On the Nature of Modal Truth in Plans*

Dana S. Nau

Department of Computer Science,
Institute for Systems Research, and
Institute for Advanced Computer Studies

University of Maryland
College Park, Maryland 20742

nau@cs.umd.edu

Abstract

Chapman's paper, "Planning for Conjunctive Goals," 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—but it is not free of problems. This paper discusses the following problems with modal truth and the modal truth criterion.

1. It is NP-hard to tell, given a plan P and a ground atom p, whether P is possibly true in P's final situation. This is true despite the fact that modal truth criterion can be computed in polynomial time.
2. The reason for this discrepancy is that the "possible truth" version of the modal truth criterion is incorrect. It tells whether $\neg p$ is not necessarily true—but this is different from telling whether p is possibly true. Possible truth is not the dual of necessary truth, as Chapman had thought it was.
3. Instead, possible truth is the dual of another problem, which is co-NP-hard: the problem of determining whether p is true over all executable completions of a plan.

Despite the above problems, the "necessary truth" version of the modal truth criterion (and hence the TWEAK planner) are still correct.

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

1 Introduction

Chapman's paper, "Planning for Conjunctive Goals," [1] 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 (for example, [2, 4, 3, 5, 7, 10, 11]). But as with much pioneering work, it is not free of problems, and this has led to confusion about the meaning of his results. Previous papers [2, 3, 6, 11] have pointed out several of these problems, and the current paper discusses some additional ones.

Chapman stated that the modal truth criterion could be used as a polynomial-time method for determining the modal truth of a proposition [1, p. 340]:

> 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-maintenance module.

Since a plan is possibly correct iff its goal conditions are possibly true, this led Kambhampati to believe he could compute the possible correctness of a plan in polynomial time [7, p. 685]:

> The algorithms presented in this paper compute the weakest conditions under which *all* topological sorts of a partially ordered plan can be guaranteed to execute successfully. Sometimes, it may be useful to compute weakest conditions under which at least some topological sort of the plan can possibly execute. Generalization algorithms for this case can be developed in a very similar fashion. In particular, the truth criterion for guaranteeing the possible truth of a proposition is specified by reversing the modalities in the [modal truth criterion]. 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, Kambhampati found that the "possible truth" version of the modal truth criterion gave him conditions that were necessary but insufficient to guarantee the possible correctness of a plan [6, p. 21].

This paper points out the following problems with the modal truth criterion:

1. Although Chapman thought that possible truth and necessary truth were duals of each other [1, p. 368], they are not. More specifically, if p is possibly true, then $\neg p$ is not necessarily true; but the reverse implication does not hold. The "possible truth" version of the modal truth criterion computes the dual of necessary truth, and therefore it is incorrect.

2. Telling whether p is possibly true is the dual of a different problem: the problem of telling whether p is true in every executable completion of a plan. This problem is co-NP-hard, and thus it is NP-hard to tell whether p is possibly true.

Despite these problems, the "necessary truth" version of the modal truth criterion (and thus Chapman's TWEAK planner) are still correct.

This paper is organized as follows. Section 2 contains basic definitions. Section 3 presents the complexity results; the proofs are in the appendix. Section 4 clarifies Chapman's terminology, and Section 5 compares my results with his. Section 6 contains concluding remarks.

2 Definitions

The planning language $\mathcal{L}$ is any function-free first-order language. Since $\mathcal{L}$ is function-free, every term is either a variable symbol or a constant symbol, and thus every ground term is a constant symbol. I follow the usual convention of defining an *atom* to be a predicate symbol followed a list of terms, a *literal* to be an atom or its negation, and a *proposition* to be a 0-ary atom. Thus, what Chapman calls a proposition, I call a literal.

A *state* is any finite collection of ground atoms of $\mathcal{L}$. If a state s contains a ground atom p, then p is true in s and $\neg p$ is false in s; otherwise p is false in s and $\neg p$ is true in s.

If T is a finite set of terms, then a *codesignation constraint* on T is a syntactic expression of the form '$t \approx u$' or '$t \not\approx u$', where $t, u \in T$. Let D be a set of codesignation constraints on T, and θ be a *ground substitution* over T (i.e., a substitution that assigns a ground term to each variable in T). Then θ *satisfies* D if $t\theta = u\theta$ for every syntactic expression '$t \approx u$' in D, and $t\theta \neq u\theta$ for every syntactic expression '$t \not\approx u$' in D. If $t\theta = u\theta$ for every θ that satisfies D, then t *codesignates with* u.

