On the Complexity of Possible Truth*

Dana S. Nau
Department of Computer Science Department
and Institute for Systems Research
University of Maryland
College Park, Maryland 20742
nau@cs.umd.edu

Introduction

Chapman's paper, "Planning for Conjunctive Goals," (Chapman, 1987) has been widely acknowledged as a major step towards understanding the nature of nonlinear planning, and it has been one of the bases of later work by others (Yang and Tenenberg, 1990; Kambhampati, 1991; Ginsberg, 1990; Erol *et al.*, 1992a; Erol *et al.*, 1992b). But as with much pioneering work, it is not free of problems—and this has led to much confusion about the meaning of his results. Erol *et al.* (1992a; 1992b) dealt with some of these problems, and the current paper discusses another one.

Chapman (1987, p. 340) states the modal truth criterion as follows:

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

On the same page, Chapman says that this can be interpreted as a polynomial-time method for determining the modal truth of a proposition:

> The criterion can be interpreted procedurally in the obvious way. It runs in time polynomial in the number of steps: the body of the criterion can be verified for each of the n^3 triples $\langle t, C, W\rangle$ with a fixed set of calls on the polynomial-time constraint-mantenance module.

These statements have led others to incorrect conclusions about how difficult it is to compute various modal properties of a plan. For example, Kambhampati (1991, p. 685) initially thought that "Using this truth criterion, we can then develop similar polynomial time EBG algorithms for possible correctness [of a plan]." However, after examining the problem in more detail, he found that the modal truth criterion provided necessary but insufficient conditions to guarantee that a plan is possibly correct (Kambhampati and Kedar, 1992, p. 21).

In this paper I show that given a plan P and a proposition p, it is NP-hard to determine whether or not there exists a completion of P that can be executed to produce a situation in which p is true. I also discuss the conflict between this result and Chapman's statements above, and how this affects the way we should interpret the term "possible truth."

Definitions

In this section, I formalize Chapman's definitions, and correct some problems with them.

Propositions

A *proposition* is either of the following:[1]

1. A list $(p_1\ p_2\ \ldots\ p_n)$, where each p_i is either a variable or a constant. In this case, the proposition is *nonnegated*).
2. An expression of the form $\sim p$, where p is a nonnegated proposition. In this case, the proposition is *negated*.

In both cases, the *content* of the proposition is the list $(p_1\ p_2\ \ldots\ p_n)$. The *negation* of the nonnegated proposition $(p_1\ p_2\ \ldots\ p_n)$ is the negated proposition $\sim(p_1\ p_2\ \ldots\ p_n)$, and vice versa.

Codesignation

If X is a set of variables and constants, then a *codesignation relation* on X is an equivalence relation $\approx$ on X such that each equivalence class contains exactly one constant. If $x \approx y$, then x *codesignates* with y.

*This work was supported in part by NSF Grants IRI-8907890 and NSFD CDR-88003012.

[1] Note that this is different from the conventional definition of a proposition as a 0-ary predicate. What I am calling a proposition would more commonly be called a literal, but the term "proposition" is necessary in order to maintain consistency with Chapman's usage.

Whenever we have a set of propositions s and a codesignation relation $\approx$ on the variables and constants in those propositions, we will extend $\approx$ so that it also applies to the propositions themselves, in the following manner: Let p and q be any two propositions in s, with contents $(p_1\ p_2\ \ldots\ p_m)$ and $(q_1\ q_2\ \ldots\ q_n)$, respectively. Then $p \approx q$ if $m = n$, $p_i \approx q_i$ for every i, and either p and q are both nonnegated, or both negated. A proposition p is *true* (or *false*) in s if it codesignates with a proposition (or the negation of a proposition) in s. s and $\approx$ are *compatible* if no proposition is both true in s and false in s.

Let X be a set of variables and constants. A *codesignation constraint* on X is a syntactic expression of the form '$x \approx y$' or '$x \not\approx y$', where $x, y \in X$. Let D be a set of codesignation constraints on X, and $\equiv$ be a codesignation relation on s. Then $\equiv$ *satisfies* D if $x \equiv y$ for every syntactic expression '$x \approx y$' in D, and $x \not\equiv y$ for every syntactic expression '$x \not\approx y$' in D.

Steps and Ordering

A *step* is a pair $a = (\text{pre}(a), \text{post}(a))$, where $\text{pre}(a)$ and $\text{post}(a)$ are collections of propositions called the *preconditions* and *postconditions* of a. Let s be a set of propositions and $\approx$ be a codesignation relation compatible with both s and $\text{post}(a)$, and suppose that every proposition $p \in \text{pre}(a)$ is true in s. Then a is *executable* in the *input state* s, *resulting* in the *output state* $a(s) = (s - s') \cup \text{post}(a)$, where s' is the set of all propositions in s that are false in $\text{post}(a)$.

Let A be a set of steps. An *ordering constraint* on A is a syntactic expression of the form '$a \prec b$' (read as "*a precedes b*"), where $a, b \in A$. Let O be a set of ordering constraints on A, and $\ll$ be a total ordering on A. Then $\ll$ *satisfies* O if for every syntactic expression '$a \prec b$' in O, $a \ll b$.

Plans

A *plan* is a 4-tuple $P = (s_0, A, D, O)$ satisfying the following properties:

1. s_0 is a set of ground propositions called P's *initial state*;[2]
2. A is a set of steps, such that no two steps have any variables in common, and no step has any variables in common with s_0;
3. D is a set of codesignation constrants on the variables and constants in s_0 and A;
4. O is a set of ordering constraints on the steps of A.

P is *complete* if the following properties hold:

1. There is a unique total ordering $\prec$ over A that satisfies O.
2. There is a unique codesignation relation $\approx$ over P's variables and constants that satisfies D.

If P is complete then its *final state* is s_n, and each proposition p in $\text{post}(a_i)$ is *asserted* in s_i by a_i.

A plan $P' = (s'_0, A', D', O')$ is a *constrainment* of a plan $P = (s_0, A, D, O)$ if $s'_0 = s_0$, $A' = A$, $O \subseteq O'$, and $D \subseteq D'$. P' is a *proper constrainment* of P if P' is a constrainment of P and there is a codesignation relation that satisfies D but not D', or a total ordering that satisfies O but not O'. P' is a *completion* of P if P' is a constrainment of P and P' is complete.[3] P is *consistent* if it has at least one completion; otherwise P is *inconsistent*.

A *planning problem* is a pair $R = (I, F)$, where I and F are sets of propositions that are called the *initial* and *final* states of R, respectively. A plan for R is a plan $P = (s_0, A, D, O)$ such that every proposition in s_0 is true in I. A plan P *solves* R (or alternatively, P is a *solution* for R) if for every completion Q of P, every proposition in F is true in Q's final state.

Situations and Modal Truth

In the definitions above, a plan's initial state is identical to what Chapman calls its initial situation. Chapman also defines several other kinds of situations for plans (Chapman, 1987, p. 338):

> A plan has an *initial situation*, which is a set of propositions describing the world at the time that the plan is to be executed, and a *final situation*, which describes the state of the world after the whole plan has been executed. Associated with each step in a plan its *input situation*, which is the set of propositions that are true in the world just before it is executed, and its *output situation*, which is the set of propositions that are true in the world just after it is executed. In a complete pan, the input situation of each step is the same as the output situation of the previous step. The final situation of a complete plan has the same set of propositions in it as the output situation of the last step.

At first glance, this approach seems quite attractive, because it gives him a convenient way to make modal statements about the situations in a plan, using the following general-purpose definition of modal truth (Chapman, 1987, p. 336):

[2]Here, I depart in two ways from Chapman's definition. First, Chapman calls s_0 the *initial situation* of P, and he also associates a number of other situations with P. However, as discussed in next section, there are a number of problems with Chapman's definition of a situation, so I am avoiding that term completely.

Second, Chapman does not require the initial situation to be ground. However, unless it is ground, it can be shown that TWEAK actually modifies the meaning of its initial state as the planning proceeds.

[3]In Chapman's definition of a completion, it is unclear whether a completion of P should include only the steps in P, or allow other steps to be added. However, various other statements in his paper make it clear that he means for a completion to include only the steps in P, so this is how I (and others (Kambhampati, 1991)) have defined it.

I will say "*necessarily p*" if p is true of all completions of an incomplete plan, and "*possibly p*" if p is true of some completion.

However, this approach leads to several difficulties:

1. In the above passage, apparently p can be any of a number of statements about a plan (e.g., the statement (Chapman, 1987, p. 341) that a plan "necessarily solves the problem"). Unless we place some restrictions on the nature of p, this leads to some dubious results. For example, if P is an incomplete plan, then all completions of P are complete, and therefore P itself is necessarily complete.
2. As pointed out by Yang and Tenenberg (1990), if a plan is incomplete, then its situations are ill-defined. For example, in defining an output situation, Chapman refers to what is actually (not modally) true after executing a step—but if a plan has more than one completion, this will vary depending on which completion we choose.

In order to avoid these problems in the technical material that follows, I will not refer to situations and modal truth at all. Instead of making statements about situations in an incomplete plan, I will instead make the corresponding statements about the states that occur in the completions of that plan; and instead of making statements about modal truth in an incomplete plan, I will instead make the corresponding non-modal statements about the completions of that plan.

