Weighted Abduction for Plan Ascription

Douglas E. Appelt and Martha E. Pollack

Artificial Intelligence Center
and
Center for the Study of Language and Information
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025 USA

Abstract

We describe an approach to abductive reasoning called *weighted abduction*, which uses inference weights to compare competing explanations for observed behavior. We present an algorithm for computing a weighted-abductive explanation, and sketch a model-theoretic semantics for weighted abduction. We argue that this approach is well suited to problems of reasoning about mental state. In particular, we show how the model of plan ascription developed by Konolige and Pollack can be recast in the framework of weighted abduction, and we discuss the potential advantages and disadvantages of this encoding.

Keywords: Plan recognition, Plan evaluation, Mental-state ascription, Abduction, Evaluation metrics

[0]This work was supported by a contract with the Nippon Telegraph and Telephone Corporation, and by a gift from the System Development Foundation. Our thanks to Kurt Konolige for valuable discussions on the research reported here.

1 Introduction

It is now widely accepted that cooperative interaction depends upon agents reasoning about one anothers' mental states. The process of "modeling the user" of an interactive system consists, in large part, in ascribing to the user a coherent set of beliefs, intentions, and possibly other mental attitudes that account for, or explain, his or her observed actions. Indeed, cooperative interaction depends not just on ascribing *any* set of explanatory mental attitudes, but rather on ascribing a mental state that in some sense provides the best explanation. Thus, a central concern in plan recognition and evaluation is to provide a precise specification of what counts as the best plan ascription to make, given some observed actions.[1]

Unfortunately, most of the existing work on plan ascription has failed to provide such a specification. As Kautz points out, typically "[o]nly a space of *possible* inferences is outlined, and little or nothing is said about why one should infer one conclusion over another, or what one should conclude if the situation is truly ambiguous" [Kau90, p. 106]. Kautz himself addresses this issue, providing an elegant formalization of plan recognition stated in terms of circumscription, in which it is made clear precisely which plan ascriptions should be preferred in a given situation. However, his account relies upon strong assumptions: that the agent performing the plan recognition (the *observer*) has complete knowledge of the domain, and that the agent whose plan is being inferred (the *actor*) has a correct plan. Pollack, in research on plan ascription in discourse understanding, has shown that these assumptions are too strong for any realistic, useful model of the process [Pol86, Pol90].

It is not obvious how to remove the strong assumptions from Kautz's

[1] We distinguish between *plan recognition*, in which the observer must determine the actor's goal as well as his plan, and *plan evaluation*, in which the observer is given the actor's goal. As we shall later show, the distinction between these problems, although subtle, influences their treatment in an abductive approach. We use the term *plan ascription* to refer to the general process that subsumes plan recognition and evaluation.

model.[2] Consequently, Konolige and Pollack [KP89] (hereafter KP) have attempted to provide a model of plan ascription that allows one to make explicit assertions about preferred ascriptions, while avoiding the overly strong assumptions inherent in Kautz's framework.

In this paper, we use KP's model as a starting point. However, where they employed an argumentation system for making inferences, we instead take an abductive approach. We do this for two reasons. First, we believe that abduction will ultimately prove to be more efficient computationally than a direct implementation of an argumentation system. Second, we have made progress on the development of a formal semantics for our abductive approach, which we sketch later in the paper. Thus, to the extent that we can provide a mapping from the argumentation scheme of KP to our abductive model, we can provide a semantic grounding for both.

Abduction is the process of reasoning from some observations to the best explanation for them. Various formalisms for abductive reasoning have been proposed in the recent AI literature [Lev89, Poo89a, Reg83]. In this paper, we focus on a particular kind of abduction, *weighted abduction*, and explore its application to the problem of reasoning about an agent's mental state. Weighted abduction involves the use of inference weights to compare completing explanations; a central theme of this paper is the use of this weighting mechanism in selecting the mental state that best explains a user's observed actions. More specifically, we show how KP's model of plan ascription can be recast in the framework of weighted abduction, and discuss the potential advantages and disadvantages of this encoding.

We do not provide a new theory of plan ascription in this paper. Our aim is not to define particular inference rules that a plan ascription theory should include, nor is it to specify preferences among specific rules. Rather, we develop a scheme that can be used for encoding rules for plan ascription and preferences for their application, a scheme that can be readily used in

[2]Kautz suggests the introduction of an "error" plan that will be inferred whenever one of the assumptions is violated. But, in general, this is insufficient, since agents need to be able not only to reason *that* a plan is incorrect, but also to reason about what makes it so.

a computational system.

2 Approaches to Abduction

The concept of abductive reasoning dates back almost a century to the work of the philosopher C. S. Pierce [Pie55]. The abduction task can be described as follows: given some proposition ϕ, explain ϕ by finding a set of additional propositions that, along with background knowledge, account for ϕ. To put this somewhat more formally: given a theory $\mathcal{T}$ and a proposition ϕ, compute a set of consistent assumptions A, such that $T \cup A \models \phi$.

Many AI uses of abductive reasoning have involved diagnosis problems, and thus have treated abduction as a type of causal reasoning, requiring the elements of the assumption set A to be *causally* related to the ϕ. However, the general characterization just given dispenses with this requirement. This is important, because a causality assumption is often not appropriate to the task of mental-state ascription. For example, a system performing plan recognition may decide that the user believes some proposition P simply because the system itself believes P, even though there may be no direct causal connection between the system's belief and the user's.

Typically, a number of different sets of propositions may play the role of the assumption set A. For example, if $\{P\}$ is an assumption set for a given abduction problem, then so is $\{P \wedge Q\}$, where Q is any proposition that is consistent both with P and with the theory $\mathcal{T}$. More generally, whenever some abductive assumption set contains P, alternative assumption sets containing $P \wedge Q$ can also be constructed, subject again to the consistency constraint. Most abductive reasoning systems have incorporated some sort of "Occam's Razor" principle to reject the latter assumption sets in favor of the former. It would be nice if such a criterion could be formulated on strictly semantic grounds, ignoring any syntactic details about how the propositions P and Q are represented in the theory. However, Levesque has shown that this is impossible in general [Lev89]. Nevertheless, it is quite desirable that the evaluation criterion be as independent of syntactic details

as possible.

The evaluation of assumption alternatives turns out to be one of the central problems of abduction—just as the evaluation of alternative explanations for an observed action is one of the central problems of plan ascription. Proposals about how to evaluate competing sets of abductive assumptions have tended to fall into two classes: those that involve a global criterion, against which an assumption set as a whole can be evaluated, and those that involve local criteria, in which individual rules in the theory are assigned evaluative metrics. We consider each approach in turn, and comment on their usefulness to problems of reasoning about mental state.

2.1 Global Criteria

Probably the simplest evaluative criteria for abductive systems are cardinality comparisons, which were first introduced for use in diagnosis applications. In these applications, one views a system as being composed of a number of components, and specifies a base theory that describes the intended input/output behavior of the system. The diagnosis problem in this setting involves reasoning from observations of the system's erroneous behavior to a set of assumptions about which components are faulty: the assumption set should identify components whose individual failures, taken together, would explain the observed behavior of the system [Rei87]. This specification of the problem leads to a clear evaluation criterion: one should accept the assumptions that imply the failure of the smallest number of components. Although this criterion may be useful in diagnostic tasks—medical as well as mechanical—it is more difficult to apply in tasks such as mental-state ascription that lack enumerable underlying entities. One might, for example, attempt to count the number of "facts" assumed, where a "fact" is some minimal syntactic unit such as a literal, but this measure is extremely sensitive to the particular syntactic details of the theory, and very likely to lead to unintended results [NM89].

Poole suggests an alternative to the cardinality criterion, under which one prefers the *least presumptive explanation* [Poo89a]. Given a set of al-

ternative assumption sets $A_1, \ldots, A_n$ that are solutions to the abduction problem $T \cup A \models \phi$, assumption set A_i is less presumptive than assumption set A_j if $T \cup A_j \models A_i$. This corresponds rather directly to what Stickel refers to as *least specific abduction* [Sti88a]. Stickel argues that least specific abduction is a good evaluation criterion for certain types of natural-language interpretation tasks. For example, in ordinary discourse, if a speaker says "My car won't start," it is likely that what the speaker intends the hearer to believe is that the car won't start. It is less likely that the speaker intends for the hearer to try to determine that the speaker believes the battery is dead or the starter solonoid is defective or any of a multitude of other facts that would account for the car's failure to start. The least specific abduction is, in this case, the most appropriate one. On the other hand, least specific abduction is not an appropriate strategy for diagnosis tasks: if a diagnostic system were given the task of determining why some car won't start, it would need to entertain each of the possible causes for the problem and judge their relative plausibility. Here, the *most* specific assumption set that can be derived should be accepted.[3]

Tasks involving reasoning about mental state, such as plan ascription, incorporate aspects of both least specific and most specific abduction. As Quilici *et al.* point out, an observed action can sometimes be explained by a range of equally likely underlying beliefs [QDF88]. In this case, a cooperative plan recognition system need not entertain the entire set of alternatives, and attempt to reason further from that entire set; instead it should ascribe only the single, less specific belief that can be strongly supported by the evidence. In fact, this is the insight inherent in several strategies that have been proposed to control plan recognition, such as Allen's forking heuristic [All83] and Sidner's single-branch strategy [Sid85]. The single-branch strategy, for instance, applies when a plan-recognition system reaches a point at which it has assumed that the actor intends to perform some action α, and

[3]We naturally assume that the definition of an assumption set incorporates some minimality criterion that rules out assumptions that are irrelevant in the sense that they play no role in concluding the observation.

then determines that α may be part of several, presumably equally likely plans, say P_1, P_2 and P_3; the strategy specifies that in this situation, the system need not reason about the whole set of alternatives, but instead should ascribe an intention to perform α, and wait to see whether further information is provided that disambiguates among the $Pi's$.

While certain parts of plan ascription thus require a least specific abduction strategy, there are other parts for which such a strategy will be invalid. For example, suppose that in some circumstance, an observed action α is most likely to be a part of a particular plan P_1, which in turn is most likely to be a subplan of some P_2. As a concrete illustration, let α be the action of getting one's keys out, performed in the circumstances in which the actor is standing in front of her front door, and let P_1 be the intention to unlock the door, and P_2 be the intention to enter the house. P_2 provides a better explanation of the observed action than does P_1, even though P_1 is a less specific assumption, in the sense that (by hypothesis) entering the house entails unlocking the door.

We thus see that neither a most specific nor a least specific abduction strategy is completely appropriate for plan ascription. Some other alternative must be adopted.

2.2 Local Criteria

An alternative to global criteria for comparing competing assumption sets in abductive reasoning is to associate evaluative metrics with individual rules in the base theory $\mathcal{T}$ and then evaluate each assumption set A by combining the weights of the rules used to derive the members of the set. In this approach, one effectively associates each assumption with information about the likelihood that it is true. Unlike global criteria, which are generally domain independent, a localized approach allows one to incorporate domain-specific details about the likelihood that particular propositions are true.

Bayesean statistical methods provide the most rigorous means of incorporating information about likelihood in a theory. To apply Bayesian methods to an abduction problem, one must first completely delimit the

space of possible assumptions. Then each assumption must be assigned an *a priori* probability, and the conditional probabilities of consequences, given particular assumptions, must be determined. The abduction problem then is simply to use Bayes's Rule to compute the probability of the various assumptions being true, given the observation to be explained, and ultimately to accept the most probable combination of assumptions that jointly explain the observations. The statistical approach to abduction has been extensively used in systems for medical diagnosis [Pop82], and its use is being investigated in systems for natural-language understanding [CG88] and plan recognition [Cal91, RZ91]. The principal disadvantage of the statistical approach is that it requires one to derive an exhaustive partitioning of the assumption space into independent alternatives. Although this requirement may be surmountable in diagnostic tasks, in which the space of hypotheses can be clearly delimited and in which independence conditions make sense, it presents a serious burden to problems involving mental-state ascription, in which the range of alternatives and their independence seem much more difficult to establish.

As an alternative to statistical methods, we propose an approach to abduction based on model preferences. In this approach, the process of making an assumption during abductive reasoning is viewed as restricting the models of the background theory $\mathcal{T}$. We encode an underlying preference ordering on the set of the models of $\mathcal{T}$, using annotations on the rules of the theory. These annotations are expressed as numeric weights—hence the named *weighted abduction*. In weighted abduction, one abduction set, A_1, is better than an alternative, A_2, if A_1 restricts the models of $\mathcal{T}$ to a more highly valued subset than does A_2. Thus an optimal assumption set is taken to be the one that restricts the models of $\mathcal{T}$ to a more highly valued subset than does any competing assumption set.

A major advantage of weighted abduction is its flexibility. As we shall see, using weighted abduction, one can express not only domain-specific information about the likelihood that any particular proposition is true, but also preferences for most specific (diagnostic-type) explanations and for

least specific explanations in situations in which either of these approaches is warranted. In other words, within a single theory, one can state the domain-specific conditions that would lead one to prefer either a most specific or a least specific explanation.

3 Weighted Abduction

We now turn to a more detailed description of weighted abduction. The algorithm for weighted abduction was introduced by Stickel, but without any theoretical analysis [Sti88a]. The key idea behind weighted abduction is that the process of making an assumption during inference should incur a cost. Thus every conjunct in conjunct in each rule in a weighted abduction theory is assigned an assumption cost, and this cost is propogated through rules, from consequents to antecedents.

Thus, in weighted abduction, the background theory $\mathcal{T}$ consists of a set of literals (facts) and a set of rules of the following form:

$$p_1^{w_1} \wedge \ldots \wedge p_n^{w_n} \supset q.$$

Each rule is a Horn clause: an implication with a single consequent literal q, and a conjunction of antecedent literals p_i. Each antecedent literal is associated with a weighting factor w_i. The proposition to be explained, ϕ, is expressed as a conjunction of literals, each of which is associated with an assumption cost.

Given a goal of proving some proposition ϕ, a weighted-abductive theorem prover can either assume ϕ at its assumption cost, or it can find a rule whose consequent unifies with ϕ and attempt to prove the antecedent literals. The assumption cost of each subgoal—that is, each proof of an antecedent literal p_i—is computed by multiplying the assumption cost of the goal by w_i, the weighting factor associated with p_i. For example, given the rule above, if the assumption cost of q is w_q, then the assumption cost of p_1 is $w_q * w_1$, and the assumption cost of p_2 is $w_q * w_2$. Each antecedent literal can either be (1) assumed at its computed assumption cost, (2) unified with

a fact in the knowledge base (a "zero cost proof"), (3) unified with a literal that has already been assumed—the algorithm only charges once for each assumption instance, or (4) proved via the use of another rule. The best solution to the abduction problem is given by the set of assumptions that lead to the lowest cost proof.

A solution to an abduction problem is admissible only when all the assumptions made are consistent with each other and with the initial theory. Therefore, a correct algorithm must filter out potential solutions that rely on inconsistent assumptions. Another possibility that must be accounted for is that, in the frequent case in which the goal proposition ϕ and its negation are both consistent with the theory, it will be possible to prove both ϕ and $\neg\phi$ abductively—in the worst case by assuming them both outright. The abduction algorithm thus must guarantee that it is impossible to *defeat* a proof by proving the negation of any of its assumptions at a cost that is cheaper than the cost of the proof itself.

The complete abduction algorithm can be described as follows:

Main Procedure:
Given a base theory $\mathcal{T}$ and a goal ϕ
    Set *Solution* = none;
    Generate all candidate assumption sets, *Cands*;
    Sort the members of *Cands* in order of increasing cost;
    While *Solution* = none **and** *Cands* $\neq$ nil
        Set A = first member of *Cands*;
        Set *Cands* = rest of *Cands*;
        If Admissible(A) [Use subroutine below.]
            Then Set *Solution* = A
        EndIf
    endWhile;
    Return *Solution*.
Subroutine for Determining Admissibility:
Given an assumption set $A = \{\psi_1, \ldots, \psi_m\}$
    Set *Admissible* = true;

Set $i = 1$;
While $i \leq m$ **and** *Admissible* = true
    Attempt to prove $\neg\psi_i$, given assumptions $\psi_1, \ldots, \psi_{i-1}, \psi_{i+1}, \ldots, \psi_m$;
    If $\neg\psi_i$ is provable with other no assumptions
        Then Set *Admissible* = false A is inconsistent]
    EndIf;
    If $\neg\psi_i$ is provable by making additional assumptions
        Then If the best proof costs less than the cost of A
            Then Set *Admissible* = false [A is defeated]
        EndIf
    EndIf;
    Set $i = i + 1$
EndWhile;
Return *Admissible*.

It is possible to express different abduction strategies through the selection of suitable weighting factors. Consider a simplified instance of the schematic rule shown above, in which there are only two conjuncts:

$$p^{w_1} \wedge q^{w_2} \supset r.$$

If $w_1 + w_2 < 1$, the rule will favor the assumption of p and q as explanations for r. This is effectively an encoding of most-specific abduction, since the cost of assuming both p and q is less than the cost of assuming r.

In contrast, when $w_1 + w_2 > 1$, least-specific abduction is favored, as it will then be less expensive to assume r than to assume p and q. However, when $1 \leq w_1 + w_2 < 2$, then although both p and q cannot be simultaneously assumed to prove r, if either one can be derived at zero, or very low, cost, then the other can be assumed to prove r. If, for instance p can be matched against a fact in the database, it then provides the evidence one needs to assume q.

The actual assignment of weighting factors to the rules in an abductive theory is a difficult question and the subject of ongoing research. We shall

say something more about it in the next section.

3.1 Computational Considerations

Should this section go at the end of the paper?

It is impossible to escape the observation that abductive reasoning is computationally hard. In particular, weighted abduction presents two computationally difficult problems: the first problem is determining a minimal cost candidate set, and the second is guaranteeing that any particular candidate set is consistent with the theory. Existing computational results in this area are not particularly encouraging. Selman, [Sel90] for example, proves that, even in the case of propositional horn clause theories, the problem of computing an explanation for an arbitrary proposition is NP hard, given some modest restrictions on what counts as an explanation. In the first-order case, explanation is undecidable in general.

Although the theoretical results are indeed pessimistic, it is important to consider the question of how well abduction can be applied in practice before rejecting the method as a viable approach. Part of the computational problem is a consequence of the fact that the problem of plan recognition is inherently hard, which means that certain computationally intractable problems have to be faced, no matter what method is chosen. Plan recognition necessarily involves drawing default conclusions about an agent's mental state based on certain premises. Because these conclusions are defaults, one must check whether the default is applicable in the particular instance of its application, and regardless of the representation and reasoning framework chosen, this requirement imposes a computational cost that must be shared by all approaches.

Checking the consistency of the assumption set is one key source of computational difficulty. Although the weighted abduction algorithm described in Section 3 assumes a complete consistency check, we believe that it is most advisable in practice to settle for an incomplete consistency check that can be computed relatively quickly, and that for a particular domain of application is capable of detecting most of the inconsistencies that are likely to

arise. For example, the TACITUS text understanding system [HSME88] relies on the fact that in the text-understanding domain, most inconsistencies result from the incorrect identification of two individuals that are actually distinct. This incorrect identification frequently results in the violation of predicate-argument type constraints among the literals in the assumption set. Therefore, most inconsistencies can be detected by checking variable typing constraints implied by the assumed literals — an operation that can be carried out at relatively little computational cost.

The role of identity assumptions appears to play a much smaller role in plan recognition than in text understanding, and therefore the heuristic applied in TACITUS offers few advantages in this domain. However, it is possible to exploit the nature of the plan-recognition and evaluation domain to limit the search required for the consistency check.

Plan recognition: assumptions about attitudes, certain attitudes are known in advance to be inconsistent: Bel(A, P) and Ach(A, P), Int(a, To(a, P)) and Bel(a, P), Int(a, By(a1, a2, P)), Bel(A, not P). But we haven't introduced the ARGH notation yet.

Another compuational problem arises in the computation of the candidate assumption sets. The problem results from the fact that the cost associated with an assumption set that does not consist entirely of pure literals cannot be guaranteed to be minimal. Suppose A_1 and A_2 are two sets of assumptions such that the cost of A_1 is less than the cost of A_2. If we were to stop computation at this point, it would be nice if we could guarantee that the best explanation would entail the literals of A_1. However, if some of the literals of A_2 are not pure, it is possible that further computation would reveal that some set of assumptions entailing A_2 is preferred. It is, in general, impossible to guarantee that a particular explanation will be entailed by the best explanation without computing all possible explanations.

In typical plan recognition problems, the set of candidate assumptions can be extremely large. This is particularly true in situations in which one wishes to admit the possiblity that the user's plan can result from misconceptions. If one is willing to entertain almost any bizzarre misconception,

the problem quickly becomes unmanagable, but one must be willing to entertain *some* misconceptions, or it is impossible to recognize or evaluate an incorrect plan. [Need to elaborate on this a bit]

This paper doesn't solve the knowledge engineering problem. Any plan ascription system will need to come up with some way of ordering rules. What we provide is a reasonable framework for encoding these rules, one that is formally grounded. here are two problems with arriving at weight assignments for a weighted abduction theory. The first problem is that rules are the vehicles by which preferences are stated. It is therefore difficult to first think of a preference, and then encode that preference as weighting factors, because the particular set of rules chosen to express the facts of the domain may not provide the right collections of literals for attaching the weights. The second problem is that the introduction of numeric weightings on a particular rule can have a non-local effect on the set of preferences as a whole. It is necessary to consider the preferences introduced by each rule in connection with all the other rules in the theory. This encoding process can be difficult, but once it is complete, it can be successfully applied computationally.

3.2 A Model-Theoretic Interpretation

We can sketch a semantics for weighted abduction that is based on model preference. A complete discussion of the semantics of weighted abduction would be beyond the scope of this paper, but understanding the principles behind the semantics is important for understanding precisely what the application of the weighting factors means. For any automated reasoning system that employs numeric techniques to express concepts of likelihood or adjudicate among alternatives, it is reasonable to ask "what do the numbers mean?" In this section we provide a formal definition of the abduction problem that corresponds to the weighted abduction algorithm described in the previous section.

The central idea is inspired by the model preference default logics of Selman and Kautz [SK89]. If $\mathcal{T}$ is a base theory to be used in weighted

abduction problems, we assume that there is an underlying preference order on the models of $\mathcal{T}$. The weights on the rules in $\mathcal{T}$ are interpreted as expressing implicit constraints on this preference order. For example, consider a rule $p^{\alpha} \supset q$, which is satisfied in any model that satisfies q. If $\alpha < 1$, this suggests that those models that satisfy p, and so in which the truth of q is in some sense "explained," are preferred to those models satisfying q but not p. It is too restrictive to interpret the rule as expressing the preference "*every* model that satisfies $p \wedge q$ is preferred to *every* model that satisfies $\neg p \wedge q$." In many practical situations, such a restrictive preference is impossible to satisfy. The proper interpretation of the rule is "There is *some model* satisfying $\neg p \wedge q$ that is less preferred than *every* model satisfying $p \wedge q$.

Under this general scheme, an abductive proof of a goal ϕ from an initial base theory $\mathcal{T}$ consists of seeking a set of assumptions A, such that the models of $\mathcal{T} \cup \mathcal{A}$ are all preferred to certain less-preferred models of $\mathcal{T} \cup \{\phi\}$ that do not satisfy $\mathcal{T} \cup \mathcal{A}$. Thus, weighted abduction can be viewed as a reasoning process that computes theories with successively better greatest-lower-bounds on the preference relation of their associated models by adding restrictions to the base theory.

Given a theory $\mathcal{T}$, a total, antireflexive, antisymmetric preference relation $\prec$ on models of $\mathcal{T}$ and an observation ϕ, an abduction problem consists in deriving a set of assumptions A that satisfies the following conditions:

1. **Adequacy.** $\mathcal{T} \cup A \vdash \phi$
2. **Consistency.** $\mathcal{T} \cup A \nvdash \neg\phi$
3. **Syntactic Minimality.** If $\psi \in A$ then $\mathcal{T} \cup A - \{\psi\} \nvdash \phi$
4. Semantic Greatest Lower Bound There is no assumption set A' such that
   (a) $\mathcal{T} \cup A'$ is adequate, consistent, and syntactically minimal, and
   (b) There exists $M \models \mathcal{T} \cup A$ such that for every $M' \models T \cup A\prime$, $M' \prec M$

5. **Defeat Condition** There is no set A'' such that there is some $\psi \in A$ such that $\mathcal{T} \cup A'' \vdash \neg\psi$ and there is some model $M \models \mathcal{T} \cup \mathcal{A}$ such that for every model $M'' \models \mathcal{T} \cup A'', M'' \prec M$

The role of the adequacy and consistency requirements of this definition should be obvious. Because it may be possible to restrict the models of a theory to a favored subset by making assumptions that have nothing to do with the observation, the syntactic minimality condition imposes the requirement that every assumption actually contribute to the solution of the problem. The greatest lower bound condition guarantees that the assumption set that constitutes the solution to the problem is one that is maximally preferred. The defeat condition says that it must not be possible to find an assumption set that entails the negation of any of its elements at a cost that is less than that of the assumption set itself.

To relate the model preference relation to the numeric weights employed by the weighted abudction algorithm, we assume that there is a function ξ, called the *model evaluation function* that maps models of $\mathcal{T}$ into the set of rational numbers. We say that model $M_1 \prec M_2$ if and only if $\xi(M_1) < \xi(M_2)$. We assume that $\mathcal{T}$ consists of literals together with rules of the form

$$p_1^{w_1} \wedge p_2^{w_2} \wedge \ldots \wedge p_n^{w_n} \supset q$$

such that each p_i is a literal and w_i is an associated numeric weighting factor.

Suppose that R is a rule of the form indicated above with n antecedent literals. A model preference constraint is introduced for each subset S_i of antecedent literals of R. For each S_i, if there exists a model M_1 such that

$$M_1 \models q \wedge \bigwedge_j p_j \wedge \bigwedge_k \neg p_k, \text{such that} p_j \in S_i, p_k \notin S_i,$$

then, for every model M_2 such that

$$M_2 \models \bigwedge_n p_n$$

it is the case that

$$\xi(M_2) < \xi(M_1) \sum_{p_m \notin S_i} w_m.$$

As an example of the application of this definition, consider the simple rule

$$p^{0.5} \wedge q^{0.8} \supset r.$$

According to the definition, this rule introduces 3 constraints on the model evaluation function:

(1) For some $M_1 \models r \wedge \neg p \wedge \neg q$, for all $M_2 \models p \wedge q$, $\xi(M_2) < 1.3\xi(M_1)$
(2) For some $M_1 \models r \wedge p \wedge \neg q$, for all $M_2 \models p \wedge q$, $\xi(M_2) < 0.8\xi(M_1)$
(3) For some $M_1 \models r \wedge \not p \wedge q$, for all $M_2 \models p \wedge q$, $\xi(M_2) < 0.5\xi(M_1)$

It is easy to see by these preferences, that if either p or q is derivable in $\mathcal{T}$, then if r is a goal, it will be preferable to assume either p or q than to assume r. However, if neither p nor q are derivable, we cannot conclude on the basis of this information alone that it is preferable to assume both p and q, than to assume r.

The weighted abduction algorithm presented in Section 3 is designed to find solutions to an abduction problem given a weighted abduction theory, and the solution criterion presented above. We rely on the soundness of the basic theorem prover to guarantee the adequacy criterion. The consistency criterion is met through the proof of the negation of each assumption. The numeric weight assigned to the initial observation represents the greatest lower bound on the preference relation for all models satisfying the observation. The numeric weightings computed for each assumption set correspond to the most conservative estimates of the greatest lower bound for the preference relations implied by the rules that were used to construct it. The weighted abduction algorithm includes a specific check for defeat. This defeat check is required to guarantee the existence of the model M_1 that satisfies both the conditions imposed by the preference contstraint specification and the rest of the assumptions in the set.

4 Application to Plan Ascription

Having sketched the theory of weighted abduction, we now show how it can be applied to the problem of plan ascription. We begin by briefly reviewing

KP's model [KP89], and then recast that model using a weighted abduction approach.