A *step* is a triple $a = (\text{name}(a), \text{pre}(a), \text{post}(a))$, where $\text{name}(a)$ is a constant symbol called the *name* of a, and $\text{pre}(a)$ and $\text{post}(a)$ are collections of literals called a's *preconditions* and *postconditions*.[1] If A is a set of steps, then an *ordering constraint* on A is a syntactic expression of the form '$a \prec b$' (read as "*a precedes b*"), where $a, b \in A$. If O is a set of ordering constraints on A and $\prec$ is a total ordering on A, then $\prec$ *satisfies* O if for every syntactic expression '$a \prec b$' in O, $a \prec b$.

A *plan* is a 4-tuple $P = (s_0, A, D, O)$, where s_0 is a state called P's *initial* state, A is a set of steps, D is a set of codesignation constraints on the terms of P (i.e., the terms in s_0 and A), and O is a set of ordering constraints on the steps of A. P is *complete* if there is a unique total ordering $a_1 \prec a_2 \prec \ldots \prec a_n$ over A that satisfies O, and a unique ground substitution θ over the terms of P that satisfies D.

Suppose that P is complete, and let k be the largest integer $\leq n$ for which there are states $s_1, s_2, \ldots, s_k$ such that the following properties are satisfied for $1 \leq i \leq k$:

1. s_{i-1} satisfies a_i's preconditions; i.e., $p\theta$ is true in s_{i-1} for every literal $p \in \text{pre}(a_i)$.
2. s_i is the state produced by performing the step a_i in the state s_{i-1}; i.e., $s_i = (s_{i-1} - f_i) \cup t_i$, where t_i is the set of all ground atoms $p\theta$ such that $p \in \text{post}(a_i)$, and f_i is the set of all negated ground atoms $p\theta$ such that $p \in \text{post}(a_i)$.

Then for $1 \leq i \leq k$, a_i is *executable* in the *input* state s_{i-1}, *producing* the *output* state s_i. If $k = n$, then P is *executable*, and it *produces* the *final* state s_n.

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 *completion* of P if P' is a constrainment of P and P' is complete.[2] P is *consistent* if it has at least one completion; otherwise P is *inconsistent*.

I now define three decision problems:

[1] Chapman's definition of a step is basically similar to this, but without $\text{name}(a)$. However, as pointed out by McAllester and Rosenblitt [9], unless we give unique names to steps, it is impossible for a plan to contain two distinct steps that have the same preconditions and postconditions.

[2] Chapman's definition of a completion does not make it entirely clear 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 most others (e.g., [7]) use the term.

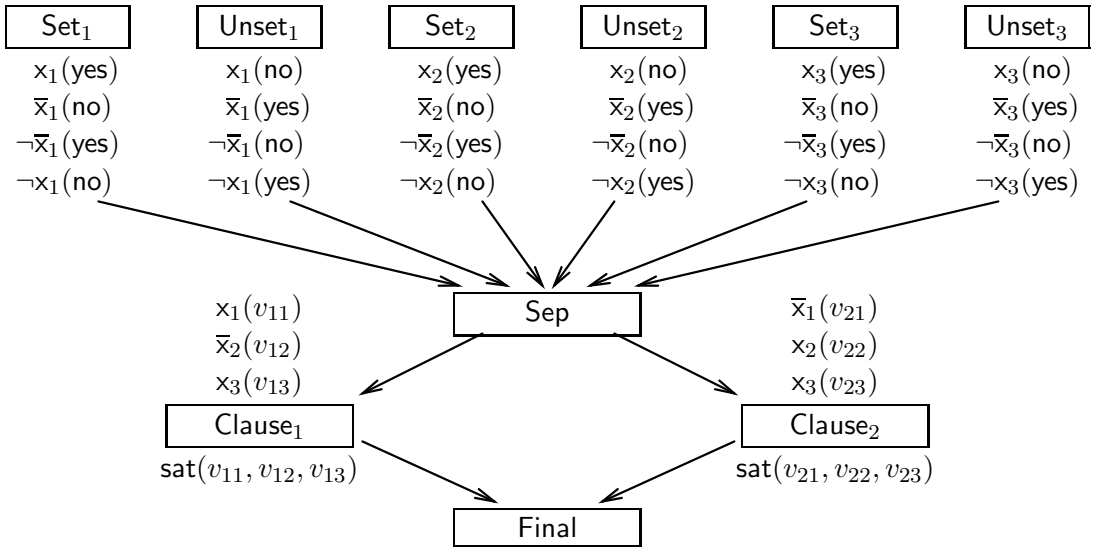

Figure 1: The plan P_X^* in the case where $X = x_1\overline{x_2}x_3 + \overline{x_1}x_2x_3$. Each step's name is in a box, with its preconditions and postconditions above and below the box.

