A Theorem Prover for Prioritized Circumscription

Andrew B. Baker and Matthew L. Ginsberg
Department of Computer Science
Stanford University
Stanford, California 94305

Abstract

In a recent paper, Ginsberg shows how a backward-chaining ATMS can be used to construct a theorem prover for circumscription. Here, this work is extended to handle *prioritized* circumscription. The ideas to be described have been implemented, and examples are given of the system in use.

1 Introduction

Of all the approaches to nonmonotonic reasoning, McCarthy's circumscription [McCarthy, 1980, McCarthy, 1986] seems to be the most popular. A great deal of work has gone into the development of new versions of circumscription designed to address various problems in commonsense reasoning. Much less work, however, has gone into the development of methods by which these formalisms might be implemented. In [Ginsberg, 1989], Ginsberg shows how a backward-chaining ATMS can be used to construct a theorem prover for circumscription. Here, this work is extended to handle *prioritized* circumscription.

In [Ginsberg, 1989], the following procedure is given to determine whether a particular sentence *q* follows from the circumscription axiom. First, you construct a default proof *d* for g, where *d* is a formula obtained by conjoining and disjoining instantiations of negations of the predicate being circumscribed. Then, you attempt to undermine this proof by constructing a default proof for ¬d. If you cannot undermine the original proof, then *q* is a consequence of the circumscription.

This procedure implements parallel circumscription since all the defaults are treated equally. For almost all applications of nonmonotonic reasoning, however, it is necessary to establish a prioritization among the defaults. In inheritance hierarchies, for example, the general view is that defaults about a subclass should override defaults about its superclasses. So if Tweety is both a penguin and a bird, the default that penguins generally do not fly should have a higher priority than the default that birds generally do.

In this paper, we show that the above method for computing parallel circumscription can be generalized to handle prioritized circumscription by extending the argument chain to consider arguments against the proposed proof of ¬d. One can view the procedure as a dispute between a believer *B* and an unbeliever *U*. *B* begins by presenting an argument for the desired conclusion. *V* then tries to *rebut* this argument by finding a counterargument at the same or higher priority. *B* now tries to *refute* the counterargument by finding a counter-counterargument at a strictly higher priority. This process of alternating rebuttals and refutations continues until one side cannot answer the other's last argument. If *B* gets the last word, the original sentence follows; if *U* gets the last word, the sentence does not follow.

In the next section, Section 2, we formalize these intuitions and show that this procedure does in fact correctly compute prioritized circumscription. In Section 3, we provide examples of the implementation at work. We finish in Section 4 with some concluding remarks.

2 Arguments, rebuttals, and refutations

Rather than working with the circumscription axiom, we will work directly with a set of sentences *T* and sets of sentences $D_1, \ldots, D_n$. *T* contains the certain facts, D_n contains the defaults with the highest priority, and so on down to D_1, which contains the defaults with the lowest priority. In the partial order corresponding to prioritized circumscription, a model M_1 is better than another model M_2 if for one of the default sets D_i the set of sentences in D_i that hold in M_1 is a proper superset of the set of sentences in D_i that hold in M_2, *and* if M_1 and M_2 agree on all the the default sets that have a higher priority than D_i:

Definition 1 *Let T be a consistent set of sentences, and let $D_1, D_2, \ldots, D_n$ be finite sets of sentences. Define a partial order on models of T, writing $M_1 > M_2$, if the following condition holds:*

For some i where $1 \le i \le n$, $M_1[D_i] \supset M_2[D_i]$; and for all j where $i < j \le n$, $M_1[D_j] = M_2[D_j]$.[1]

A model that is maximal in this partial order will be called D-maximal.

Proposition 1 *Let T be a set of sentences without function symbols, such that T includes domain closure and uniqueness of names assumptions. Let P be a collection of predicates that are prioritized into n levels such that*

[1] *M[D]* denotes the set of sentences in *D* that are satisfied by *M*. The symbol ⊃ indicates *strict* set inclusion.

the ones in level n will be circumscribed at the highest priority, and the ones at level 1 will be circumscribed at the lowest priority. Let D_i be the set of all propositions of the form $\neg p(x)$ where p is a predicate at level i and x is a ground instantiation of p's arguments. Now for any sentence q, q follows from the prioritized circumscription of all the predicates in P in T while allowing all other predicates to vary if and only if q is true in all D-maximal models of T.

Proof This is an easy consequence of Proposition 1' in [Lifschitz, 1985]. □

The assumptions made by Proposition 1 are rather strong. It may be possible to relax some of these assumptions, but this particular issue is not the concern of the current paper.

Now let us define what we mean by an argument:

Definition 2 Let p be a sentence. We will say that p is an argument if the following conditions hold:

1. p is of the form

$$\bigvee \bigwedge d_{ij}$$

for some collection of $d_{ij} \in D_1 \cup \ldots \cup D_n$, and

2. $T \cup \{p\}$ is satisfiable.

The priority of an argument is the least k such that the argument contains some $d_{ij} \in D_k$.[2]

Definition 3 Let p and q be sentences. We will say that p is an argument for q if p is an argument, and $T \cup \{p\} \models q$.

As discussed in the introduction, a prioritized circumscription proof will be viewed as a dispute between a believer B and an unbeliever U. It is important to understand that B and U are not playing by the same rules. B is trying to prove that some query q follows from the available assumptions, while U is trying merely to undermine B's efforts. Thus the unbeliever U does not need to actually refute B's arguments; it is sufficient for him to generate counterarguments that B cannot refute himself.[3] Therefore, U's arguments may be at the same priority level as the arguments they are rebutting; B's arguments must be at a higher level than the arguments they are refuting:

Definition 4 Let p and q be arguments. Then, p rebuts q if and only if p is an argument for $\neg q$, and the priority of p is greater than or equal to the priority of q; p refutes q if and only if p is an argument for $\neg q$, and the priority of p is strictly greater than the priority of q.

This is really a definition of attempted rebuttals and refutations. What we are most interested in is which arguments ultimately survive:

[2]Note that p may be the empty conjunction, i.e. true; in this case, we leave the priority undefined.

[3]In conventional nonmonotonic terms, B is trying to show that q holds in all relevant extensions of some default theory, while U is trying only to show that q does not hold in some extension. Were it our intention to accept a conclusion valid in any extension (as suggested by Reiter [1980]), the roles of B and U would be reversed. This remark is made formal in Proposition 3.

Definition 5 Let p and q be arguments. Then, p ultimately rebuts q if and only if p rebuts q, and p is not ultimately refuted; p ultimately refutes q if and only if p refutes q, and p is not ultimately rebutted.

At first glance, this definition may look circular since ultimate rebuttals and ultimate refutations are defined in terms of one another. But since a refuting argument must always have a higher priority than the argument that it is refuting, we have really defined an ultimate rebuttal at level k in terms of ultimate refutations at levels of at least k + 1. Since there are only a finite number of prioritization levels, the definition is well-founded.

We can now state our result:

Proposition 2 Let q be a sentence. Then q holds in all D-maximal models of T if and only if there is an argument for q that is not ultimately rebutted.

The proofs of Proposition 2 and of subsequent propositions are contained in Appendix A.

One might conjecture that there would be an analogous result stating that q holds in some D-maximal model if and only if there is an argument for q that is not ultimately refuted. The "only if" claim, however, is incorrect. If q were a new symbol, for instance, there would be no arguments for g, and yet q would still hold in some maximal model (since there also would be no arguments for $\neg q$.) In order to state the correct corollary, we define the notion of an extension:

Definition 6 Define the following equivalence relation on D-maximal models of T: $M_1 \sim M_2$ if for all i where $1 \le i \le n$, $M_1[D_i] = M_2[D_i]$; that is, if M_1 and M_2 agree on all the defaults. We will call these equivalence classes extensions. We will say that a sentence holds in an extension if it holds in every model in that extension.

This corresponds to the usual meaning of an extension.

Proposition 3 Let q be a sentence. Then q holds in some extension of the default theory if and only if there is an argument for q that is not ultimately refuted.

3 Implementation

To use Proposition 2 effectively, we need some way to determine the various arguments for a given sentence. Since there may be many such arguments, and since each of them may in turn have many counterarguments, it would be inefficient to consider each of them individually. The following proposition, however, lets us limit our attention to the single weakest argument. This is obtained by disjoining all the minimal conjunctive arguments that have appropriate priorities.

Proposition 4 Let q be a sentence, let i be a priority level, and assume that there is some argument for q with at least this priority. Let p be the disjunction of all the minimal conjunctive arguments for q that have priorities of at least i. Then, (1) p is an argument for q with priority of at least i, and (2) p is ultimately rebutted (refuted) if and only if every argument for q with priority of at least i is ultimately rebutted (refuted).

It turns out that these weakest disjunctive-normal-form arguments correspond exactly to the labels in an ATMS [de Kleer, 1986, Reiter and de Kleer, 1987]. Therefore, we can calculate these arguments using the backward-chaining ATMS provided with Ginsberg's multivalued logic system, MVL [Ginsberg, 1988]. Whenever the prioritized circumscriptive theorem prover invokes MVL, it supplies it with the relevant priority limit. This makes sure that default assumptions whose priorities are not high enough will not be considered.

We now present two examples of the system at work: the standard nonflying penguin example and Reiter's Nixon diamond [Reiter and Criscuolo, 198]]. The output is as produced by the program, except for minor textual modifications. (For example, the database is maintained in clausal form, but is displayed below using a PROLOG-like syntax.)

3.1 Tweety the penguin

This is the database for the penguin example:

```
Penguin(Tweety).
Bird(x) :- Penguin(x).
Flies(x) :- Bird(x).                P3 (priority 1)
Not(Flies(x)) :- Penguin(x).        P4 (priority 2)
```

Tweety is a penguin and therefore a bird. Birds normally fly; penguins normally do not fly. The P3 and P4 tags are used by the ATMS to keep track of the default assumptions. Note that we have assigned a higher priority to the penguin default. We ask the theorem prover to find something that does not fly:

```
Not(Flies(x))?
     Trying to prove Not(Flies(x)).
     Invoking MVL.
     Values returned are:
      bindings: -[x = Tweety},
      argument: P4 with x - Tweety.
          Trying to rebut (Not(Flies(Tweety))
            :- Penguin(Tweety)).
          Invoking MVL at priority >= 2.
          Values returned are: nil.
          Rebuttal fails.
     Proof succeeds!
x = Tweety.
```

The theorem prover begins by finding a default proof that Tweety does not fly. It then tries to rebut this argument. But since the default that penguins do not fly has a higher priority than the default that birds do, the prover is unable to rebut the argument. Thus, the proof ultimately succeeds with *x* bound to Tweety.

When we ask the prover to find something that flies, we get the following:

```
Flies(x)?
     Trying to prove Flies(x).
     Invoking MVL.
     Values returned are:
      bindings: {x = Tweety},
      argument: P3 with x = Tweety.
          Trying to rebut (Flies(Tweety) :-
           Bird(Tweety)).
          Invoking MVL at priority >= 1.
          Values returned are:
           argument: P4 with x = Tweety.
               Trying to refute
                (Not(Flies(Tweety)) :-
                 Penguin(Tweety)).
               Invoking MVL at priority > 2.
               Values returned are: nil.
               Refutation fails.
          Rebuttal succeeds.
     Proof fails!
```

At first, an argument is found that Tweety flies. But since the argument that birds fly has a priority of only 1, it is rebutted by the argument that penguins do not fly. Since this rebuttal cannot be refuted, the original query has no solution.

