A Resolution-based Procedure for Default Theories with Extensions*

Monica D. Barback¹ and Jorge Lobo²

¹ Department of EECS
Northwestern University
Evanston, Illinois 60208-3118
barback@eecs.nwu.edu

² EECS Department M/C 154
University of Illinois at Chicago
851 S. Morgan Street #1120 SEO
Chicago, Illinois 60607-7053
jorge@eecs.uic.edu

Abstract. Default logic, introduced by Reiter, is an effective nonmonotonic reasoning system. In addition to being a stand alone paradigm for nonmonotonic reasoning, default logic has also been used to represent other nonmonotonic reasoning systems such as the stable model semantics and the well-founded semantics of logic programs. This paper complements these results by describing a sound and complete resolution based proof procedure for a large class of default theories. The procedure can be used to determine if an arbitrary formula is in an extension. It also has the advantage of generating a partial extension, which is contained in the final extension, thus helping to describe the meaning of a given default theory. Further, the procedure does not necessarily have to compute every extension in order for a single query to be answered. The procedure has been implemented in C-Prolog.

1 Introduction

Default logic, introduced by Reiter [Rei80], is an effective nonmonotonic reasoning system (cf.[Eth87, Eth88]). In addition to being a stand alone paradigm for nonmonotonic reasoning, default logic has also been used to represent other nonmonotonic reasoning systems such as the stable model semantics ([MT89], [LS92], [BF91]) and the well-founded semantics [Bar94] of logic programs. We complement this integral relationship between default logic and logic programming by developing a sound and complete proof procedure for certain default theories. Readers familiar with proof procedures for logic programs will notice the similarities with our procedure.

A given default theory may have zero, one, or many extensions. The purpose of a proof procedure is to be able to query the default theory with a formula

* Partial support was provided by the National Science Foundation, under grant Nr. IRI-92-10220.
and determine if it is in one extension (the extension membership problem) or
determine if it is in every extension (the entailment problem). Without a way to
answer these questions, a default theory becomes meaningless.

In Reiter's original work, a sound and complete resolution based proof pro-
cEDURE was introduced for the class of normal default theories, which answered
the extension membership problem. The implementation involved successive ap-
plications of resolution to the consequents of the defaults in the default theory.
At the conclusion of the proof all defaults involved in the proof are subject to a
consistency check. However, little work has been done to expand this refutation
procedure for general default theories.

There have been several bottom-up approaches to generate extensions. Ether-
ington [Eth87], developed a bottom up algorithm to generate all of the exten-
sions of the class of default theories called finite ordered default theories. Finite
ordered default theories are more general than normal default theories, thus ex-
tending the work of Reiter. Although this algorithm is guaranteed to generate all
of the extensions of a finite ordered default theory, the procedure is not guaran-
teed to halt, even if an extension exists. More general bottom-up procedures are
described by Marek and Truszczynski in [MT93]. There have also been procedures
for computing answers to modified versions of default logic. An example of such
procedures can be found in [Bes89].

Our goal when we started this investigation was to find a top-down proof
procedure based on resolution for the standard definition of default logic.

The proof procedure that we developed is sound and complete for finite semi-
normal default theories with the property that each subset of the default theory
has an extension. Normal default theories have this property. The set of even
default theories also has this property. Even default theories are a larger class of
default theories than the finite ordered default theories of Etherington. We also
show how we could relax the assumption that each subset of the default theory
has an extension if we work with alternative definitions of default logic.

The advantages of our procedure can be stated as follows:

- The algorithm can be used to determine if a formula is in any extension of
  a default theory $\Pi$.
- We can determine if a formula is in every extension from a query that must
  fail.
- The algorithm generates a partial extension that is contained in some exten-
sion of the default theory, thus giving us some more information about the
structure of the default theory.
- Neither complete extensions nor every extension needs to be generated in
order for a query to be answered. Thus, it is possible that generating a proof
could be much less expensive than deriving all extensions as in the bottom
up approaches.
- The algorithm is implemented in C-Prolog, so the actual cost of the algorithm
  can be tested with real-world examples.

The organization of the paper is as follows. Section 2 gives some preliminaries,
Sections 3 and 4, detail the development of the proof procedure. Many of the