On Strongest Necessary and Weakest Sufficient Conditions*

Fangzhen Lin (flin@cs.ust.hk)
Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

Abstract

Given a propositional theory T and a proposition q, a sufficient condition of q is one that will make q true under T, and a necessary condition of q is one that has to be true for q to be true under T. In this paper, we propose a notion of strongest necessary and weakest sufficient conditions. Intuitively, the strongest necessary condition of a proposition is the most general consequence that we can deduce from the proposition under the given theory, and the weakest sufficient condition is the most general abduction that we can make from the proposition under the given theory. We show that these two conditions are dual ones, and can be naturally extended to arbitrary formulas. We investigate some computational properties of these two conditions and discuss some of their potential applications.

1 Introduction

Given a propositional theory T and a proposition q, a sufficient condition α of q is one that will make q true under T: $T \models \alpha \supset q$. Symmetrically, a necessary condition α of q is one that has to be true for q to be true under T: $T \models q \supset \alpha$.

For instance, consider the following theory T:

$$rain \supset grassWet,$$
$$sprinklerOn \supset grassWet.$$

A sufficient condition for $grassWet$ to be true under T is for $rain$ to be true. Another (trivial) one is $grassWet$ itself. Yet another one is $sprinklerOn$. And yet another one is $rain \vee sprinklerOn$.

In this paper we shall propose a notion of weakest sufficient and strongest necessary conditions, study their properties, and consider ways of computing them. There are many potential applications. The following are two examples:

*An earlier version of this paper appeared in the *Proceedings of KR2000*, and was selected as the co-winner of the Best Paper Awards.

- **Abduction** As we can see from the above example, given an observation, there may be more than one abductive conclusions that we can draw. It should be useful to find the weakest of such conclusions, i.e. the weakest sufficient condition of the observation.

- **Definability** It is often necessary to determine whether a given theory yields a definition of a proposition in terms of a set of other propositions. For example, such computation is essential in both Simon's [17] and Pearl's [14] approaches to causation. This is also what is needed in order to compute successor state axioms [15] from causal theories. As we shall see, the problem of whether a proposition can be defined in terms of other propositions, and if so, finding an explicit definition of this proposition can be reduced to that of computing the two conditions that we shall propose in this paper. Furthermore, we believe definability is best handled using these two conditions. While a proposition may or may not be definable in terms of a set of base propositions, our strongest necessary and weakest sufficient conditions, as we shall show, always exist and are unique up to logical equivalence under the given theory. Furthermore, these two conditions are useful to have even when a proposition cannot be defined in terms of others. For instance, in the context of reasoning about actions, this situation arises when an action has an indeterminate effect on a proposition. In this case, the strongest necessary and weakest sufficient conditions of this proposition can be used to completely capture the effect of this action on the proposition.

This paper is organized as follows. We include logical preliminaries in section 2. In section 3, we define the notion of strongest necessary and weakest sufficient conditions, and prove some results that characterize these conditions. In section 4, we extend these two conditions to that of arbitrary formulas. In Section 5 we outline some algorithms for computing these two conditions and discuss some experimental results. In section 6, we discuss some related work, and finally in section 7, we conclude this paper.

2 Logical preliminaries

We assume a propositional language. As usual, a truth assignment is a function from the set of propositions in the language to $\{true, false\}$, and the satisfiability relation $M \models \varphi$ is defined as usual between a truth assignment M and a formula φ.

Given a set P of propositions, a formula that mentions only propositions in P is called a *formula of P*. It is clear that the truth value of such a formula is determined by the truth values of the propositions in P only.

Given a formula φ, and a proposition p, we denote by $\varphi(p/true)$ the result of replacing every p in φ by $true$. We extend this notation to a set of formulas as well: if T is a finite set of formulas, then by $T(p/true)$ we mean the formula

$$\bigwedge_{\varphi \in T} \varphi(p/true).$$

We define $\varphi(p/false)$ and $T(p/false)$ similarly. It is clear that for any truth assignment M, $M \models \varphi(p/true)$ iff $M' \models \varphi$, where M' is the truth assignment that is exactly like M but $M'(p) = true$. A similar result holds for $\varphi(p/false)$.

A notion of forgetting, or eliminations of "middle terms" as Boole ([2], page 99) called it, will be essential here. Given a formula φ, and a proposition p, the result of forgetting p in φ, written $forget(\varphi; p)$, is the following formula:

$$\varphi(p/true) \vee \varphi(p/false).$$

Now given a finite set of propositions P, the result of forgetting P in φ, written $forget(\varphi; P)$, is defined inductively as:

$$forget(\varphi; \emptyset) = \varphi,$$
$$forget(\varphi; P \cup \{q\}) = forget(forget(\varphi; q); P).$$

It can be shown that for any formula φ and any propositions p_1 and p_2, $forget(forget(\varphi; p_1); p_2)$ and $forget(forget(\varphi; p_2); p_1)$ are equivalent. So $forget(\varphi; P)$ is well defined.

Since Boole [2] first defined it in 1854, this notion of forgetting, or eliminations of middle terms, has been re-discovered several times. For instance, Weber [19] re-introduced it and used it for updating propositional knowledge bases. It is also a special case of a more general notion of forgetting defined on first-order languages by Lin and Reiter [12] for capturing database progression [11] and certain notions of relevance in problem solving. Lewis [8] (page 155) observed that "For purposes of application of the algebra[1] to ordinary reasoning, elimination is a process more important than solution, since most processes of reasoning take place through the elimination of 'middle' terms." As we shall see, this notion of "the elimination of middle terms" is certainly central for us here.