3.2 The Nixon diamond

In the penguin example, the competing defaults were at distinct priority levels. Here, we consider the classic example of competing defaults with the same priority:

```
Republican(Nixon).
Quaker(Nixon).
Hawk(x) :- Republican(x).          P7 (priority 1)
Dove(x) :- Quaker(x).              P8 (priority 1)
Not(Hawk(x)) :- Dove(x).
```

Nixon is both a Republican and a Quaker. Republicans are typically hawks, but Quakers are typically doves, and therefore not hawks. We ask the system to find a hawk:

```
Hawk(x)?
     Trying to prove Hawk(x).
     Invoking MVL.
     Values returned are:
      bindings: {x = Nixon},
      argument: P7 with x = Nixon.
          Trying to rebut (Hawk(Nixon) :-
           Republican(Nixon)).
          Invoking MVL at priority >= 1.
          Values returned are:
           argument: P8 with x = Nixon.
               Trying to refute (Dove(Nixon)
                 :- Quaker(Nixon)).
               Invoking MVL at priority > 1.
               Values returned are: nil.
               Refutation fails.
          Rebuttal succeeds.
     Proof fails!
```

Since P7 and P8 have the same priority, the argument that Nixon should be a hawk is rebutted by the argument that he should be a dove. Thus, there is no solution to the original query. (If the priorities had been different, this would not be the case. Thus, for example, if P7 had a higher priority than P8, we would be able to conclude that Nixon is a hawk.) It should be noted that since default P8 has the same priority as default P7, P8 can rebut P7, but it cannot refute it. That is, P8 has enough force to prevent P7 from being accepted as a conclusion, but it does not have enough force to stop P7 from interfering with other arguments. In particular, if we considered the argument that Nixon is a dove, then P7 would ultimately rebut this argument.

Things would not work as smoothly if we allowed arguments to refute other arguments with the same priority. *B* might start by asserting an argument p. U might rebut with some argument *q* at the same level. At this point, *B* could simply repeat his original argument! (If $T \cup \{q\} \models \neg p$, then $T \cup \{p\} \models \neg q$.) And so the dispute would continue interminably.

4 Conclusion

In this paper, we have presented a method of computing prioritized circumscription (for theories that contain uniqueness of names and domain closure axioms). Some comparisons to other work should be mentioned.

Using ideas from [Gelfond *et al.*, 1989], Przymusinski [1989] also presents an algorithm for prioritized circumscription, but it does not make full use of the priorities of the relevant arguments. Regardless of the query, Przymusinski's algorithm steps mechanically through each prioritization level, invoking his parallel circumscriptive theorem prover at each stage. This would be rather inefficient if the default theory had many prioritization levels, but only a few of these levels contained arguments that were relevant to the given query. Furthermore, Przymusinski's algorithm is described in terms of a specific first-order inference algorithm: MILO-resolution, a variant of ordered resolution. Ours, on the other hand, can use an arbitrary first-order theorem prover. This decomposition of the problem makes our method easier to understand, and it puts us in a position to benefit more easily from future advances in theorem-proving technology.

Much of the interesting research on defeasible reasoning has been described in procedural, rather than model-theoretical, terms. Using Proposition 2, we can now make some comparisons between this procedurally oriented work and prioritized circumscription.

Loui [1987], for example, discusses the general question: When is a defeater defeater a reinstater? In other words, if p is an argument against *q,* and *q* rebuts r, under what circumstances does p allow us to conclude r after all? For prioritized circumscription, we can give a sharp answer to this question: p reinstates r if and only if p ultimately refutes *q* as defined in Definition 5; it is not sufficient for p merely to ultimately rebut *q.*

Our proof procedure is also similar to some of the work by Horty and Thomason on inheritance hierarchies [Horty and Thomason, 1988]. They present a recursive definition of entailment that accepts any argument all of whose rebutters are themselves defeated by acceptable arguments. There are a number of differences, however, between the proposals; we will list only a few of these differences. First, since our system uses first-order logic instead of a graph-based formalism, it can handle disjunctive arguments. Consider, for example, Ginsberg's extension to the Nixon diamond where it is known that both hawks and doves are "politically motivated." Circumscription will conclude that Nixon is politically motivated since *Hawk(Nixon)* V *Dove(Nixon)* cannot be rebutted; Horty and Thomason will not reach this conclusion since there is no single path in the inheritance network that sanctions it.