Results

Theorem 1 *It is NP-hard to determine, given a proposition p and a plan P, whether there is a completion of P that can be executed to produce a situation in which p is true.*

Proof. The proof is by reduction from 3SAT. In particular, let $X = c_1 c_2 \ldots c_m$ be a CNF formula over the Boolean variables $x_1, x_2, \ldots, x_n$, with three literals in each disjunctive clause c_i. We construct a plan $Q^* = (s_0, A, D, O)$ and a proposition (sat yes yes ... yes), such that there exists a completion of Q^* that can be executed to produce (sat yes yes ... yes) iff X is satisfiable. As illustrated in Fig. 1, Q^* is the following plan:

Initial state. Q^*'s initial state s_0 is the empty set.

Steps. For each Boolean variable x_i, P^* contains two steps, S_i and U_i. S_i has no preconditions, and four postconditions:

$$\sim(\mathsf{xval}_i\ \mathsf{false\ yes}),\quad (\mathsf{xval}_i\ \mathsf{false\ no}),$$
$$\sim(\mathsf{xval}_i\ \mathsf{true\ no}),\quad (\mathsf{xval}_i\ \mathsf{true\ yes}).$$

U_i has no preconditions, and four postconditions:

$$(\mathsf{xval}_i\ \mathsf{false\ yes}),\quad \sim(\mathsf{xval}_i\ \mathsf{false\ no}),$$
$$(\mathsf{xval}_i\ \mathsf{true\ no}),\quad \sim(\mathsf{xval}_i\ \mathsf{true\ yes}).$$

Here, true, false, yes, and no are constants. The interpretations of (xval_i true yes), (xval_i false no), (xval_i false yes), and (xval_i true no) are that the Boolean variable x_i is true, not false, false, and not true, respectively. Thus, the interpretations of S_i and U_i are that they make the Boolean variable x_i true and false, respectively.

Q^* contains a step V, which has no preconditions and no postconditions. The only purpose of V is to provide a separator between the steps S_i and U_i defined above, and the steps L_{ij} defined below.[4]

For each c_i, there let l_{i1}, l_{i2}, l_{i3} be the literals in c_i; i.e., $c_i = l_{i1} + l_{i2} + l_{i3}$. Corresponding to these literals, there are three steps L_{i1}, L_{i2}, L_{i3}, as follows. Each literal l_{ij} is either x_k or $\bar{x}_k$ for some x_k. If $l_{ij} = x_k$, then L_{ij}'s precondition is (xval_k true v_{ij}), where v_{ij} is a variable; if $l_{ij} = \bar{x}_k$, then L_{ij}'s precondition is (xval_k false v_{ij}). L_{ij} has exactly one postcondition, (csat_i v_{ij}).

If $v_{ij} \approx$ yes, then the interpretation of (csat_i v_{ij}) is that c_i is satisfied. Otherwise, (csat_i v_{ij}) has no particular interpretation. Thus, the interpretation of L_{ij} is that if l_{ij} satisfies c_i, then Lij asserts that c_i is satisfied.

Q^* contains a step W whose preconditions are (csat_1 v_1), (csat_2 v_2), ..., (csat_m v_m), where $v_1, v_2, \ldots, v_m$ are variables. W has one postcondition: (sat v_1 v_2 ... v_m). If $v_1 \approx v_2 \approx \ldots \approx v_m \approx$ yes, then the interpretation of (sat v_1 v_2 ... v_m) is that X is satisfied. Otherwise, (sat v_1 v_2 ... v_m) has no particular interpretation. Thus, the interpretation of W is that if every clause c_i of X is satisfied, then W asserts that X is satisfied.

Constraints. O contains an ordering constraint '$S_i \prec V$' for every S_i, an ordering constraint '$U_i \prec V$' for every U_i, and ordering constraints '$L_{ij} \prec V$' and '$L_{ij} \prec W$' for every L_{ij}. There are no other ordering constraints. There are no codesignation constraints; i.e., $D = \emptyset$.

Then in every completion of Q^*, V's input and output states will both be the set of propositions

$$s = s_1 \cup s_2 \cup \ldots \cup s_n,$$

where each s_k corresponds to an assignment of a truth value to x_k. It follows that each step L_{ij} will assert (csat_i yes) iff the truth value assigned to the corresponding Boolean variable x_k satisfies l_{ij}. Otherwise, L_{ij} will assert (csat_i no).

Since any ordering of the S_i and U_i is possible, every assignment of truth values to the x_k is represented in at least one completion of Q^*. Thus if X is satisfiable,

[4] It is easy to add preconditions and postconditions to V, and preconditions to the L_{ij}, in such a way that a partial-order planner such as TWEAK would construct Q^*. However, just as Chapman did at various points in his paper, I have omitted these preconditions and postconditions to keep the presentation simple.