The following lemma is entailed by Proposition 3 in (Lin and Reiter [12]), and will play an important role in this paper.

Lemma 1 *For any formula φ and any set P of propositions, if ϕ does not mention any propositions in P, then*

$$\varphi \models \phi \Leftrightarrow forget(\varphi; P) \models \phi.$$

In particular, we have that $\varphi \models forget(\varphi; P)$.

3 Strongest necessary and weakest sufficient conditions

We begin with strongest necessary conditions. As we shall see, weakest sufficient conditions are dual ones.

[1]The author's note: "the algebra" = Boolean algebra as we know it today.

Definition 1 *Let T be a theory, P a set of propositions in T, and q a proposition in T but not in P. A formula φ of P is said to be a* necessary condition *of q on P under T if $T \models q \supset \varphi$. It is said to be a* strongest necessary condition *if it is a necessary condition, and for any other necessary condition φ', we have that $T \models \varphi \supset \varphi'$.*

In this paper, both the background theory T and the base set P of propositions are assumed to be finite. The following proposition is straightforward:

Proposition 1 *If both φ and φ' are strongest necessary conditions of q on P under T, then $T \models \varphi \equiv \varphi'$.*

This means that if there is a strongest necessary condition, then it is unique up to logical equivalence under the background theory. So sometimes, we also call a strongest necessary condition *the* strongest one.

Example 1 The following examples provide some intuitions about our notion of strongest necessary conditions.

1. $T_1 = \{q \supset (p_1 \wedge p_2)\}$. The strongest necessary condition of q on $\{p_1, p_2\}$ under T_1 is $p_1 \wedge p_2$, and the strongest necessary condition of q on $\{p_1\}$ is p_1.
2. $T_2 = \{q \supset (p_1 \vee p_2)\}$. The strongest necessary condition of q on $\{p_1, p_2\}$ is $p_1 \vee p_2$, and the strongest necessary condition of q on $\{p_1\}$ is $true$ because neither p_1 nor $\neg p_1$ follow from q under T_2.
3. $T_3 = \{q \supset p_1, q\}$. A strongest necessary condition of q on $\{p_1\}$ is p_1. Another one is $true$. Of course, $T_3 \models p_1 \equiv true$.

The following theorem captures this notion of strongest necessary conditions in terms of truth assignments. It says that a necessary condition φ of q is a strongest one on P if and only if for any model of T, if it satisfies φ, then it can be modified into a model of q without changing the truth values of the propositions in P:

Theorem 1 *Let T, P, and q be as in Definition 1. Let φ be a necessary condition of q on P under T. Then φ is a strongest necessary condition of q on P under T iff for any model M of T and φ, there is an assignment M' such that:*

1. *for any $p \in P$, $M'(p) = M(p)$.*
2. *$M' \models T \cup \{q\}$.*

Proof: The "if" part: suppose φ is the strongest necessary condition (SNC) of q, and $M \models T \cup \{\varphi\}$. We need to show the existence of a model M' that satisfies the two conditions in the theorem. We prove by contradiction. Suppose there is no such M', i.e. there is no model of $T \cup \{q, M(P)\}$, where $M(P)$ is the conjunction of p, if $M(p) = true$, or $\neg p$, if $M(p) = false$, for every $p \in P$.

Then $T \models q \supset \neg M(P)$. This means that $\neg M(P)$ is a necessary condition (NC) of q on P. Thus $T \models \varphi \supset \neg M(P)$, this contradicts with the assumption that $M \models T \cup \{\varphi\}$.

The "only if" part: suppose for any model M of T and φ, there is another model M' that satisfies the two conditions in the theorem. We need to show that φ is the SNC of q. Let φ' be another NC of q: it is a formula of P and $T \models q \supset \varphi'$. We need to prove that $T \models \varphi \supset \varphi'$. Let M be any model of φ and T. We show that M is also a model of φ'. By our assumption, there is another model M' that satisfies the two conditions in the theorem. This means that $M' \models q$ and $M' \models T$. So $M' \models \varphi'$. But $M(P) = M'(P)$, and φ' is a formula of P. So we have $M \models \varphi'$ as well. ■

Notice that it is essential that the formula φ in Theorem 1 be a necessary condition of q. Otherwise, we could always let φ be $false$.

We can use this theorem to verify that a certain necessary condition is in fact the strongest. For instance, consider $T_1 = \{q \supset p_1 \wedge p_2\}$ in Example 1 above. Clearly, p_1 is a necessary condition of q on $\{p_1\}$. It's a strongest one because given any model of T_1, if it satisfies p_1, then we can modify it into another model of T_1 and q while preserving the truth value of p_1. However, although it is also a necessary condition of q on $\{p_1, p_2\}$, it is not the strongest one this time, because if $M \models p_1 \wedge \neg p_2 \wedge \neg q$, then $M \models T_1 \cup \{p_1\}$, but it cannot be modified into a model of q and T_1 without changing the truth value of p_2.