Second, it appears that Horty and Thomason's graph-based formalism can support a richer structure of default orderings than can prioritized circumscription; for many of their hierarchies, there is no obvious way of assigning priorities to the default links in order to translate the hierarchy into prioritized circumscription. Third, and most importantly, they derive priorities based on specificity while we have to state the priorities explicitly. Therefore, the whole class of issues related to specificity is simply not addressed by our work. One intriguing possibility would be some kind of "dynamic circumscription" in which the theorem prover could be recursively invoked to determine whether one argument had a higher priority than another. This prioritization could be based on specificity or perhaps some more general scheme. We have implemented a primitive version of such a system although its formal properties are not entirely clear.

Finally, it is interesting that the proofs generated by our prioritized circumscriptive theorem prover resemble the way a person might reason: first tentatively drawing a conclusion, then thinking of possible objections to this conclusion, and finally trying to dispose of these objections. We find it encouraging that the proofs have this intuitive quality.

Acknowledgements

This work has been supported by the Rockwell Palo Alto Laboratory, by General Dynamics, and by NSF under grant number DCR-8620059. The first author is supported by an AFOSR graduate fellowship.

A Proofs

It will be convenient to describe the D-maximal models of *T* in terms of the defaults holding in them. Given a D-maximal model M, let p^i_M be the conjunction of all the defaults of priority at least *i* that hold in M.[4]

The following lemma will be useful in proving Propositions 2 and 3.

Lemma 1 *Let q be an argument. Then, (1) q holds in some D-maximal model of T if and only if it is not ultimately refuted; and (2) q holds in all D-maximal models of T if and only if it is not ultimately rebutted.*

Proof We will prove this by induction on the priority *i* of *q.* Specifically, when proving part (1) of the lemma for an argument of priority i, we will assume that both parts of the lemma hold for arguments with priorities greater than t; when proving part (2) for an argument of priority i, we will assume that both parts of the lemma hold for arguments with priorities greater than z, and that part (1) holds for arguments with priorities equal to i.

(1)(=>) If *q* is ultimately refuted by p, then *p* has a higher priority than q,, and *p* is not ultimately rebutted. By our inductive assumption, *p* holds in all D-maximal models, and hence *q* does not hold in any D-maximal models.

[4]This definition and the following proofs are based on less general versions in [Ginsberg, 1989].

(1)($\Leftarrow$) Assume that q is false in every D-maximal model. If q were to have priority n (the highest priority), then for any model M in which q held, there would be some D-maximal model M' such that $M[D_n] \subseteq M'[D_n]$. But then q would hold in M' as well; thus the priority of q must be less than n.

Let p be the disjunction of all the p_M^{i+1} for the various D-maximal models M. By our inductive assumption, p is not ultimately rebutted. Now, consider an arbitrary model M in which p holds. There will be some D-maximal model M' such that $M[D_j] = M'[D_j]$ for $i + 1 \leq j \leq n$ and $M[D_i] \subseteq M'[D_i]$. Since q is false in M', and since q consists solely of positive defaults of levels i and above, q must be false in M as well. Therefore, p ultimately refutes q.

(2)($\Rightarrow$) If q is ultimately rebutted by p, then p has a priority greater than or equal to that of q, and p is not ultimately refuted. By our inductive assumption, p holds in some D-maximal model, and hence q is false in this model.

(2)($\Leftarrow$) Assume that q is false in the D-maximal model M. Then, consider the argument p_M^i. By our inductive assumption, p_M^i is not ultimately refuted. Furthermore, p_M^i must rebut q, or else M would not be D-maximal. Therefore, p_M^i ultimately rebuts q. □

Proposition 2 *Let q be a sentence. Then q holds in all D-maximal models of T if and only if there is an argument for q that is not ultimately rebutted.*

Proof ($\Rightarrow$) Let p be the disjunction of all the p_M^1 for the various D-maximal models M. Since p holds in all of the D-maximal models, by Lemma 1 it is not ultimately rebutted. Furthermore, p holds only in the D-maximal models; therefore p is an argument for q.