POSSIBLE CORRECTNESS: given a ground atom p and a plan P, is there a completion of P that produces a final state in which p is true?

NECESSARY CORRECTNESS: given a ground atom p and a plan P, does every completion of P produce a final state in which p is true?

CONDITIONAL CORRECTNESS: given a ground atom p and a plan P, does every executable completion of P produce a final state in which p is true?

As discussed later in Section 5, POSSIBLE CORRECTNESS and NECESSARY CORRECTNESS are equivalent to the problems of determining whether p is possibly or necessarily true, respectively. This is the reason for giving them the names they have.

3 Complexity Results

Theorem 1 CONDITIONAL CORRECTNESS *is co-NP-hard.*

The proof is by reduction from the complement of 3SAT (the satisfiability problem with three literals per clause). In particular, let $X = c_1 + c_2 + \ldots + c_m$ be a DNF formula over the Boolean variables $x_1, x_2, \ldots, x_n$, where each c_i is a conjunct of three literals $c_i = l_{i1}l_{i2}l_{i3}$. I encode X as a plan P_X^* and a ground atom $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$, such that every executable completion of P_X^* produces a final situation that contains $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$ iff X is a tautology. Fig. 2 gives an example of P_X^*; the proof is in the appendix.

Theorem 2 POSSIBLE CORRECTNESS *is NP-hard.*

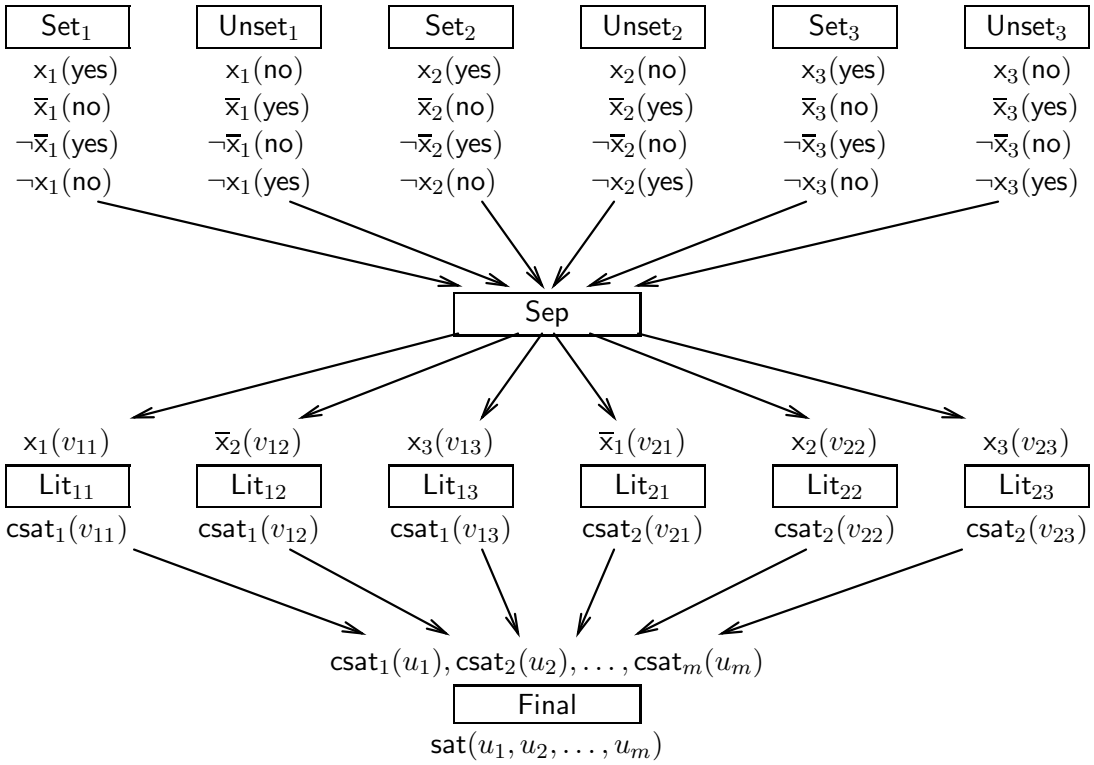

Figure 2: The plan Q_X^* in the case where $X = (x_1 + \overline{x_2} + x_3)(\overline{x_1} + x_2 + x_3)$. Each step's name is in a box, with its preconditions and postconditions above and below the box.

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. I encode X as a plan Q_X^* and a ground atom $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \ldots, \mathsf{yes})$, such that some completion of Q_X^* can be executed to produce $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \ldots, \mathsf{yes})$ iff X is satisfiable. Fig. 2 gives an example of Q_X^*; the proof is in the appendix.