4.1 A Model of Plan Ascription

**** Throughout, fix to distinguish pred's from functions.*

In KP's model of plan ascription, like many of the earlier ones, the process of plan ascription consists in ascribing to an agent a set of beliefs and intentions—a *plan*—that explains an action or actions that the agent is observed to perform. The "building blocks" that KP employ to define this process are predicates that describe [aspects of] an agent's mental state, as well as plan fragments:

Mental State

INT(a, α)**:** agent a intends α

BEL(a, p)**:** agent a believes p

ACH(a, p)**:** agent a believes p will become true as a consequence of the actions he performs

EXP(a, p)**:** BEL$(a, p) \vee$ ACH(a, p)

Plan Fragments

$TO(\alpha, p)$**:** the plan consisting of doing α to make p true

$BY(\alpha, \beta, p)$**:** the plan consisting of doing β by doing α, while p is true (p "enables" the relation)

Details of the representation language can be found in KP's paper. Here we rely largely on the reader's intuitions about their intended meanings.

In KP's model, the observer ascribes plan fragments to the actor until a globally coherent plan that can account for all his observed actions is found. The ascription process is controlled by a direct argumentation system, ARGH [Kon88]. ARGH is a formal system, in the sense that its

elements are formal objects, and the processes that manipulate them could be implemented on a computer. It is similar in many respects to so-called justification-based Truth Maintenance Systems [Doy79], but differs in the diversity of argumentation allowed, and the fact that arguments for a proposition and its negation may coexist without contradiction. It also differs from formal nonmonotonic logic approaches, such as circumscription or default logic, in that it makes arguments the direct subject matter of the system. Finally, it differs from other direct argumentation systems in that it has an explicit notion of argument *support* independent of belief, and allows a flexible specification of domain-dependent conditions for adjudicating among arguments.

The purpose of argumentation is to formulate connections between propositions, so that an agent can come to plausible conclusions based on initial data. Formally, an argument is a relation between a set of propositions (the *premises* of the argument), and another set of propositions (the *conclusion* of the argument). Using ARGH, one can state support relations between premises of an argument and conclusions, and one can also state defeat rules, which express the relative strengths of potentially conflicting arguments. Defeat is what makes arguments defeasible, and is one of the most complicated and interesting parts of defining a domain. When we recast the ARGH framework using weighted abduction, the defeat principles are captured using the weighting mechanism.

The most important rules in KP's plan recognition scheme are those that are used to ascribe plan fragments, such as the following example:

$$\begin{aligned}&\mathrm{BEL}(a,\, TO(\alpha, p)), \mathrm{INT}(a, \alpha), \mathrm{ACH}(a, p) \xrightarrow{to} \\ &\mathrm{INT}(a,\, TO(\alpha, p))\end{aligned}$$

This rule (actually, rule schema) says that, if an agent a believes that p is an effect of performing α, and he intends to do α and to achieve p, then it is plausible that his reason for doing α is to achieve p. A similar rule is used

to coalesce fragments involving the BY relation:

$$\mathrm{BEL}(a, BY(\alpha, \beta, p)), \mathrm{INT}(a, \alpha), \mathrm{INT}(a, \beta), \mathrm{EXP}(a, p) \xrightarrow{by} \mathrm{INT}(a, BY(\alpha, \beta, p))$$

Additional rules, which we shall not repeat here, extend ascribed plan fragments; an example of such a rule states that if an agent believes that p is an effect of performing α, and he intends to do α, then he both intends to do α in order to make p true (that is, he intends the TO fragment) and plans to achieve p. Such rules correspond closely to the classical plan-recognition rules in a system such as Allen's [All83].

As noted earlier, the second important set of rules in the system are the defeat rules, which express the relative strength of arguments. One example is *Purposeful Action Defeat*. This rule encodes the presumption that agents engage in purposeful actions: they do not typically intend actions whose effects they already believe to be true. In the ARGH system, this rule is stated as follows:

Purposeful Action Defeat If $x \xrightarrow{to} Ach(a, p)$ is an argument whose premises x are supported, and so is $y \xrightarrow{belasc} Bel(a, p)$, then the belief ascription argument (the one labeled *belasc*) is defeated.

To understand *Purposeful Action Defeat*, assume that there is an argument that supports the conclusion that the agent intends to perform some action to bring about p; this argument makes use of the *to* rule schema presented above. Assume further that there is another argument that supports the conclusion that the agent believes that p will hold independent of his actions, and that this argument is based only on the reasoner's own beliefs that p will hold. Although we have not provided the belief transfer rule here, it is captured in KP's system by a *belasc* rule. *Purposeful Action Defeat* specifies that the former argument, the one using the *to* rule, defeats the latter argument, the one using *belasc*. Note that *Purposeful Action Defeat* applies only to arguments in a belief that p is attributed on the basis of simple belief ascription: the observer believes p and therefore it's plausible

that the actor does, too. This is the only kind of ascription countenanced by *belasc*. There may well be additional, stronger evidence that the observer believes p, in which case arguments using that evidence—by means of other ascription rules—can defeat the argument using the *to* rule.

Below, we shall consider some other defeat rules introduced by KP, illustrating their use in an example. First, however, we consider how the argumentation-system approach can be recast using weighted abduction.

4.2 Applying Weighted Abduction

Our task is to convert both the support rules and the defeat rules into the weighted abduction scheme. Conversion of the support rules is straightforward. For each support rule, we simply conjoin all the propositions of the left-hand side, that is, the premises, into a single conjunctive proposition, and then form a rule in which the conjunction entails the right-hand side, that is, the conclusion, of the support relation. For example, the *to* rule schema above becomes:

$$\forall a, \alpha, p\, \mathrm{BEL}(a, TO(\alpha, p)), \mathrm{INT}(a, \alpha), \mathrm{ACH}(a, p) \supset \mathrm{INT}(a, TO(\alpha, p))$$

Before adding this rule to the background theory, we must assign appropriate weighting factors to it. To determine the weighting factors, we consider the relationship between this rule and other rules in the system that might lead to conflicting results.

To illustrate this process of assigning weights, consider the Purposeful Action Defeat. In the weighted abduction framework, what we do is to assign a lower weighing factor to the premise of the abduction rule that is derived from the *to* rule, than to the premise of the rule that is derived from the *belasc* rule. This will guarantee that abductive inferences using the former rule will be preferred to those using the latter.

Most of the potential defeat rules can be handled in this fashion, by assigning appropriate weightings. However, two of the defeat rules given by KP are in fact handled directly by the weighted abduction system. *Initial*

Fact Defeat states that an initially known fact defeats any argument supporting a conflicting proposition; this is handled directly by the abduction system's requirement that the assumption set be consistent. Of course, for this to work, conflicting propositions must be properly defined. Thus, for example, KP specify that BEL(a, p) and ACH(a, p) conflict—because one cannot believe both that a particular proposition will be made true by some action he intends to perform and that it will also be true independent of his actions. These propositions must thus be defined to be logically inconsistent in the base theory of the abductive system.

A second defeat rule, *Conflicting Action Defeat*, handles cases in which there are arguments leading to conclusions for two intentions, to do two different actions at the same time, each of which has the same effect. *Conflicting Action Defeat* states that in such cases, if the *support set* of one of the intentions is a proper superset of the support set of the other, then the argument for the former defeats the argument for the latter. In ARGH, a support set of a proposition p consists of those initial facts that are used in some argument chain that supports p. Conflicting action defeat captures the intuition that, when there are competing alternative actions and one of them is part of a coherent set of ascribed plan fragments while the other is not, we prefer to ascribe an intention to do the former. In the abductive model, this corresponds to the assumption set for the former proposition being a proper subset of the assumption set for the latter; because we do not permit negative weights, it follows from this that the weight of the former set will be less than the weight of the latter set. Thus, the former assumption set will be preferred.