($\Leftarrow$) If there is an argument for q that is not ultimately rebutted, then by Lemma 1, this argument (and thus q) must hold in all D-maximal models. □

Proposition 3 *Let q be a sentence. Then q holds in some extension of the default theory if and only if there is an argument for q that is not ultimately refuted.*

Proof ($\Rightarrow$) Suppose q holds in the extension defined by some conjunction of defaults p. By Lemma 1, p is not ultimately refuted. Since q holds in every model that p does, p must be an argument for q.

($\Leftarrow$) If there is an argument for q that is not ultimately refuted, then by Lemma 1, this argument must hold in some D-maximal model. Clearly, q holds in the extension defined by this argument. □

Proposition 4 *Let q be a sentence, let i be a priority level, and assume that there is some argument for q with at least this priority. Let p be the disjunction of all the minimal conjunctive arguments for q that have priorities of at least i. Then, (1) p is an argument for q with priority of at least i, and (2) p is ultimately rebutted (refuted) if and only if every argument for q with priority of at least i is ultimately rebutted (refuted).*

Proof The first claim is obvious, as is the ($\Leftarrow$) half of the second claim. For the ($\Rightarrow$) half of the second claim, assume that p is ultimately rebutted (refuted) by some argument r, but there is some argument p' for q with priority of at least i that is not ultimately rebutted (refuted). Each of the disjuncts in p' will be subsumed by a disjunct in p; hence, if $T \cup \{r\} \models \neg p$, then $T \cup \{r\} \models \neg p'$. Therefore, if r does not rebut (refute) p', this must mean that the priority of p' is greater than (greater than or equal to) the priority of r. But then since $T \cup \{p'\} \models \neg r$, and since p' is not ultimately rebutted (refuted), it must be the case p' ultimate refutes (rebuts) r, and hence r cannot ultimate rebut (refute) anything, which contradicts our assumption. □

References

[de Kleer, 1986] Johan de Kleer. An assumption-based TMS. Artificial Intelligence, 28:127-162, 1986.

[Gelfond et al, 1989] Michael Gelfond, Halina Przymusinska, and Teodor Przymusinski. On the relationship between circumscription and negation as failure. Artificial Intelligence, 38:75-94, 1989.

[Ginsberg, 1988] Matthew L. Ginsberg. Multivalued logics: A uniform approach to reasoning in artificial intelligence. Computational Intelligence, 4:265-316, 1988.

[Ginsberg, 1989] Matthew L. Ginsberg. A circumscriptive theorem prover. Artificial Intelligence, 1989. To appear.

[Horty and Thomason, 1988] John F. Horty and Richmond H. Thomason. Mixing strict and defeasible inheritance. In Proceedings of the Seventh National Conference on Artificial Intelligence, pages 427-432, 1988.

[Lifschitz, 1985] Vladimir Lifschitz. Computing circumscription. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence, pages 121-127, 1985.

[Loui, 1987] Ronald P. Loui. Theory and Computation of Uncertain Inference and Decision. PhD thesis, University of Rochester, Rochester, NY, 1987.

[McCarthy, 1980] John McCarthy. Circumscription — a form of non-monotonic reasoning. Artificial Intelligence, 13:27-39, 1980.

[McCarthy, 1986] John McCarthy. Applications of circumscription to formalizing common-sense knowledge. Artificial Intelligence, 28:89-116, 1986.

[Przymusinski, 1989] Teodor C. Przymusinski. An algorithm to compute circumscription. Artificial Intelligence, 38:49-73, 1989.

[Reiter, 1980] Raymond Reiter. A logic for default reasoning. Artificial Intelligence, 13:81-132, 1980.

[Reiter and Criscuolo, 1981] Raymond Reiter and Giovanni Criscuolo. On interacting defaults. In Proceedings of the Seventh International Joint Conference on Artificial Intelligence, pages 270-276, 1981.

[Reiter and de Kleer, 1987] Raymond Reiter and Johan de Kleer. Foundations of assumption-based truth maintenance systems: Preliminary report. In Proceedings of the Sixth National Conference on Artificial Intelligence, pages 183-188, 1987.