Remark. Note that POSSIBLE CORRECTNESS and CONDITIONAL CORRECTNESS are dual problems: given a ground literal p and a plan P, there is an executable completion of P that produces a final state in which p is true iff no executable completion of P produces a final state in which $\neg p$ is true. For this reason, either of Theorems 1 and 2 could have been proved as a corollary of the other. The reason I did not do so is that I will need to use both of the plans P_X^* and Q_X^* later in this paper.

4 Comparison with Chapman's Terminology

4.1 Situations

Chapman defines a *situation* to be a collection of literals.[3] Given a literal p and a situation s, he defines p to be true if it codesignates with a literal in s, and false if it codesignates with the negation of a literal in s. Chapman also makes the following definitions [1, p. 338]:

> A plan has an *initial situation*, which is a set of [literals] 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 [literals] that are true in the world just before it is executed, and its *output situation*, which is the set of [literals] 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 [literals] in it as the output situation of the last step.

This approach leads to several difficulties:

1. As pointed out by Yang and Tenenberg [11], if a plan P is incomplete, then its situations are ill-defined. For example, suppose P consists of two unordered steps a and b, such that a asserts p and denies q, and b asserts q and denies p. Then P's final situation is either $\{p\}$ or $\{q\}$, depending on which completion of P we choose.

2. If a situation contains literals that are not completely ground, then what those literals mean is problematic. For example, suppose that a plan's initial situation contains the literal $p(x)$, where x is a variable symbol. This literal cannot mean $(\forall x)p(x)$, because Chapman's TWEAK planner may later constrain $x \not\approx y$ for some constant or variable y. It cannot mean $(\exists x)p(x)$, because TWEAK may later constrain $x \approx y$ for some constant or variable y. Apparently, it means $p(x)$ for some undetermined x, and TWEAK gets to choose what x is. In other words, if the initial situation contains any variables, then TWEAK changes the meaning of the initial situation as it goes along.

To handle these problems, I define situations as follows. If P is a plan, then associated with every action a of P are two symbols $\mathsf{in}(a)$ and $\mathsf{out}(a)$, called a's *input* and *output* situations. Associated with P are symbols init and fin called the *initial* and *final* situations of P. All of these symbols must be distinct. Whenever $a \prec b$, I will also say that $x \prec y$, where x may be a or $\mathsf{in}(a)$ or $\mathsf{out}(a)$, and y may be b or $\mathsf{in}(b)$ or $\mathsf{out}(b)$.[4]

I define what is *true* and *false* in a situation as follows. Let P be a complete plan, and p be a ground literal. Then p is true in init if p is true in P's initial state, and p is true in fin if p is true in P's final state. If a is an executable step of P, then p is true in $\mathsf{in}(a_i)$ (or $\mathsf{out}(a_i)$) if p is true in a's input state (or output state, respectively). p is false in a situation

[3] He calls them propositions—but as mentioned at the beginning of Section 2, I call them literals instead.

[4] According to this definition $\mathsf{out}(a)$ and $\mathsf{in}(b)$ are always distinct, hence I would say $\mathsf{out}(a) \prec \mathsf{in}(b)$ in some cases where Chapman would say $\mathsf{out}(a) = \mathsf{in}(b)$. However, this makes no significant difference in any of the results.

s iff $\neg p$ is true in s. Note that the law of the excluded middle does not apply here: if P is not executable, then p is neither true nor false in P's final situation.

As a consequence of the above definitions, it follows that p is true in s iff the following three conditions are satisfied:

Establishment. Either p codesignates with a postcondition of some step a that precedes s, or $p \in s_0$.

Nondeletion. For every step b between a (or s_0) and s, $\neg p$ does not codesignate with a postcondition of b.

Executability. Every step that precedes s is executable.

4.2 Modal Truth

Chapman defines modal truth as follows [1, p. 336]:

> 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.

In this definition, apparently p can be nearly any statement about a plan: examples in his paper include statements about specific literals and situations in the plan (as in the modal truth criterion quoted earlier), and also statements about the entire plan (e.g., the statement [1, p. 341] that a plan "necessarily solves the problem"). However, unless we place some restrictions on the nature of p, this has 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.

Therefore, I will modify Chapman's definition of modal truth as follows. I will say "*necessarily p*" if p is true of all completions of a plan, and "*possibly p*" if p is true of some completion, where p is restricted to be one of the following kinds of statements:

- "a is true in s" or "a is false in s," where a is a ground literal and s is a situation;
- "u precedes v," where u and v are steps or situations.