4.3 Example

In this section, we provide a brief example that illustrates the application of weighted abduction to plan ascription. As we noted earlier, we distinguish between two closely related types of plan ascription problems: plan recognition and plan evaluation. They differ in the type of information that is given as part of the problem statement, and the type of information that is sought

as a solution. In plan recognition, the information provided is a set of one or more observed actions; the task is to find some plan that explains those actions. In plan evaluation, the given information also includes a statement of the actor's goal; the task then is to find some plan that relates the observed actions to the known goal. As we shall see, in an abductive approach, this difference has an important influence on the specification of the three problem components: the base theory, $\mathcal{T}$, the goal proposition ϕ, and the assumption set A.

We first consider plan recognition. In an abductive statement of a plan recognition problem, the base theory $\mathcal{T}$ contains the observer's existing beliefs about the actor's mental state, along with general beliefs about the domain. The goal proposition ϕ consists of the conjunction of intentions to perform each of the observed actions. (A more complete statement of the problem might also require an inference from each observation of an action to an assumption that it was intentional; however, we simplify our discussion by presuming that this default inference is automatic. Similarly, when we consider intentions that are described in a discourse, we assume an automatic inference from the description to the intention itself.) Given this much, a solution to the problem is an assumption set A that ascribes a plan to the actor, such that the intentions of ϕ are entailed by A and $\mathcal{T}$ together. Notice that the plan ascribed is a mental state, that is, it typically consists of some set of beliefs and intentions satisfying various coherence constraints that are defined in the theory.

In plan evaluation, the actor's goal is also known to the observer.[4] It seems natural to formalize plan evaluation in a similar manner to plan recognition: the only difference would be that in the case of plan evaluation there is an additional proposition to be conjoined in ϕ, namely, an intention to perform the given goal. Unfortunately, this treatment does not sufficiently constrain the problem. In particular, it does not require the observed ac-

[4]Examination of natural-language discourse has shown that speakers often tell their hearers what their goals are, presumably to facilitate the plan ascription process [Pol86]. Thus, plan evaluation is a natural problem to study.

tions to be seen as intended components of a plan to achieve the given goal. Consider a situation in which the observed action is α, and the given goal is P. The actor, unbeknownst to the observer, erroneously believes that doing α brings about P. If, however, α actually brings about Q, a solution to the problem as stated could account for α as a way of achieving Q, and account for P independently. There is nothing that forces the consideration of P as the goal towards which α is aimed.

To avoid this problem, we adopt a different encoding of the plan evaluation problem, in which the intentions to perform the observed actions are taken to be part of the base theory $\mathcal{T}$, and the known goal of the actor is defined to be ϕ. Then the inference rules in $\mathcal{T}$ that are used to generate the members of candidate assumption sets A can take into account the intended actions and relate them to the known goal. In the example described in the previous paragraph, a base rule can be used that relates intended actions with the effects that the actor believes they have. This will lead to an ascription both of the erroneous belief (that α leads to P) and of the consequently ill-fated intention (to perform α as a way of bringing about P). In general, assumptions that are generated using rules that rely on the agent's intentions will be highly valued.

We illustrate these ideas with a simple plan evaluation problem, derived from [KP89], in which a robot, Flakey, is asked by another agent, Harry, to get some particular report for itself (Flakey). This might come about through a request like, "I want you to get the report, so you will have it," in which the intended referent for "the report" is easily recoverable. This request might sensibly be made with the ultimate intention of getting Flakey to deliver the report to a third party, but that will not concern us here. Assume that, unbeknownst to Harry, Flakey already has the report. What can Flakey conclude about Harry's mental state from this request? We shall set up this problem by specifying a fragment of a base theory, from which we will attempt to abduce that Harry intends that Flakey get the report, believing that this will bring it about that Flakey has the report.

According to the discussion above, the base theory contains initial facts

about the domain, including Harry's described intentions. These facts are expressed by axioms (1) through (3):

$$\forall x, y\, \mathrm{To}(\mathrm{Get}(x, y), \mathrm{Has}(x, y)) \tag{1}$$

$$\mathrm{Has}(\mathrm{Flakey}, \mathrm{Report}) \tag{2}$$

$$\mathrm{INT}(\mathrm{Harry}, \mathrm{Get}(\mathrm{Flakey}, \mathrm{Report})) \tag{3}$$

Axiom (1) represents Flakey's belief that the result of getting an object is having it. Axiom (2) represents Flakey's belief that it already has the report, and Axiom (3) represents the initial observation about Harry's intention, as expressed by his request.

Recall from Section 4.1 that we also need an axiom to represent the inconsistency of BEL and ACH:

$$\forall a\, \mathrm{BEL}(a, P) \supset \neg \mathrm{ACH}(a, P) \tag{4}$$

The following two axioms provide a very simple theory of belief ascription. We assume one agent can ascribe certain beliefs to another agent, and can therefore reason about what the latter agent can conclude; we further assume that an agent can use its own beliefs as a model to fill gaps where direct information about the other agent's beliefs is incomplete.

$$\forall a\, P \wedge \mathrm{Common}(a, P) \wedge \mathrm{Shared}(a, P)^{0.1} \supset \mathrm{BEL}(a, P) \tag{5}$$

$$\forall a\, P \wedge \mathrm{Private}(a, P) \wedge \mathrm{Shared}(a, P)^{0.9} \supset \mathrm{BEL}(a, P) \tag{6}$$

$$\forall a, \alpha\, \mathrm{Common}(a, \mathrm{To}(\alpha, P)) \tag{7}$$

$$\forall a, x, y\, \mathrm{Private}(a, \mathrm{has}(x, y)) \tag{8}$$

$$\forall a\, \text{Common}(a, P) \equiv \neg\text{Private}(a, Q) \tag{9}$$

Axiom (5) states if Flakey believes a proposition, which, moreover, is thought to be common, then it can be concluded that the actor also believes that proposition. Indeed, precisely because the proposition is thought to be common, it is quite likely that other agents believe it; this is reflected in the rather low assumption weight (0.1) associated with the "Shared" predicate. Axiom (6) is similar to Axiom (5), except that it applies to beliefs that are thought to be private: because such beliefs are less likely to be shared, the associated assumption weight (0.9) is much higher. Axioms (7) and (8) specify certain propositions as being common or private: Flakey believes that everyone with whom it interacts knows about the effects of domain actions, but Flakey does not believe that everyone knows what objects everyone else has. Axiom (9) specifies that no propositions are thought to be both private and common: these are incompatible predicates.

Finally, we need the following axiom, which restates the first plan-ascription rule described in Section 4.1.

$$\forall a, \alpha\, \text{BEL}(a, To(\alpha, P))^{0.9} \wedge \text{INT}(a, \alpha)^{0.5} \wedge \text{ACH}(a, P)^{0.5} \supset \text{INT}(a, \text{To}(\alpha, P)) \tag{10}$$

Axiom (10) captures the relationship between an intention to achieve P by doing α, on the one hand, and the belief that α brings about P, the intention to do α, and the intention to achieve P, on the other. The high assumption weight associated with the BEL predicate reflects the implausibility of the actor's doing α to achieve P without believing that α achieves P. The weights on the INT and ACH literals reflect the intuition that evidence is required for the existence of the relevant goals and intentions—while it may be reasonable to assume one or the other, *both* of them will not be assumed.

