Resolution for Skeptical Stable Semantics

P. A. Bonatti

Dip. di Informatica - Università di Torino
Corso Svizzera 185, I-10149 Torino, Italy
E-mail: bonatti@di.unito.it

Abstract. An extension of resolution for skeptical stable model semantics is introduced. Unlike previous approaches, our calculus often needs to consider only a strict subset of the program rules. Moreover, we characterize a large class of programs whose derivations may proceed in a thoroughly goal-directed way. Some inferences, which depend on non-ground negative goals, can be drawn without resorting to negation-as-failure; as a consequence, many goals which flounder in the standard setting, have a successful skeptical derivation. The paper contains a preliminary study of some interesting derivation strategies.

KEYWORDS: Stable semantics, Skeptical derivations, Resolution, Floundering, Strategies.

1 Introduction

The stable model semantics [6] is one of the most important semantics for normal logic programs. It constitutes a neat, general formalization of the idea of negation as failure; moreover, it enjoys numerous interesting relations with other formalisms, deduction procedures and problem domains of great interest (e.g. default and autoepistemic logics, truth maintenance systems, abduction and diagnosis).

Procedural and proof-theoretic accounts of the stable model semantics still constitute an open area of research. The stable model semantics associates each logic program $P$ to a set of minimal models called stable models. $P$ credulously entails a goal $G$ if $G$ holds in some stable model of $P$; $P$ skeptically entails $G$ if $G$ holds in all the stable models of $P$. Previous research on implementations and proof techniques has mainly focussed on credulous entailment, which can be easily adapted to implement skeptical entailment ([2, 5, 11] are just a few of the recent approaches based on the construction of whole stable models). Each procedural approach must tackle the following technical problems:

1. Both credulous and skeptical entailment are intrinsically difficult (for ground programs, they are NP-complete and coNP-complete, respectively). There are techniques for dividing programs into smaller modules [8], some of which can be interpreted through specialized procedures (e.g. for stratified programs), thereby restricting the demanding general algorithms to (hopefully small) subsets of the given program.
2. Credulous and skeptical procedures must detect infinite derivations. For example \( P = \{ p \leftarrow p \} \) entails not \( p \) because the goal \( \leftarrow p \) originates only an infinite derivation \( \leftarrow p, \leftarrow p, \ldots \). Apt et al. [1] proved some difficulties related to loop-checking in the presence of variables (no program-independent loop-checking exists, even for function-free programs). Recently, Chen and Warren [4] introduced a tabulation mechanism that solves this problem for function-free programs; this mechanism is also a convenient way of factorizing common sub-derivations.

3. In general, a deduction procedure which constructs one or more stable models, must take into account the whole program (as opposed to the subset relevant to the given query); this is an obstacle to the development of realistic query processing mechanisms. More details will be given in the following.

4. A partially related problem is that no general goal-directed procedure can be found. The answer to a query may depend on parts of the program which are syntactically unrelated to the query itself.

5. There are well-known difficulties in dealing with non-ground negative goals. Some of the existing approaches apply to restricted (e.g. range restricted) programs [4]; some others [2] compute the ground instantiation of the program (which is too expensive for many real world applications); other approaches make use of disunifiers or related notions [7].

In this paper, points 3, 4 and (partially) 5 are tackled by introducing a calculus for skeptical entailment, that may exploit the tabulation mechanism by Chen and Warren, as well as other forms of tabulation and delay, or loop-checking. Skeptical entailment has always been based on credulous reasoning in the past, in the sense that at least one entire stable model needs to be constructed (with the exception of [10], where logic programs and goals are translated into a classical propositional theory, and nonmonotonic deduction is reduced to classical propositional entailment). However, point 3 cannot be tackled through credulous reasoning. Consider the following result.

**Proposition 1.** For all programs \( P \) and all goals \( G \) such that \( P \) credulously entails \( G \), there exists a program \( P' \supseteq P \) such that \( G \) is not credulously entailed by \( P' \).

Essentially, this proposition states that no correct credulous procedure can be a function of a strict subset of the given program \( P \). On the contrary, some skeptical conclusions depend only on a small part of the program. For example,

---

1. This observation does not hold for semi-monotonic logics. Indeed [12] presents a skeptical deduction system based on credulous reasoning for constrained default logic (a semi-monotonic variant of Reiter’s default logic) which in general does not need to consider all the rules of the theory.

2. Sketch of the proof. Let \( r \) be a ground atom which cannot be unified with the head of any rule of \( P \) (if necessary, let \( r \) be a new propositional symbol). Let \( P' = P \cup \{ r \leftarrow \text{not } r \} \). It is easy to see that \( P' \) has no stable models; it follows immediately that \( G \) is not credulously entailed by \( P' \).