From this, it follows that if p is an atom, P is a plan, and s is a situation in P, then

1. p is necessarily true in s iff the establishment, nondeletion, and executability conditions hold in every completion of P;
2. p is possibly true in s iff the establishment, nondeletion, and executability conditions hold in at least one completion of P.

A closely related concept is *conditional modal truth*, which is like modal truth except that it does not require executability:

1. p is *conditionally necessarily true* in s iff the establishment and nondeletion conditions hold in every completion of P;
2. p is *conditionally possibly true* in s iff the establishment and nondeletion conditions hold in at least one completion of P.

Others [8] have proposed using this as a definition of modal truth rather than conditional modal truth, but this leads to problems. For example, if some of P's completions are not executable, then sometimes p is conditionally possibly true in P's final situation, even if it is impossible to execute P in such a way as to produce p.

To see this, consider the plan Q^*_X used in the proof of Theorem 2, and suppose that X is unsatisfiable. Then from the proof of Theorem 2, it follows that no completion of Q^*_X can be executed to make $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \ldots, \mathsf{yes})$ true. However, there is a *unexecutable* completion of Q^*_X in which Final's postcondition $\mathsf{sat}(v_1, v_2, \ldots, v_m)$ codesignates with $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \ldots, \mathsf{yes})$, so since conditional modal truth does not require executability, $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \ldots, \mathsf{yes})$ is conditionally possibly true in Q^*_X's final situation.

5 Comparison with Chapman's Results

From the above, it follows that NECESSARY CORRECTNESS and POSSIBLE CORRECTNESS are equivalent to the following problems, respectively:

1. Given a ground atom p and a plan P, is p necessarily true in P's final situation?
2. Given a ground atom p and a plan P, is p possibly true in P's final situation?

As a consequence, Theorem 2 shows that the second problem is NP-hard, conflicting with Chapman's statement (quoted in Section 1) that the modal truth criterion can be computed in polynomial time. To discover the reason for this discrepancy, let us examine the modal truth criterion in more detail. Chapman stated it as follows [1, p. 340]:

> **Modal Truth Criterion.** A [literal] 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 [literal] q possibly codesignating with p which C denies, there is a step Final necessarily between C and s which asserts r, a [literal] 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).

Thus, the criterion for possible truth is as follows:

> A literal p is possibly true in a situation s iff two conditions hold: there is a situation t equal or possibly previous to s in which p is possibly asserted; and for every step C necessarily before s and every literal q necessarily codesignating with p which C denies, there is a step Final possibly between C and s which asserts r, a literal such that r and p codesignate whenever p and q codesignate.

The conditions stated in the criterion are not sufficient for guaranteeing that p is possibly true (this is basically equivalent to Kambhampati's observation [6] that the criterion provides necessary but insufficient conditions for guaranteeing that a plan is possibly correct). To prove this, let the plan $P^\dagger_X$ be the same as the plan P^*_X of Theorem 1, except for the following differences:

- The initial state s_0 is the set $\{\mathsf{unsat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})\}$.
- The step Final has one precondition $\mathsf{sat}(u, v, w)$, and one postcondition $\neg\mathsf{unsat}(u, v, w)$, where u, v, w are variable symbols.

Now, suppose X is a tautology. Then from the proof of Theorem 1, every executable completion of P_X^* will assert $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$. Thus, no executable completion of $P_X^\dagger$ will produce $\mathsf{unsat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$, so $\mathsf{unsat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$ is not possibly true in $P_X^\dagger$'s final situation.

If we apply the criterion for possible truth, we will reach a different conclusion. $\mathsf{unsat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$ is asserted in s_0 and thus in P's initial situation; and P's initial situation precedes its final situation. Thus the first condition of the criterion is satisfied. The only step that can ever deny $\mathsf{unsat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$ is Final. However, Final does not *necessarily* deny $\mathsf{unsat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$, because not every completion of $P_X^\dagger$ is executable (for example, if we constrain Final's precondition $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$ to codesignate with $\mathsf{sat}(\mathsf{foo}, \mathsf{bar}, \mathsf{baz})$, then Final is non-executable). Thus, the criterion concludes, incorrectly, that $\mathsf{unsat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$ is possibly true in $P_X^\dagger$'s final situation.

The reason why this problem was not immediately evident in Chapman's paper was because he did not prove the criterion for possible truth explicitly, but instead said that it followed by "modal duality" [1, p. 368]. However, necessary truth and possible truth are not dual to each other. To see this, let p be a ground atom and P be a plan, and consider the following two statements:

1. p is possibly true in P's final situation;
2. $\neg p$ is not necessarily true in P's final situation.

