Representing Exceptions in Rule-Based Systems

P. L. Tan  T. S. Dillon  J. Zeleznikow

Department of Computer Science
La Trobe University
Bundoora, Vic, 3083
Australia

Abstract

Making inferences and decisions in an imprecise and changing world limits the potential leverage of a system, particularly, in occurrence of exceptions. Many rule-based systems which make inferences using rules attempt to deal with expected exceptions by augmenting the rules in the knowledge base. The result is a whole tree of rules which widens the search space of a goal. This paper introduces several kinds of exceptions and presents a methodology for representing exceptions at the object level and meta level of rule-based systems. The proposed approach accommodates instance exceptions as well as assertion exceptions, hence allowing for the recognition of exception rules as distinct from the normal rules in rule-based systems.

Keywords and phrases: object-level rules, meta-level rules, rule model, exceptions

1. Introduction

Rule based systems have been well known for problem solving and inferencing using heuristic rules embedded in the knowledge base. In the real world with its irregular and changing environment, rules of an application may evolve, and often there may be events that do not fit or match the knowledge in the knowledge base. When this happens, the system which does not accommodate the unexpected events will fail to carry out correct inferencing. As a consequence, most systems require the knowledge base to be updated or augmented with additional rules to take account for the changes in the environment. However, this process is normally performed by the knowledge engineer at the time of creation of the knowledge base without provision for dynamic accommodation for
exceptions at runtime. Without an adequate mechanism to represent exceptions in the knowledge base, the system is denied any ability to learn in the presence of an unexpected event. Not being able to cope with this situation is a serious handicap to a system that claims anything other than trivial intelligence.

So far, some progress has been made towards exceptions in inheritance in knowledge based systems [Touretsky 86]. However, to the best of the authors' knowledge, no work has been carried out in representing exceptions in rule-based systems where rules are arranged in a non-hierarchical framework. This paper presents an approach to accommodate the unexpected events at runtime without any additional rules. As well as presenting a method of representing exceptions, the proposed approach stores the atypical instances, hence allowing for the relationship between the atypical instances and the atypical assertions.

The next section examines the problems of exceptional knowledge at both object level and meta-level of rule-based systems. It further considers ways in which the knowledge is refined to accommodate exceptions for each knowledge level. An approach to representing such exceptions is introduced and an example is shown to illustrate this approach.

2. Representation

2.1 Knowledge Rules

Data organisation and knowledge representation are key points in the design of most rule-based systems. A system which contains incorrect knowledge or information gives wrong conclusions or wrong inferencing. Similarly, a relevant rule that was not executed reflects the low efficiency level of inferencing power and a false sense of reliability of the system. Despite much effort having been invested in designing rule-based systems, there is always a possibility of less productivity due to unexpected events. In such situation, a system that is not designed to deal with these events may fail to give the correct inferencing.

A rule is invoked when the antecedents are successfully matched with assertive facts in the working memory. This will result in activation of the consequent of the rule. A rule typically has the following representation:

Rule-id: Antecedent: IF <condition>
Consequent: THEN <action>

In a rule-based system, rules may stored at 2 levels of knowledge: