-92)*, pages 126–137, Cambridge, MA, USA, Oct. 1992.

[4] C. Bäckström and I. Klein. Parallel non-binary planning in polynomial time. In IJCAI,[10] pages 268–273.

[5] C. Bäckström and I. Klein. Planning in polynomial time: The SAS-PUBS class. *Comput. Intell.*, **7**(3):181–197, Aug. 1991.

[6] C. Bäckström and B. Nebel. Complexity results for SAS$^+$ planning. In *Proc 13th Int'l Joint Conf. on Artif. Intell. (IJCAI-93)*, Chambery, France, Aug.–Sept. 1991.

[7] T. Bylander. Complexity results for planning. In IJCAI,[10] pages 274–279.

[8] T. Bylander. Complexity results for serial decomposability. In AAAI,[1] pages 729–734.

[9] K. Erol, D. S. Nau, and V. S. Subrahmanian. On the complexity of domain-independent planning. In AAAI,[1] pages 381–386.

[10] *Proc 12th Int'l Joint Conf. on Artif. Intell. (IJCAI-91)*, Sydney, Australia, Aug. 1991.

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

[12] P. Jonsson and C. Bäckström. Complexity results for state-variable planning under mixed syntactical and structural restrictions. Research report, Department of Computer and Information Science, Linköping University, 1994. In preparation.

[13] P. Jonsson and C. Bäckström. Tractable planning with state variables by exploiting structural restrictions. In *Proc. 12th (US) Nat'l Conf. on Artif. Intell. (AAAI-94)*, Seattle, WA, USA, July–Aug. 1994. To appear.

[14] R. E. Korf. Planning as search: A quantitative approach. *Artif. Intell.*, **33**:65–88, 1987.

[15] F. Mädler. Towards structural abstraction. In *Proc. 1st Int'l Conf. on Artif. Intell. Planning Sys. (AIPS-92)*, pages 163–171, College Park, MD, USA, June 1992.

[16] E. D. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artif. Intell.*, **5**(2):115–135, 1974.

[17] D. E. Smith and M. A. Peot. Postponing threats in partial-order planning. In *Proc. 11th (US) Nat'l Conf. on Artif Intell. (AAAI-93)*, pages 500–506, Washington, DC, USA, July 1993.