If necessary truth and possible truth were dual, then these two statements would be equivalent, but they are not. The implication goes in one direction, but not the other:

($\Rightarrow$): Let p be a ground atom and P be a plan, and suppose p is possibly true in P's final situation. Then p is true in the final state of at least one completion of P, so $\neg p$ is false in the final state of this completion of P. Thus, $\neg p$ is not necessarily true in P's final situation.

($\not\Leftarrow$): Suppose $\neg p$ is not necessarily true in P's final situation. There are two ways this can happen: either p is true in the final state of some completion of P, or else $\neg p$ is true in the final state of every executable completion of P but some completion of P is not executable. In the first case, p is possibly true in P's final situation; but in the second case, it is not.

What the "possible truth" version of the modal truth criterion computes is whether $\neg p$ is not necessarily true. This can be computed in polynomial time as Chapman stated it could, but it is not the same as computing whether p is possibly true.

Since the "necessary truth" version of the modal truth criterion refers at several points to the possible truth of various statements, the problems with the "possible truth" version raise the question whether the "necessary truth" version might also be incorrect. However, in the special case where every completion of a plan is executable (which is required for the necessary truth criterion to succeed), p is possibly true iff $\neg p$ is not necessarily true. Thus the "necessary truth" version (and hence Chapman's TWEAK planner) is correct, even though the "possible truth" version is not.

6 Concluding Remarks

The definition of modal truth says that p is possibly true in a plan P if it is true in some completion of P, and p is necessarily true in P if it is true in every completion of P. Chapman had thought that these were dual concepts, but they are not. Instead, telling whether p is possibly true in a plan P is the dual of telling whether p is true in every *executable* completion of P—a problem that is co-NP-hard. This has the following consequences:

1. Unless P=NP, possible truth cannot be computed in polynomial time as Chapman had thought it could. Instead, the problem is NP-hard.

2. The "possible truth" version of the modal truth criterion is wrong. It tells whether $\neg p$ is not necessarily true, but this is different from telling whether p is possibly true.

Because of the wide impact of Chapman's paper, it is important to correct any misimpressions that may result from it. I hope readers will find this paper useful for that purpose.

Acknowledgement

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

References

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

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

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

[4] M. L. Ginsberg. What is a modal truth criterion? Unpublished manuscript, November 1990.

[5] Steven Hanks and Daniel S. Weld. Systematic adaptation for case-based planning. In *Proc. First Internat. Conf. AI Planning Systems*, pages 96–105, June 1992.

[6] S. Kambhampati and S. Kedar. 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, April 1992.

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

[8] Subbarao Kambhampati. Private communication, 1993.

[9] David McAllester and David Rosenblitt. Systematic nonlinear planning. In *AAAI-91*, pages 634–639, July 1991.

[10] M. A. Peot. Conditional nonlinear planning. In *Proc. First International Conference on AI Planning Systems*, pages 189–197, 1992.

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

Appendix

Proof of Theorem 1. Let $X = c_1 + c_2 + \ldots + c_m$ be a DNF formula over the Boolean variables $x_1, x_2, \ldots, x_n$, where each c_i is a conjunct of three literals $c_i = l_{i1}l_{i2}l_{i3}$. P^*_X is the following plan:

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

Steps. For each Boolean variable x_i, there are two steps, Set_i and Unset_i. Set_i has no preconditions, and the following postconditions:

$$\neg\overline{\mathsf{x}}_i(\mathsf{yes}), \overline{\mathsf{x}}_i(\mathsf{no}), \neg\mathsf{x}_i(\mathsf{no}), \mathsf{x}_i(\mathsf{yes}).$$

Unset_i has no preconditions, and the following postconditions:

$$\overline{\mathsf{x}}_i(\mathsf{yes}), \neg\overline{\mathsf{x}}_i(\mathsf{no}), \mathsf{x}_i(\mathsf{no}), \neg\mathsf{x}_i(\mathsf{yes}).$$

In the above, yesand no are constant symbols. The interpretations of $\mathsf{x}_i(\mathsf{yes})$, $\overline{\mathsf{x}}_i(\mathsf{no})$, $\overline{\mathsf{x}}_i(\mathsf{yes})$, and $\mathsf{x}_i(\mathsf{no})$ are that the Boolean variable x_i is true, not false, false, and not true, respectively. Thus, the interpretations of Set_i and Unset_i are that they make the Boolean variable x_i true and false, respectively.

There is a step Sep, which has no preconditions and no postconditions.[5] The only purpose of Sep is to provide a separator between the steps Set_i and Unset_i defined above, and the steps Clause_i defined below.