In the current example, Flakey is told that Harry wants to perform the get action in order to bring it about that it has the report. Following the

procedure outlined for plan evaluation, we thus need to abduce

$$\text{INT}(\text{Harry}, \text{To}(\text{Get}(\text{Flakey}, \text{Report}), \text{Has}(\text{Flakey}, \text{Report})))$$

given the set of axioms listed above. We assign an arbitrary initial assumption cost of 100 to the goal to be proved.

Axiom (10) unifies with the goal proposition, and leads to three subgoals:

$$\text{BEL}(\text{Harry}, \text{To}(\text{Get}(\text{Flakey}, \text{Report}), \text{Has}(\text{Flakey}, \text{Report}))) \text{ assumable at cost } 90$$

$$\text{INT}(\text{Harry}, \text{Get}(\text{Flakey}, \text{Report})) \text{ assumable at cost } 50$$

$$\text{ACH}(\text{Harry}, \text{Has}(\text{Flakey}, \text{Report})) \text{ assumable at cost } 50$$

The INT subgoal matches Axiom (3) directly, so it is proved and thus need not be assumed. The ACH subgoal does not match any axioms in our limited set, and therefore must be assumed at a cost of 50 (that is, 100, the assumption cost of the goal proposition, times .5, the weight of the relevant predicate). There is no information in our tiny knowledge base that directly applies to Harry's beliefs, but the simple belief transfer theory we presented above can be used to conclude that Harry's beliefs about the effects of performing a get action are the same as Flakey's. In particular, the BEL subgoal unifies with the consequent of Axiom (5). Two of the three antecedents then direct match facts in the knowledge base: Axiom (1) expresses Flakey's belief about the effect of the get action, and Axiom (7) expresses its belief that this is likely to be commonly known. The remaining subgoal,

$$\text{Shared}(\text{Harry}, \text{To}(\text{Get}(\text{Flakey}, \text{Report}), \text{Has}(\text{Flakey}, \text{Report})))$$

is assumed at cost 9 (90 * .1), for a total proof assumption cost of 59. Our abductive proof therefore yields a solution that suggests that, if we know that Harry wants Flakey to get a report, and we assume that Harry believes that Flakey's getting a report results in its (Flakey's) having it, and we further assume that Harry has the goal of Flakey having the report, then

we can conclude that Harry wants Flakey to get the report so that it will have it.

We are not quite finished yet, because we still need to check the assumption set for consistency with the base theory. We do this by attempting an abductive proof of the negation of each assumption in turn, with a cost equal to the initial, arbitrarily chosen cost for the main proof. The attempt to prove the negation of the shared belief assumption at any cost less than than the cost of assuming it outright fails, so this assumption is consistent with the database. The situation is more complicated for the proof of

$$\neg \mathrm{ACH}(\mathrm{Harry}, \mathrm{Has}(\mathrm{Flakey}, \mathrm{Report})).$$

This negated assumption unifies with the consequent Axiom(4), and the resulting BEL subgoal can be proved by using Axioms (6), (2), and (8), and assuming, at a cost of 90, that

$$\mathrm{Shared}(\mathrm{Harry}, \mathrm{Has}(\mathrm{Flakey}, \mathrm{Report})).$$

The further recursive attempt to prove the negation of this assumption fails, and therefore we have found that it is possible to attribute to Harry the belief that Flakey already has the report; this then could potentially defeat the argument that he has the goal of Flakey obtaining it. However, note that the assumption cost of this refutation is greater than the assumption cost of the proof. We therefore reject the refutation in favor of the original proof. In the ARGH formulation, we noted that the ascription of a belief that resulted from the use of the *belasc* rule is defeated by any inconsistent ascription that resulted from the use of the *to* rule. We have represented this same defeat condition within the weighted abduction framework by a choice of weighting factors that establishes a preference for the consistency of the actor's goals and associated actions over the consistency between the observer's beliefs and those of the actor.

5 Computational Considerations

It is impossible to escape the observation that abductive reasoning is computationally hard. In particular, weighted abduction presents two computationally difficult problems: the first problem is determining a minimal cost candidate set, and the second is guaranteeing that any particular candidate set is consistent with the theory. Existing computational results in this area are not particularly encouraging. Selman, [Sel90] for example, proves that, even in the case of propositional horn clause theories, the problem of computing an explanation for an arbitrary proposition is NP hard, given some modest restrictions on what counts as an explanation. In the first-order case, explanation is undecidable in general.

Although the theoretical results are indeed pessimistic, it is important to consider the question of how well abduction can be applied in practice before rejecting the method as a viable approach. Part of the computational problem is a consequence of the fact that the problem of plan recognition is inherently hard, which means that certain computationally intractable problems have to be faced, no matter what method is chosen. Plan recognition necessarily involves drawing default conclusions about an agent's mental state based on certain premises. Because these conclusions are defaults, one must check whether the default is applicable in the particular instance of its application, and regardless of the representation and reasoning framework chosen, this requirement imposes a computational cost that must be shared by all approaches.

Checking the consistency of the assumption set is one key source of computational difficulty. Although the weighted abduction algorithm described in Section 3 assumes a complete consistency check, we believe that it is most advisable in practice to settle for an incomplete consistency check that can be computed relatively quickly, and that for a particular domain of application is capable of detecting most of the inconsistencies that are likely to arise. For example, the TACITUS text understanding system [HSME88] relies on the fact that in the text-understanding domain, most inconsistencies result from the incorrect identification of two individuals that are actually

distinct. This incorrect identification frequently results in the violation of predicate-argument type constraints among the literals in the assumption set. Therefore, most inconsistencies can be detected by checking variable typing constraints implied by the assumed literals — an operation that can be carried out at relatively little computational cost.

The role of identity assumptions appears to play a much smaller role in plan recognition than in text understanding, and therefore the heuristic applied in TACITUS offers few advantages in this domain. However, it is possible to exploit the nature of the plan-recognition and evaluation domain to limit the search required for the consistency check.

For example, certain attitudes are known in advance to be inconsistent. Examples of inconsistent attitudes are BEL(A, P) and ACH(A, P), or INT(A, To(A, P)) and BEL(a, P), or INT(A, By(a1, a2, P)) and BEL(A, ¬P). A quick examination of the knowledge base for co-occurances of these predicates can eliminate certain assumtpions whose inconsistency would be complicated to derive from basic principles. Also, spatial and temporal constraints on actions provide constraints that can be exploited. For example, we assume that an agent cannot perform multiple primitive actions at the same time in different locations.

Another compuational problem arises in the computation of the candidate assumption sets. The problem results from the fact that the cost associated with an assumption set that does not consist entirely of pure literals cannot be guaranteed to be minimal. Suppose A_1 and A_2 are two sets of assumptions such that the cost of A_1 is less than the cost of A_2. If we were to stop computation at this point, it would be nice if we could guarantee that the best explanation would entail the literals of A_1. However, if some of the literals of A_2 are not pure, it is possible that further computation would reveal that some set of assumptions entailing A_2 is preferred. It is, in general, impossible to guarantee that a particular explanation will be entailed by the best explanation without computing all possible explanations.