We can similarly define the notion of weakest sufficient conditions:

Definition 2 *Let T be a theory, P a set of propositions in T, and q a proposition in T but not in P. A formula ψ of P is said to be a* sufficient condition *of q on P under T if $T \models \psi \supset q$. It is said to be a* weakest sufficient condition *if it is a sufficient condition, and for any other sufficient condition ψ', we have that $T \models \psi' \supset \psi$.*

Strongest necessary and weakest sufficient conditions are in fact dual conditions. The easiest way to make this dual relationship precise is to extend these conditions from propositions to arbitrary formulas and to show that for any formula A, a formula φ is a strongest necessary (weakest sufficient) condition of A iff $\neg\varphi$ is a weakest sufficient (strongest necessary) condition of $\neg A$. This will be done in section 4.

We now relate these two conditions to the notion of definability. We say that a theory T *defines a proposition* q on a set P of propositions iff there is a formula φ of P such that $T \models q \equiv \varphi$. The following proposition is straightforward:

Proposition 2 *A theory T defines a proposition q on P iff $T \models \varphi \supset \phi$, where φ is any strongest necessary condition of q on P and ϕ any weakest sufficient condition of q on P, both under the theory T.*

Notice that when the condition $T \models \varphi \supset \phi$ holds, we have that $T \models \varphi \equiv \phi$ and $T \models q \equiv \varphi$. So this proposition reduces the problem of computing definability to that of computing strongest necessary and weakest sufficient conditions.

An advantage of working with the latter is that the two conditions always exist and are unique up to logical equivalence under any given theory. Furthermore, they are useful to have even when they do not yield a definition of a proposition. For instance, in logic of programs, Dijkstra's notion of weakest preconditions is a special case of our notion of weakest sufficient conditions, and as shown in [1, 6], these conditions provide a good way to reason about the indeterminate effects of actions for which successor state axioms (definitions) do not exist.

4 Strongest necessary and weakest sufficient conditions of a formula

Our notion of strongest necessary and weakest sufficient conditions can be extended from a proposition to an arbitrary formula.

Definition 3 *Let T be a propositional theory, α a formula, and P a set of propositions in $T \cup \{\alpha\}$. A formula φ of P is said to be a necessary condition of α on P iff $T \models \alpha \supset \varphi$. It is said to be a strongest necessary condition if for any other necessary condition φ', we have that $T \models \varphi \supset \varphi'$. Sufficient conditions and weakest sufficient conditions are defined similarly.*

Computing the strongest necessary and the weakest sufficient conditions of a formula can be reduced to that of a proposition, as the following proposition shows.

Proposition 3 *Let T, P, and α be as in Definition 3. A formula φ of P is the strongest necessary (weakest sufficient) condition of α on P under the theory T iff it is the strongest necessary (weakest sufficient) condition of q on P under the theory $T' = T \cup \{q \equiv \alpha\}$, where q is a new proposition not in T and α.*

Proof: We prove this for SNC. The case for WSC is similar. The "if" part: suppose φ is the SNC of α. We show that it is also an SNC of q under T'. It is easy to see that it is an NC of q: $T' \models q \supset \varphi$. Let φ' be any other necessary condition: $T' \models q \supset \varphi'$. Thus $T' \models \alpha \supset \varphi'$. Since q does not occur in α and φ', by Lemma 1, we have that $forget(T', q) \models \alpha \supset \varphi'$. But $forget(T', q)$ is equivalent to T since q does not occur in T and α. So φ' is also a necessary condition of α on P under T. Thus $T \models \varphi \supset \varphi'$. So $T' \models \varphi \supset \varphi'$.

The "only if" part: suppose φ is the SNC of q under T'. First of all, it is an NC: $T' \models q \supset \varphi$. Similar to the reasoning above, we have $T \models \alpha \supset \varphi$. So φ is also an NC of α on P under T. Now if φ' is any other NC of α under T: $T \models \alpha \supset \varphi'$. Then it is also an NC of q under T'. Thus by our assumption that φ is the SNC of q, we have that $T' \models \varphi \supset \varphi'$. Thus $T \models \varphi \supset \varphi'$ since q does not occur in T. ■

Although not necessary in principle, the notion of sufficient and necessary conditions of a formula is very useful in expressing many properties. The following are some interesting ones.

The first one says that, as expected, if two formulas are equivalent under T, then they have the same strongest necessary and weakest sufficient conditions:

Proposition 4 *If $T \models \alpha \equiv \beta$, then for any set of propositions P, a formula φ is a strongest necessary (weakest sufficient) condition of α on P iff it is a strongest necessary (weakest sufficient) condition of β on P.*

Proof: Suppose φ is the SNC of α on P under T. Then it is NC of β as well. Now let φ' be any SC of β: $T \models \beta \supset \varphi'$. Then we have $T \models \alpha \supset \varphi'$, so $T \models \varphi \supset \varphi'$. Therefore φ is also an SNC of β on P. The proof that if φ is an SNC of β, then it is also an SNC of α is symmetric.

The proof for WSC is similar. ■

The following proposition makes precise the dual relation between strongest necessary and weakest sufficient conditions:

Proposition 5 *A formula φ is the strongest necessary (weakest sufficient) condition of q iff $\neg\varphi$ is the weakest sufficient (strongest necessary) condition of $\neg q$, where all the conditions are on a common set of propositions and under a common theory.*

Proof: Suppose φ is the SNC of q. Then $T \models q \supset \varphi$. Thus $T \models \neg\varphi \supset \neg q$. So $\neg\varphi$ is a SC of $\neg q$. Suppose φ' is any other SC of $\neg q$: $T \models \varphi' \supset \neg q$. Then $T \models q \supset \neg\varphi'$. Thus $T \models \varphi \supset \neg\varphi'$. So $T \models \varphi' \supset \neg\varphi$. This proves that $\neg\varphi$ is the WSC of $\neg q$.

The proof of the other part of the proposition is similar. ■

Proposition 6 *If φ_1, ..., φ_k are the strongest necessary conditions of $\alpha_1, ..., \alpha_k$, respectively, then $\varphi = \varphi_1 \vee \cdots \vee \varphi_k$ is the strongest necessary condition of $\alpha_1 \vee \cdots \vee \alpha_k$, where all the conditions are on a common set of propositions and under a common theory.*

Proof: First of all, φ is an NC: $T \models \alpha \supset \varphi$. We show that it is the SNC using Theorem 1, generalized to arbitrary formula using Proposition 3: we show that if $M \models \varphi$ and $M \models T$, then there is a M' such that $M' \models T$, $M' \models \alpha$, and $M'(P) = M(P)$. Since $M \models \varphi$, there must be an $1 \leq i \leq k$ such that $M \models \varphi_i$. By Theorem 1 and Proposition 3, there is an M' such that $M'(P) = M(P)$, $M' \models T$, and $M' \models \alpha_i$, thus $M' \models \alpha$. ■

Symmetrically, for weakest sufficient conditions, we have:

Proposition 7 *If φ_1, ..., φ_k are the weakest sufficient conditions of $\alpha_1, ..., \alpha_k$, respectively, then $\varphi = \varphi_1 \wedge \cdots \wedge \varphi_k$ is the weakest sufficient condition of $\alpha_1 \wedge \cdots \wedge \alpha_k$, where all the conditions are on a common set of propositions and under a common theory.*

This proposition is particularly useful in planning when we do not have a successor state axiom for every fluent: given a conjunctive goal $g_1 \wedge \cdots \wedge g_n$, to achieve it using action A, we need to compute the weakest sufficient condition of this conjunction, and reduce the conjunctive goal to it. By this proposition, instead of having to compute the weakest sufficient condition for every possible conjunction, it is enough to compute the condition for each conjunct ahead of time.

5 Computing strongest necessary and weakest sufficient conditions

There are several ways to compute these two conditions. We begin with prime implicates [18]. As with most reasoning tasks, strongest necessary and weakest sufficient conditions can be easily computed using prime implicates.

Proposition 8 *Let T be a theory, q a proposition, and P a set of propositions. Let Π be the set of prime implicates of T that mention only propositions in $P \cup \{q\}$.*

1. *The strongest necessary condition of q on P under T is equivalent to the following conjunction:*

$$\bigwedge \{C \mid \neg q \vee C \in \Pi\},$$

where if there is no such prime implicate in Π, then the conjunction is considered to be $true$, and if $\neg q \in \Pi$, then the conjunction is considered to be $false$.

2. *The weakest sufficient condition of q on P under T is equivalent to the following disjunction:*

$$\bigvee \{\neg C \mid q \vee C \in \Pi\},$$

where if there is no such prime implicate in Π, then the disjunction is considered to be $false$, and if $q \in \Pi$, then the conjunction is considered to be $true$.

Proof: We prove this proposition for SNC. Let W be the conjunction in the proposition. Clearly, $T \models q \supset W$, so W is an NC. Now let ψ be any other NC: $T \models q \supset \psi$, and Γ the set of prime implicates of ψ. Since ψ does not mention q, the following set

$$\Gamma' = \{\neg q \vee C \mid C \in \Gamma\}$$

is the set of prime implicates of $q \supset \psi$, where if $C = true$ (i.e., ψ is a tautology), then $\neg q \vee C$ is $true$, and if $C = false$ (i.e., ψ is not satisfiable), then $\neg q \vee C$ is $\neg q$. Therefore for each $\neg q \vee C$ in Γ', there is a prime implicate $\neg q \vee C'$ in Π such that C is subsumed by C'. Thus each prime implicate in Γ is subsumed by a conjunct in W. So $W \models \psi$. ■

However, computing the strongest necessary (similarly weakest sufficient) condition of a proposition using prime implicates is almost always a bad idea, unless one wants these conditions for all propositions on all possible sets of propositions and wants them in the form of prime implicates. This is because, in general, there is no viable way of computing the set Π in Proposition 8 short of computing the set of all prime implicates of a theory. But once having the set of prime implicates, one can just "read" out these two conditions for any proposition using Proposition 8.

Nevertheless, Proposition 8 is interesting in that it relates our notion of weakest sufficient conditions to Reiter and de Kleer's notion of *minimal support clauses* of a set of clauses [16]. According to Corollary 4 in [16], a clause C is a minimal support clause for a proposition q with respect to T iff there is a prime implicate C' of T such that $q \in C'$ and $C = C' - \{q\}$. So from above proposition, for any theory T, and proposition q, if P is the set of propositions that are in T but different than q, then the weakest sufficient condition of a q on P under T is equivalent to the disjunction of the negation of each minimal support clause for q with respect to T.

