Adding Negation-as-Failure to Intuitionistic Logic Programming

Anthony J. Bonner
L. Thorne McCarty

Rutgers University
Department of Computer Science
New Brunswick, NJ 08903
bonner@paul.rutgers.edu
mccarty@cs.rutgers.edu

Abstract

Intuitionistic logic programming is an extension of Horn-clause logic programming in which implications may appear "embedded" on the right-hand side of a rule. Thus, rules of the form $A(x) \leftarrow [B(x) \leftarrow C(x)]$ are allowed. These rules are called *embedded implications*. In this paper, we develop a language in which negation-as-failure is combined with embedded implications in a principled way. Although this combination has been studied by other researchers, Gabbay has argued in [10] that the entire idea is logically incoherent since modus ponens would not be valid in such a system. We show how to solve this problem by drawing a distinction between *rules* and *goals*. To specify the semantics of rules and goals, we then develop an analogue of Przymusinski's perfect model semantics for stratified Horn-clause logic [20]. Several modifications are necessary to adapt this idea from classical logic to intuitionistic logic, but we eventually show how to define a *preferred model* of a *stratified intuitionistic rulebase*, and this enables us to specify the semantics of such a rulebase by reference to its preferred models. Finally, we prove a soundness and completeness theorem. Throughout the paper, we discuss various examples of the use of intuitionistic embedded implications plus negation-as-failure, to demonstrate the utility of the language.

This paper appears in *Proceedings of the North American Conference on Logic Programming (NACLP)*, pages 681-703. MIT press. Conference held in Austin, TX, Oct 29–Nov 1 1990.