In typical plan recognition problems, the set of candidate assumptions can be extremely large. This is particularly true in situations in which one

wishes to admit the possiblity that the user's plan can result from misconceptions. If one is willing to entertain almost any bizzarre misconception, the problem quickly becomes unmanagable, but one must be willing to entertain *some* misconceptions, or it is impossible to recognize or evaluate an incorrect plan. The solution to this problem is to use the weighting factors to guide the search. If the weighting factor on a subgoal is high, one should attempt to prove the subgoal without making any assumptions. If this fails, then the entire line of reasoning should be abandoned until all other possiblities have been considered and rejected or found to also result in expensive assumptions.

Finally, this paper doesn't solve the knowledge engineering problem. Any plan ascription system will need to come up with some way of ordering rules. What we provide is a reasonable framework for encoding these rules, one that is formally grounded. There are two problems with arriving at weight assignments for a weighted abduction theory. The first problem is that rules are the vehicles by which preferences are stated. It is therefore difficult to first think of a preference, and then encode that preference as weighting factors, because the particular set of rules chosen to express the facts of the domain may not provide the right collections of literals for attaching the weights. The second problem is that the introduction of numeric weightings on a particular rule can have a non-local effect on the set of preferences as a whole. It is necessary to consider the preferences introduced by each rule in connection with all the other rules in the theory. This encoding process can be difficult, but once it is complete, it can be successfully applied computationally.

6 Conclusion

In this paper, we have suggested how one can recast an argumentation-style reasoning framework for plan ascription using a weighted abduction system. The question of which approach will turn out to be better suited to the problem remains open. In some respects, argumentation systems have more

expressive power than does weighted abduction. There are certain defeat rules that simply cannot be expressed within the framework of weighted abduction. For example, in principle the *Conflicting Action Defeat* rule could have been written "the other way around": one could have specified that an argument whose support set is a *sub*set of the support set of another argument defeats that latter argument. Nothing in the argumentation formalism would rule out such a defeat rule, although it would be highly counterintuitive. On the other hand, such a rule would not be expressible in the weighted abduction framework. However, if all such rules are counterintuitive, this may actually count as an advantage for weighted abduction: in general, it is computationally useful to restrict the generality of a system, so that just those facts that are reasonable to express can be expressed. Although no claims about relative computational costs can yet be made, because the implementation of ARGH is still preliminary, the general claim is consistent with experience.

A preliminary implementation of the weighted abduction formulation of plan recognition and evaluation described here has been completed using the Prolog Technology Theorem Prover (PTTP) [Sti88b]; it has been applied to all of the examples discussed by KP.

In general, weighted abduction carries certain implicit assumptions about how arguments are to be compared with respect to their global coherence. Argumentation systems require that these implicit assumptions be made explicit in many different ways. It remains an empirical question precisely how well suited the implicit global evaluation of weighted abduction is for various user-modeling tasks. The ongoing work, reported on in this paper, involves investigating further the claim that these assumptions are well suited to tasks like plan ascription that involve reasoning about mental state.

References

[All83] J. F. Allen (1983): Recognizing intentions from natural language utterances. In: M. Brady and R. C. Berwick, eds.: *Computa-*

tional Models of Discourse. Cambridge, MA: MIT Press.

[Cal91] R. J. Calistri-Yeh (1991): Utilizing user models to handle ambiguity in robust plan recognition. *User Modeling and User-Adaptive Interaction*, this issue.

[CG88] E. Charniak and R. Goldman (1988): A logic for semantic interpretation. In *26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY, 87-94.

[CMP90] P. R. Cohen, J. Morgan and M. E. Pollack (1990): *Intentions in Communication*. Cambridge, MA: MIT Press.

[Doy79] J. Doyle. (1979): A truth maintenance system. *Artificial Intelligence*, 12(3), 231-272.

[HSME88] J. R. Hobbs, M. Stickel, P. Martin, and D. Edwards (1988): Interpretation as abduction. In *26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY, 95-103.

[Kau90] H. A. Kautz (1990): A circumscriptive theory of plan recognition. In Cohen, Morgan, and Pollack, eds.: *Intentions in Communication*.

[Kon88] K. Konolige (1988): Defeasible argumentation in reasoning about events. In *Proceedings of the International Symposium on Machine Intelligence and Systems*, Torino, Italy.

[KP89] K. Konolige and M. E. Pollack (1989): Ascribing plans to agents: Preliminary report. In *Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI, 924-930.

[Lev89] H. J. Levesque (1989): A knowledge-level account of abduction. In *Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI, 1061-1067.

[NM89] H. T. Ng and R. J. Mooney (1989): Occam's razor isn't sharp enough: The importance of coherence in abductive explanation. In *Second AAAI Workshop on Plan Recognition*, Detroit, MI.

[Pie55] C. Pierce (1955): *Abduction and induction.* New York, NY: Dover.

[Pol86] M. E. Pollack (1986): Inferring domain plans in question-answering. TR 403, Artificial Intelligence Center, SRI International, Menlo Park, CA. (Also appears as a University of Pennsylvania PhD thesis.)

[Pol90] M. E. Pollack (1990): Plans as complex mental attitudes. In Cohen, Morgan, and Pollack (1990).

[Poo89a] D. Poole (1989): Explanation and prediction: an architecture for default and abductive reasoning. *Computational Intelligence*, 5(2), 97-110.

[Pop82] H. Pople (1982): Heuristic methods for imposing structure on ill-structured problems: the structuring of medical diagnosis. In P. Szolovits, ed.: *Artificial Intelligence in Medicine*. Boulder, CO: Westview Press.

[QDF88] A. Quilici, M. Dyer, and M. Flowers (1988): Recognizing and responding to plan-oriented misconceptions. *Computational Linguistics*, 14(3), 38-51.

[RZ91] B. Raskutti and I. Zukerman (1991): Generation and selection of likely interpretations during plan recognition. *User Modeling and User-Adaptive Interaction*, this issue.

[Reg83] J. Reggia (1983): Diagnostic expert systems based on a set-covering model. *International Journal of Man-Machine Studies*, 19(5), 437-460.

[Rei87] R. Reiter (1987): A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1), 57-96.

[Sel90] B. Selman and H. Levesque (1990) Abductive and Default Reasoning: A Computational Core. *Eighth National Conference on Artificial Intelligence (AAAI)*, 343-348.

[Sid85] C. L. Sidner (1985): Plan parsing for intended response recognition in discourse. *Computational Intelligence*, 1(1), 1-10.

[SK89] B. Selman and H. Kautz (1988). The complexity of model-preference default theories. In M. Reinfrank, J. deKleer, M. L. Ginsberg, and E. Sandewall eds.: *Second International Workshop on Non-Monotonic Reasoning*. Berlin: Springer Verlag.

[Sti88a] M. E. Stickel (1988). A Prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. In *International Computer Science Conference*, Hong Kong, 343-350.

[Sti88b] M. E. Stickel (1988). A Prolog technology theorem prover: implementation by an extended Prolog compiler. *Journal of Automated Reasoning*, 4, 353-380.