However, our two conditions are defined independent of prime implicates. In fact, a better way of computing them is in terms of the notion of forgetting that we defined in Section 2, by using the following theorem:[2]

Theorem 2 *Let T be a theory, P a set of propositions, and q a proposition in T but not in P. Let P' be the set of propositions that are in T but not in $P \cup \{q\}$. Then we have:*

1. *The strongest necessary condition of q on P is $forget(T(q/true); P')$.*
2. *The weakest sufficient condition of q on P is $\neg forget(T(q/false); P')$.*

Proof: We prove the case for SNC. The case for WSC is similar. First of all, $forget(T(q/true); P')$ is an NC: $T \models forget(T; P')$, so $T \models q \supset forget(T; P')$, thus $T \models q \supset forget(T(q/true); P')$ because $q \notin P'$. It is the strongest: suppose α is an NC: α is a formula of P, and $T \models q \supset \alpha$. Thus, by Lemma 1, we have that $forget(T; P') \models q \supset \alpha$, Therefore $q \models forget(T; P') \supset \alpha$. So $q \models forget(T(q/true); P') \supset \alpha$. Thus $\models forget(T(q/true); P') \supset \alpha$ because q does not occur in $forget(T(q/true); P') \supset \alpha$. ■

We illustrate the use of this theorem using the following theory

$$T = \{q \supset p_1 \vee p_2\}.$$

According to the theorem, the strongest necessary condition of q on $\{p_1, p_2\}$ is $forget(T(q/true); \emptyset)$, which is $T(q/true)$ and equivalent to $p_1 \vee p_2$, the same

[2] Lang and Marquis [7] showed that a proposition q is definable on P under T iff $T \models q \equiv forget(T(q/true); P')$. By Proposition 2, our theorem generalizes this result.

as given in Example 1. The strongest necessary condition of q on $\{p_1\}$ is $forget(T(q/true); \{p_2\})$, which is equivalent to $forget(p_1 \vee p_2; \{p_2\})$, which by definition is $(p_1 \vee true) \vee (p_1 \vee false)$, thus equivalent to $true$.

By its definition, for any formula φ and proposition p, $forget(\varphi; p)$ returns a formula that is twice the size as that of φ; and for a set P of propositions, $forget(\varphi; P)$ returns a formula whose size is 2^n times the size of φ, where n is the size of P. Indeed, computing forgetting in the worst case is expensive. For instance, it is easy to see that a formula is satisfiable iff forgetting all propositions in it returns $true$, and a formula is not satisfiable iff forgetting all propositions in it returns $false$. However, $forget(\varphi; p)$ can often be simplified. For instance, if φ is in disjunctive normal form, then $forget(\varphi; p)$ can be computed efficiently to yield a formula that is shorter than φ: it is the result of replacing both p and $\neg p$ in φ by $true$. More generally, as Darwiche [5] showed, if φ is what he called decomposable negation normal form, then $forget(\varphi; p)$ can be computed efficiently.

In this paper, we consider two ways of computing forgetting: directly using its definition or through (again) prime implicates. They are embodied below as Algorithms 1 and 2 for computing strongest necessary conditions. The ones for weakest sufficient conditions are similar.

Input A set T of axioms, a set P of propositions, and a proposition q not in P.
Output A formula of P that is the strongest necessary condition of q on P under T.

Algorithm 1

1. Let $T_1 = \{\varphi \mid \varphi \in T \text{ is a sentence of } P\}$, and $T_2 = T - T1$. Replace q in T_2 by $true$, and transform the resulting theory into a *minimal set of clauses* C (see below for a definition). Similarly, transform T_1 into a minimal set of clauses C_0.
2. Let P' be the set of propositions in C but not in P.
3. If P' is empty, then go to post-processing, otherwise select a proposition p in P', and delete it from P'.
4. Transform $C(p/true) \vee C(p/false)$ into a minimal set of clauses, assign it to C, and go back to step 3.
5. *Post-processing:* although the resulting C is a strongest necessary condition of q, it is often an unwieldy set of clauses; it can be simplified as follows:
   - eliminate those clauses in C that are subsumed by one of the clauses in C_0;
   - for each clause α in C, transform $(C - \{\alpha\}) \cup C_0$ into a minimal set of clauses C_α, and eliminate α from C if it is subsumed by one of the clauses in C_α.

Where a minimal set of clauses is one such that:

- all unit clauses are resolved;
- none of the clauses is subsumed by any other clause in the set.

Our experience with this algorithm has been that it spends the bulk of the time on computing minimal sets of clauses in step 4. But if we do not require a set of clauses to be minimal, then the program quickly runs out of space.

Notice that according to our definition, for a set of clauses C to be a minimal one, only unit clauses in it need to be resolved. So our notion of minimality is much weaker than that of prime implicates. For comparison, let us consider computing forgetting using prime implicates.

The following algorithm makes use of the fact that for any φ and P, $forget(\varphi; P)$ is equivalent to the conjunction of prime implicates of φ that do not mention any propositions in P.

Algorithm 2

1. Let $T_1 = \{\varphi \mid \varphi \in T \text{ is a sentence of } P\}$, and $T_2 = T - T1$. Transform T_1 into a minimal set of clauses C_0.
2. Generate the prime implicates of $T_2(q/true)$ using Tison's procedure [18], and let C be the set of those prime implicates that mention only propositions in P.
3. Go to post-processing as in Algorithm 1.

We have implemented the above two algorithms in SWI-Prolog[3], and run them on both random 3CNFs[4] and action theories used for computing successor state axioms in various action domains [10]. It turned out that Algorithm 2 is always slower than Algorithm 1, with on average a slow down of between 10 to 15 times. The reason is that computing prime implicates is expensive. Our notion of minimal sets of clauses seems to serve our purpose well here in cutting down the number and sizes of clauses without incurring too much overhead in computing them.

Notice that in step 4 of Algorithm 1, after we have chosen a proposition to forget, we compute immediately the result of forgetting this proposition, $C(p/true) \vee C(p/false)$, by a set of minimal clauses. This is like breadth-first search, and as we have mentioned, is the bottleneck of this algorithm. Alternatively, we could work on each disjunct separately, and compute disjunctions only after all those propositions that need to be forgotten have been eliminated. As it turned out, this variant of Algorithm 1 performed far better than Algorithm 1 on random 3CNFs. For random 3CNFs with 50 variables and 215 clauses, this variant of Algorithm 1 spent on average 15 minutes[5] to return a strongest necessary condition of a proposition on a set of 10 other propositions, while Algorithm 1 did not return a solution after running overnight. However, this variant of Algorithm 1 is about 20% slower on action theories. The key is with

[3] SWI-Prolog is developed by Jan Wielemaker at the University of Amsterdam

[4] Generated using a program by Kautz and Selman.

[5] On a SPARC Ultra2 machine running SWI-Prolog.

the disjunction $C(p/true) \vee C(p/false)$: it can be simplified a lot for many benchmark action theories, but not much for random theories.

Regardless of which approach one uses, we have found that sometimes, it is a lot easier to compute the strongest necessary condition than the weakest sufficient condition, and sometimes, it is the other way around. For instance, we have found that for many action theories, strongest necessary conditions are a lot easier to compute than weakest sufficient conditions. When one condition is much easier to compute, the following proposition will be very useful in computing the other one.

Proposition 9 *Let T be a theory, q a proposition, and P a set of propositions.*

1. *If φ is a necessary condition of q on P under T, and ψ a weakest sufficient condition of q on P under $T \cup \{\varphi\}$, then $\varphi \wedge \psi$ is a weakest sufficient condition of q on P under T.*

2. *If ψ is a sufficient condition of q on P under T, and φ a strongest necessary condition of q on P under $T \cup \{\neg\psi\}$, then $\varphi \vee \psi$ is a strongest necessary condition of q on P under T.*

Proof:

1. First of all, $\varphi \wedge \psi$ is an SC: $T \cup \{\varphi\} \models \psi \supset q$, so $T \models \varphi \wedge \psi \supset q$. It is the weakest: suppose α is an SC: $T \models \alpha \supset q$. We need to show that $T \models \alpha \supset \varphi \wedge \psi$. Since $T \models q \supset \varphi$. We have $T \models \alpha \supset \varphi$. But α is also an SC of q under $T \cup \{\varphi\}$, so $T \cup \{\varphi\} \models \alpha \supset \psi$ because ψ is the WSC of q under $T \cup \{\varphi\}$. Therefore $T \models \alpha \supset (\varphi \wedge \psi)$.

2. Suppose φ' is an NC of q under T: $T \models q \supset \varphi'$. From $T \models \psi \supset q$, we have $T \models \psi \supset \varphi'$. From $T \cup \{\neg\psi\} \models q \supset \varphi$ and $T \cup \{\neg\psi\} \models q \supset \varphi'$, we have $T \cup \{\neg\psi\} \models \varphi \supset \varphi'$. Thus $T \models \neg\psi \wedge \varphi \supset \varphi'$, this together with $T \models \psi \supset \varphi'$, we have $T \models \varphi \supset \varphi'$.

■

We illustrate this phenomenon for the problem of generating successor state axioms in a robot domain. To simplify our presentation, we assume that there is just one object in this domain, and the robot can carry the object around by moving from one location to another. Now suppose that $move(1, 2)$ stands for the action that the robot moves from location 1 to location 2, and that we have the following propositions:

- at_i - initially, the object is at location i, for each location i.
- $at1_i$ - after the action $move(1, 2)$, the object is at location i.
- atR_i - initially, the robot is at location i.
- $atR1_i$ - after the action, the robot is at location i.
- h - initially, the robot is holding the object.

- $h1$ - after the action, the robot is holding the object.