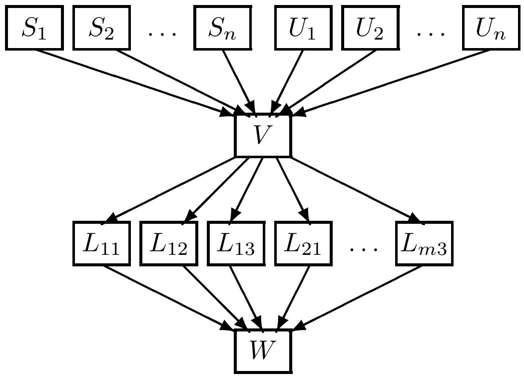

Figure 1: The plan Q^*.

then there is a completion of Q^* such that for every c_i, at least one of l_{i1}, l_{i2}, l_{i3} is satisfied, whence L_{ij} will assert (csat_i yes).

Thus, there is a completion of Q^* such that in W's input state, (csat_i v_{ij}) is true for every i, so that W will assert (sat yes yes ... yes).

If X is not satisfiable, then for every completion of Q^*, there will be at least one i such that none of l_{i1}, l_{i2}, l_{i3} is satisfied. Thus (csat_i no) will be true in W's input state, but (csat_i yes) will not. Thus in every executable completion of Q^*, the proposition (sat v_1 v_2 ... v_m) asserted by W will contain at least one $v_i \approx$ no. ■

Discussion and Conclusions

Because of the wide impact of Chapman's paper, it is important to correct any misimpressions that may result from it—and there appears to be a problem with the notion of modal truth. In particular, Theorem 1 seems to be in direct contradiction to Chapman's statement that modal truth can be computed in polynomial time.

In discussing Theorem 1 with me, Subbarao Kambhampati has expressed a different point of view: that the modal truth criterion is a "local truth criterion," in which we say that a proposition p is possibly true in a plan P if there is a completion P' of P in which some action asserts p, regardless of whether or not it will actually be possible to execute P' to produce p. According to this interpretation, the possible truth of a proposition is computable in polynomial time as Chapman states.

However, I see several difficulties with this interpretation. First, it appears to be a nonstandard interpretation of what "possible truth" means (for example, see Ginsberg (1990)). Second, it will not solve the problem that Kambhampati had wanted to solve, of finding the "weakest conditions under which at least some topological sort of the plan can possibly execute" (Kambhampati, 1991, p. 685). Finally—and most seriously—it leads to nonsensical conclusions. For example, it sometimes would lead us to say that a proposition p is possibly true in a plan P, even if it is impossible to execute P in such a way as to produce p.

To see this, consider the plan Q^* developed in the proof of Theorem 1, and suppose that the formula X is unsatisfiable. Then as proved in Theorem 1, there is no completion of Q^* that can ever be executed in such a way as to make (sat yes yes ... yes) true. However, we can produce a completion of Q^* in which W's postcondition (sat v_1 v_2 ... v_m) is constrained to codesignate with (sat yes yes ... yes). This completion is not executable, because W's preconditions cannot be satisfied—but if we interpret possible truth as a "local truth criterion" as described above, then we would ignore the fact that this completion cannot be executed, and say that (sat yes yes ... yes) is possibly true in W's output situation.

Thus, I would argue that the only reasonable alternative is to say that the question "is p possibly true in P's final situation?" is equivalent to the question "is there a completion of P that can be executed to produce p?" From this, it follows from Theorem 1 that unless P=NP, possible truth cannot be computed in polynomial time.

Acknowledgement

I wish to thank Kutluhan Erol, Jim Hendler, Subbarao Kambhampati, and V. S. Subrahmanian for their helpful criticisms and comments.

References

Chapman, David 1987. Planning for conjunctive goals. *Artificial Intelligence* 32:333–379.

Erol, K.; Nau, D.; and Subrahmanian, V. S. 1992a. Complexity, decidability and undecidability results for domain-independent planning. Submitted for publication.

Erol, K.; Nau, D.; and Subrahmanian, V. S. 1992b. When is planning decidable? In *Proc. First Internat. Conf. AI Planning Systems.* 222–227.

Ginsberg, M. L. 1990. What is a modal truth criterion? Unpublished manuscript.

Kambhampati, S. and Kedar, S. 1992. A unified framework for explanation-based generalization of partially ordered and partially instantiated plans. Technical Report TR-92-008, Department of Computer Science and Engineering, Arizona State University.

Kambhampati, Subbarao 1991. Explanation-based generalization of partially ordered plans. In *AAAI-91.* 679–685.

Yang, Q. and Tenenberg, J. D. 1990. Abtweak: Abstracting a nonlinear, least commitment planner. In *AAAI-90.* 204–209.