For each conjunct $c_i = l_{i1}l_{i2}l_{i3}$ in X, there is a step Clause_i. Corresponding to the literals in c_i, Clause_i has preconditions $\mathsf{Lit}_{i1}, \mathsf{Lit}_{i2}, \mathsf{Lit}_{i3}$, as follows. Each l_{ij} is either x_k or $\overline{x_k}$ for some x_k. If $l_{ij} = x_k$, then Lit_{ij} is $\mathsf{x}_k(v_{ij})$, where v_{ij} is a variable; if $l_{ij} = \overline{x_k}$, then Lit_{ij} is $\overline{\mathsf{x}}_k(v_{ij})$. Clause_i has one postcondition: $\mathsf{sat}(v_{i1}, v_{i2}, v_{i3})$.

The interpretation of $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$ is that X is satisfied. For any other constant symbols u, v, w, $\mathsf{sat}(u, v, w)$ has no particular interpretation. Thus, the interpretation of Clause_i is that if $c_i = l_{i1}l_{i2}l_{i3}$ is satisfied, then Clause_i asserts that X is satisfied.

There is one other step, Final, which has no preconditions and no postconditions. Final's purpose is to provide a final step in the plan.

[5] In the proof of his Intractability Theorem, Chapman also uses steps that have no preconditions and postconditions. However, the use of such steps raises the question of whether TWEAK can ever create a plan such as P^*_X. It is easy to modify P^*_X in such a way that TWEAK will construct it; the modification is as follows. For each step x of P^*_X, add a new postcondition $\mathsf{done}(\mathrm{name}(x))$ (recall that $\mathrm{name}(x)$ is a constant symbol). For each ordering constraint '$x \prec y$' of P^*_X, give y a new precondition $\mathsf{done}(\mathrm{name}(x))$.

Constraints. O contains an ordering constraint '$\mathsf{Set}_i \prec \mathsf{Sep}$' for every Set_i, an ordering constraint '$\mathsf{Unset}_i \prec \mathsf{Sep}$' for every Unset_i, and ordering constraints '$\mathsf{Clause}_i \prec \mathsf{Sep}$' and '$\mathsf{Clause}_i \prec \mathsf{Final}$' for every Clause_i. There are no other ordering constraints. There are no codesignation constraints; i.e., $D = \emptyset$.

Let P be any executable completion of P^*_X, and θ be the unique ground substitution that satisfies P's codesignation constraints. In P, Sep's input and output states will both be a set s of ground atoms of the form $\overline{\mathsf{x}}_i(u)$ and $\mathsf{x}_i(v)$, corresponding a truth value for x_i. More specifically,

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

where each s_k is either $\{\overline{\mathsf{x}}_k(\mathsf{yes}), \mathsf{x}_k(\mathsf{no})\}$ (indicating that x_k is false), or $\{\overline{\mathsf{x}}_k(\mathsf{yes}), \mathsf{x}_k(\mathsf{no})\}$ (indicating that x_k is true).

For each Clause_i, Clause_i's input state will consist of some ground atoms of the form $\mathsf{sat}(u, v, w)$, plus the set s described above. Since Clause_i is executable, each precondition Lit_{ij} of Clause_i codesignates with an atom in Clause_i's input state. In particular, since each Lit_{ij} is either $\mathsf{x}_k(v_{ij})$ or $\overline{\mathsf{x}}_k(v_{ij})$ for some k, it follows that $\mathsf{Lit}_{ij}\theta \in s_k$. Thus, either $v_{ij}\theta = \mathsf{yes}$ or $v_{ij}\theta = \mathsf{no}$, depending on whether s_k corresponds to a truth value for x_k that makes l_{ij} true, or one that makes l_{ij} false. Clause_i will assert $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$ iff s corresponds to a set of truth values that make l_{i1}, l_{i2}, and l_{i3} all true.

From the above, we see that P will produce a final state containing $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$ if s corresponds to a set of truth values that makes at least one of the conjuncts $c_i = l_{i1}l_{i2}l_{i3}$ true. Since s may correspond to any assignment of truth values to $x_1, x_2, \ldots, x_n$, this means that P will produce a final state containing $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \mathsf{yes})$ iff $X = c_1 + c_2 + \ldots + c_m$ is true for all assignments of truth values to $x_1, x_2, \ldots, x_n$. ■

Proof of Theorem 2. Let $X = c_1c_2 \ldots c_m$ be a CNF formula over the Boolean variables $x_1, x_2, \ldots, x_n$, with three literals in each disjunctive clause c_i. Q^*_X is the following plan:

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

Steps. For each Boolean variable x_i, Q^*_X contains two steps, Set_i and Unset_i, which are are identical to the corresponding steps of the plan P^*_X of Theorem 1. Also, Q^*_X contains a step Sep, which is identical to the step Sep of P^*_X.