The background theory is then the following set of axioms:

$$
\begin{aligned}
& atR_1 \wedge atR1_2, \\
& atR1_i \equiv atR_i \wedge \neg \bigvee_{j \neq i} atR1_j, \\
& at1_i \equiv [atR1_i \wedge h1] \vee [at_i \wedge \neg \bigvee_{j \neq i} at1_j], \\
& h1 \equiv h, \\
& atR_i \supset \neg atR_j, \text{ for any } j \neq i \\
& at_i \supset \neg at_j, \text{ for any } j \neq i
\end{aligned}
$$

where the disjunctions are ranged over all locations. These axioms are generated from the direct effect action axiom: after $move(1,2)$, the robot will be at location 2, and domain constraints such as "the object and the robot can be at only one location at any given time" and "if the robot is holding the object, then the robot and the object are always at the same location" using a predicate completion procedure in [9]. More details about this procedure, and the way the above axioms are generated from a causal theory can be found in (Lin [10]). Now let q be $at1_2$, and P the set of propositions about the initial situation:

$$P = \{h\} \cup \{at_i \mid i \text{ is a location}\} \cup \{atR_i \mid i \text{ is a location}\}.$$

Then the weakest sufficient condition of q on P is the weakest condition about the initial situation that would ensure that the package is at location 2 after the action is performed, and the strongest necessary condition is the strongest conclusion that one can infer about the initial situation once one knows that the package is at location 2 after the action is performed.

Figure 1 compares the performance of computing the weakest sufficient condition of q on P using Algorithm 1 with or without the aid of Proposition 9. In the figure, the x-axis is the number of locations, and y-axis the CPU run time in seconds on a SPARC Ultra 2. The line labeled by *snc* corresponds to the strongest necessary condition of q on P,[6] the one labeled by *wsc* the weakest sufficient condition of q on P,[7] and *wsc*1 the weakest sufficient condition of q on P once the strongest sufficient condition has been added to the theory[8] according to Proposition 9. The speedup of *wsc*1 over *wsc* is quite obvious, and similar speedup is also achieved for other propositions and other action domains that we have experimented with, which include most of the benchmark planning domain in McDermott's PDDL library [13].

[6] For which the program outputs $h \vee at_2$ no matter how many locations there are in the domain.

[7] For which the program outputs $h \vee at_2$.

[8] For which the program outputs $true$.

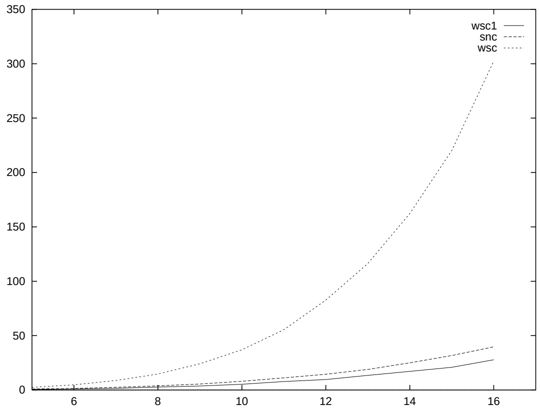

Figure 1: A robot domain

6 Related work

Our notion of strongest necessary and weakest sufficient conditions bear structural resemblance to Dijkstra's notion of strongest postconditions and weakest preconditions of programs (cf. [6]). However, Dijkstra was mainly concerned about compositional rules of these conditions over program operators, once these conditions are given for basic assignment statements. In contrast, our study of these two conditions are completely general. Even in applications to computing the effects of actions, our concerns for these two conditions differ from that of Dijkstra, and similarly recent applications of Dijkstra's notion to reasoning about action by Bjäreland and Karlsson ([1]), in that we are interested in computing the effect of *primitive* actions from a set of direct effect axioms and causal rules.

Our notion of weakest sufficient conditions is closely related to abduction and diagnosis. We have mentioned the relationship between our weakest sufficient conditions and Reiter and de Kleer's minimal support clauses. In general, we are interested here in logical formulas that capture all possible explanations without committing ourself into the exact formats that these formulas ought to have, while much of the work on abduction and diagnosis has been model-based, and mainly concerned with computing parsimonious and minimal explanations.

However, there are some exceptions. In logic programming, Console *et al.* ([3]) investigated abduction through deduction through the Clark completion of a logic program. They defined the *explanation formula* of an abduction problem to be *the most specific formula* in the language of abducible atoms such that the formula is entailed by the completion of the program and the observation. Here the most specific formula is like our strongest condition, but with a difference. According to Console *et al.*, a formula φ is more specific than φ' if $\varphi \models \varphi'$, while for us, this entailment only needs to be true under the given logical theory. This somewhat minor difference has some far reaching ramifications, as we shall see in connection with a more recent work by Darwiche ([4]) who proposed a very similar notion and applied it to diagnosis problems.

Given a theory T, a formula α, and a set of propositions P such that none of the propositions in P occur in α, Darwiche [4] defined the *consequence* of α on P to be a formula φ such that $T \cup \{\alpha\} \models \varphi$, and for any such φ', we have that $\varphi \models \varphi'$. Similarly, he defined the *argument* of α to be a formula φ such that $T \cup \{\varphi\} \models \alpha$, and for any such φ', we have that $\varphi' \models \varphi$. As one can see, Darwiche's notion of consequences generalizes that of explanation formulas by Console *et. al* above, and his notion of arguments corresponds to our notion of weakest sufficient conditions. But again, the crucial difference is in how formulas are compared. In Darwiche's case, as in the case of Console *et al.*, when φ and φ' are compared, they are compared to under logical entailment. But in our case, they are compared to under logical entailment *relative* to the background theory T. The difference has some significant ramifications. First of all, if T defines α: $T \models \alpha \equiv \varphi$, then φ is both the strongest necessary and weakest conditions of α. But φ may be neither the consequence nor the argument of α according to Darwiche's definitions. Secondly, the post-processing step in Algorithm 1 will not be correct according to Darwiche's definitions. Perhaps most significantly, our Proposition 9 would not be true under Darwiche's definitions. For instance, consider the theory $T = \{p \supset q, q \supset \neg p\}$. According to Darwiche's definitions, the consequence of q on $\{p\}$ w.r.t. T is $\neg p$. So for Proposition 9 to hold in Darwiche's framework, the argument of q on $\{p\}$ w.r.t. T would have to be equivalent to the conjunction of $\neg p$ and the argument of q on $\{p\}$ w.r.t. $T \cup \{\neg p\}$. Unfortunately, this is not true, as the conjunction is equivalent to $false$, which is not the argument of q on $\{p\}$ w.r.t. T. Notice that $false$ is the weakest sufficient condition of q on $\{p\}$ under T, which means that under no condition on p can q be true as $\neg p$ is entailed by the theory. This last difference between our notion and Darwiche's is significant as Proposition 9 is the basis of one of the key strategies that we used for computing the weakest sufficient conditions in relation to action theories.

