Semantics of Non-monotonic Reasoning based on Perfect Model

fangqing DONG         hirosi NAKAGAWA

Dept. of Electrical and Computer Eng.
Yokohama National Univ.

Abstract

Because of the lack of general semantics for non-monotonic formalisms, the relationship between the major forms of non-monotonic formalisms has not been understood very well. Recently, it has been proposed by C. Przymusinski that perfect model semantics for logic programming is also suitable for some special cases of non-monotonic formalisms. The importance of this result is not only for shedding new light on these two fields, but also for establishing a close relationship between main forms of non-monotonic formalisms. However, these understandings are achieved only in stratified programs. In this paper, we introduce a new class of programs called conditional programs, define its declarative semantics by least exceptional models, show its relation with perfect model semantics and non-monotonic logics, and give a procedure about how to transform a conditional program into a general program. This new semantics of conditional programs shares a lot of common characteristics with logic programming and non-monotonic reasoning, such as priorities between predicates, minimizing extensions of predicates, and minimizing abnormality. A better understanding about the relationship between forms of non-monotonic formalisms can be achieved by least exceptional model semantics of conditional programs.

1 Introduction

The declarative semantics for logic programming has been represented by the minimal model. However, it has been pointed out that this semantics is not suitable for general programs. Recently, perfect model semantics, a new approach to the declarative semantics of logic programming, has been proposed. In this semantics, priorities are given to all predicates occurring in a general program. The perfect model for a general program are defined by minimizing the extensions of predicates according to their priorities. The lower its priority is, the smaller its extension should be. Moreover, it has been shown that the perfect model semantics is equivalent to suitable main non-monotonic formalisms.

However, this new semantics is only adequate for the model-based semantics of some special cases of non-monotonic formalisms. Furthermore, we argue that the epistemological difficulty of how to reason about the defaults of real world should be considered.
In view of these problems, we introduce a new class of programs, called conditional programs, with negative predicate in head and define its declarative semantics which shares some common characteristics with perfect model semantics and non-monotonic reasoning, such as showing priorities between predicates, minimizing extensions of predicates according to their priorities, and minimizing abnormality etc. In section 2, we illustrate the perfect model for a general program and its relation with non-monotonic logics. In section 3, we define conditional programs and its declarative semantics by least exceptional models. In section 4, we show how to transform a conditional program into a general program on the equality of the least exceptional model of a conditional program with the perfect model of the corresponding general program. At last, we discuss the relationship between least exceptional model semantics and non-monotonic logics.

2 Perfect Model

Perfect model semantics[Przy88a] [Przy88b] is a new declarative semantics for logic programming and adequate for the model-based semantics of non-monotonic formalisms. In this section, we only consider general programs without functions and introduce their perfect models. Firstly, we give several concepts here.

The following clause is called general program clause, where A's, B's and C are atoms.

\[ A_1 \land \ldots \land A_n \land \neg B_1 \land \ldots \land \neg B_m \supset C \]

A set of finite general program clauses P is called general program.

A general program P can be represented by a dependency graph G[Przy88b] [Geff88], whose vertices are predicate symbols occurring in P, and there is a directed edge from A to B iff there is a clause in P such that A occurs in its head and B in its body. If B is a negative premise of A, then the edge is called negative. If there is a directed edge from A to B, and the edge is negative, then it is said that B has lower priority than A. More generally, for any ground general program clause of P

\[ A_1 \land \ldots \land A_n \land \neg B_1 \land \ldots \land \neg B_m \supset C, \]

we have the following priority relations.

\[ A_i \leq C \quad 1 \leq i \leq n \]
\[ B_j < C \quad 1 \leq j \leq m \]

By priority relation < between ground atoms, Herbrand base \( B_L \) could be decomposed into several strata according to the priorties of ground atoms. \( \neg A \) should be true if A could not be derived from low stratum to its stratum. More precisely, a model M should be preferable to another N, if there is a predicate A whose extension in M is larger than the one in N, then there must exist a predicate B whose priority is lower than A and extension in M is smaller than the one in N. These observations are formalized by the notation of a perfect model.

Definition 2.1 \( M \prec N \) and perfect model