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 $\mathsf{Lit}_{i1}, \mathsf{Lit}_{i2}, \mathsf{Lit}_{i3}$, as follows. Each literal l_{ij} is either x_k or $\overline{x_k}$ for some x_k. If $l_{ij} = x_k$, then Lit_{ij} has the precondition $\mathsf{x}_k(v_{ij})$, where v_{ij} is a variable symbol. If $l_{ij} = \overline{x_k}$, then Lit_{ij} has the precondition $\overline{\mathsf{x}}_k(v_{ij})$ instead. Lit_{ij} has one postcondition: $\mathsf{csat}_i(v_{ij})$.

The interpretation of $\mathsf{csat}_i(\mathsf{yes})$ is that c_i is satisfied. For any other constant symbol v, $\mathsf{csat}_i(v)$ has no particular interpretation. Thus, the interpretation of Lit_{ij} is that if l_{ij} satisfies c_i, then Lij asserts that c_i is satisfied.

Q^*_X contains a step Final whose preconditions are $\mathsf{csat}_1(u_1), \mathsf{csat}_2(u_2), \ldots, \mathsf{csat}_m(u_m)$, where $u_1, u_2, \ldots, u_m$ are variables. Final has one postcondition: $\mathsf{sat}(u_1, u_2, \ldots, u_m)$. The interpretation of $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \ldots, \mathsf{yes})$ is that X is satisfied. For any other constant symbols $u_1, \ldots, u_m$, $\mathsf{sat}(u_1, u_2, \ldots, u_m)$ has no particular interpretation. Thus, the

interpretation of Final is that if every clause c_i of X is satisfied, then Final asserts that X is satisfied.

Constraints. O contains an ordering constraint '$\mathsf{Set}_i \prec \mathsf{Sep}$' for every Set_i, an ordering constraint '$\mathsf{Unset}_i \prec \mathsf{Sep}$' for every Unset_i, and ordering constraints '$\mathsf{Lit}_{ij} \prec \mathsf{Sep}$' and '$\mathsf{Lit}_{ij} \prec \mathsf{Final}$' for every Lit_{ij}. There are no other ordering constraints. There are no codesignation constraints; i.e., $D = \emptyset$.

Let $\mathcal{Q}$ be the set of all completions of Q^*_X such that all steps except possibly Q are executable. Note that every executable completion of Q^*_X is in $\mathcal{Q}$.

In every plan in $\mathcal{Q}$, Sep's input and output states will both be a set s of ground atoms of the form $\overline{\mathsf{x}}_i(u)$ and $\mathsf{x}_i(v)$, corresponding a truth value for x_i. More specifically,

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

where each s_k is either $\{\overline{\mathsf{x}}_k(\mathsf{yes}), \mathsf{x}_k(\mathsf{no})\}$ (indicating that x_k is false), or $\{\overline{\mathsf{x}}_k(\mathsf{yes}), \mathsf{x}_k(\mathsf{no})\}$ (indicating that x_k is true). It follows that each step Lit_{ij} will assert $\mathsf{csat}_i(\mathsf{yes})$ iff the truth value assigned to the corresponding Boolean variable x_k satisfies l_{ij}. Otherwise, Lit_{ij} will assert $\mathsf{csat}_i(\mathsf{no})$.

Suppose X is satisfiable. Since any ordering of the Set_i and Unset_i is possible, every assignment of truth values to the x_k is represented in at least one plan in $\mathcal{Q}$. Thus there is a plan in $\mathcal{Q}$ such that for every c_i, at least one of l_{i1}, l_{i2}, l_{i3} is satisfied, whence Lit_{ij} will assert $\mathsf{csat}_i(\mathsf{yes})$. Thus, there is a plan $Q \in \mathcal{Q}$ such that in Final's input state, $\mathsf{csat}_i(v_{ij})$ is true for every i, so that Final will assert $\mathsf{sat}(\mathsf{yes}, \mathsf{yes}, \ldots, \mathsf{yes})$.

Suppose X is not satisfiable. Then for every plan in $\mathcal{Q}$, there will be at least one i such that none of l_{i1}, l_{i2}, l_{i3} is satisfied. Thus $\mathsf{csat}_i(\mathsf{no})$ will be true in Final's input state, but $\mathsf{csat}_i(\mathsf{yes})$ will not. Thus in every executable completion of Q^*_X, the ground atom $\mathsf{sat}(u_1, u_2, \ldots, u_m)$ asserted by Final will contain at least one $u_i = \mathsf{no}$. ■