7 Concluding remarks and future work

We have proposed a notion of strongest necessary and weakest sufficient conditions of a proposition, and considered ways of computing them. We believe these conditions have many potential applications in various areas including abduction and reasoning about actions.

There are several directions for future work. One of them is to extend the results here to the first-order case. This can be a difficult task. For instance, a result in [11] shows that forgetting in the first-order case cannot in general be expressible in first-order logic. As a consequence, we expect that strongest necessary conditions of a proposition under a first-order theory cannot in general be expressible in first-order logic either. It seems that the best hope for dealing with the first-order case is to first reduce it to the propositional case, and then try to learn a first-order description from a set of propositional ones.

Acknowledgments

I would like to thank Maurice Pagnucco for his comments on an earlier version of this paper, and Adnan Darwiche for discussions related to the subject of this paper. This work was supported in part by the Research Grants Council of Hong Kong under Competitive Earmarked Research Grants HKUST6091/97E and HKUST6145/98E.

References

[1] M. Bjäreland and L. Karlsson. Reasoning by regression: pre- and postdiction procedures for logics of action and change with nondeterminism. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI–97), IJCAI Inc. Distributed by Morgan Kaufmann, San Mateo, CA.*, pages 1420–1425, 1997.

[2] G. Boole. *An Investigation of the Laws of Thought.* Walton, London, 1854. (Reprinted by Dover Books, New York, 1954).

[3] L. Console, D. Dupre, and P. Torasso. On the relationship between abduction and deduction. *Journal of Logic and Computation*, 1(5):661–690, 1991.

[4] A. Darwiche. A logical notion of conditional independence: Properties and applications. *Artificial Intelligence*, 97(1-2):45–82, 1997.

[5] A. Darwiche. Compiling knowledge into decomposable negation normal form. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI–99), IJCAI Inc. Distributed by Morgan Kaufmann, San Mateo, CA.*, 1999.

[6] E. W. Dijkstra and C. S. Scholten. *Predicate Calculus and Program Semantics.* Springer-Verlag, New York, 1990.

[7] J. Lang and P. Marquis. Two forms of dependence in propositional logic: controllability and definability. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI–98), AAAI Press, Menlo Park, CA.*, 1998.

[8] C. I. Lewis. *A Survey of Symbolic Logic.* University of California Press, Berkeley, 1918. (Reprinted by Dover Pub's., Inc., New York, 1960. Chapter II in The Classic or Boole-Schroder Algebra of Logic).

[9] F. Lin. Embracing causality in specifying the indirect effects of actions. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI–95), IJCAI Inc. Distributed by Morgan Kaufmann, San Mateo, CA.*, pages 1985–1993, 1995.

[10] F. Lin. From causal theories to successor state axioms: bridging the gap between nonmonotonic action theories and STRIPS-like formalisms. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI–2000), AAAI Press, Menlo Park, CA.*, To appear. Available at http://www.cs.ust.hk/faculty/flin.

[11] F. Lin and R. Reiter. How to progress a database. *Artificial Intelligence*, (92)1-2:131–167, 1997.

[12] F. Lin and R. Reiter. Forget it! In R. Greiner and D. Subramanian, editors, *Working Notes of AAAI Fall Symposium on Relevance*, pages 154–159. The American Association for Artificial Intelligence, Menlo Park, CA, November 1994. Also available at `http://www.cs.toronto.edu/cogrobo/forgetting.ps.Z`.

[13] D. McDermott et al. (AIPS-98 Planning Competition Committee). PDDL – the planning domain definition language. Technical Report TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, 1998.

[14] J. Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–709, 1995.

[15] R. Reiter. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 418–420. Academic Press, San Diego, CA, 1991.

[16] R. Reiter and J. de Kleer. Foundations of ATMS. In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI–87)*, 1987.

[17] H. Simon. Causal ordering and identifiability. In W. Hood and T. Koopmans, editors, *Studies in Econometric Method*. John Wiley and Sons, New York, 1953.

[18] P. Tison. Generalized consensus theory and application to the minimization of boolean functions. *IEEE Trans. on Electronic Computers*, EC-16/4:446–456, 1967.

[19] A. Weber. Updating propositional formulas. In *Proceedings First Conference on Expert Database Systems*, pages 487–500, 1986.