Theory and Algorithm for Rule Base Refinement

Hai Zhuge\(^1\), Yunchuan Sun\(^2\), and Weiyu Guo\(^1\)

\(^1\)Knowledge Grid Research Group, Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Science, 100080, Beijing, China
zhuge@ict.ac.cn
Http://kg.ict.ac.cn

\(^2\)Department of Computer Science, College of Information Science, Beijing Normal University, 100875, Beijing, China
Sun88@mail.bnu.edu.cn

Abstract. Rule base refinement plays an important role in enhancing the efficacy and efficiency of utilizing a rule base. A rule base concerns three types of redundancies: implication-rule redundancy, abstraction-rule redundancy and dead-end-condition redundancy. This paper proposes two approaches: one is to remove implication redundant rules by using the closure of literal set and the other is to remove abstraction redundant rules by using rule-abstraction. We have developed a software tool to support the first approach. Experiments show that the tool can work correctly and efficiently. The proposed approach can be applied to more application fields.

1 Introduction

Rule base refinement plays an important role in enhancing the efficacy and efficiency of utilizing a rule base. Many approaches and tools have been proposed for detecting and eliminating redundant and inconsistent rules. The static rule checking utility was proposed to check inconsistencies and incompleteness [16]. The system CHECK was developed to verify the consistency and completeness of knowledge base. Its operations are based on the construction of a dependency chart, which shows the dependencies among rules and between rules and classes [12, 13]. The EVA project aimed to build an integrated set of tools to check the redundancy, consistency, completeness, and correctness of any KBS written by any KBS language, but it only achieved its goal in a limited manner. Redundancy was checked in EVA by theorem proving [3, 4]. The Expert System Checker checks the completeness and consistency of a knowledge base through decision-tables [5]. The system KB-Reducer was developed for knowledge base reduction. It can also check inconsistency and redundancy in rule base [6]. The system COVER can carry out seven checking: redundancy, conflict, subsumption, unsatisfiable conditions, dead-end rules, circularity and missing rules [8, 14, 15]. Two general-purpose verification systems VSE and VSE II were developed based on the core of an interactive inductive theorem prover [9]. Applying the classical methods to detect some anomalies in non-monotonic knowledge base has been discussed in [7]. The issue of analysis and

\(^1\)This Work was supported by National Science Foundation of China.

© Springer-Verlag Berlin Heidelberg 2003
verification of selected properties of rule-based systems and a tabular form of single-level rule-based systems was studied [11]. A methodology for the validation of rule-based expert systems proposed in [10]. A partial instantiation schema that exports local search to first-order knowledge bases is proposed in [2], and it can handle the forms of depth-limited consistency and inconsistency. Zhuge presented a theory about object and object-abstraction and theories about rule-order and rule-mapping formalism, and also established an analogical reasoning model OAM [18]. Based on our previous work, this paper proposes the notion of abstraction rule redundancy and then develops a method for rule base refinement that uses object abstraction. We suppose that each rule in the rule base takes the form of Horn-clause for the convenience. Actually, any rule can be translated into one or several rule(s) with the form of Horn-clause, which is (are) equivalent to the original rule.

2 Implication Redundancy

2.1 Basics

If a rule \( r \) can be deduced through logical reasoning by some rules in a rule base \( R \), we say \( r \) is implied by \( R \), denoted as \( R \models r \). If each rule of a rule set \( S \) is implied by \( R \), we say \( S \) is implied by \( R \), denoted as \( R \models S \). A rule is called trivial, if it takes the form of \( p_1 \land p_2 \land \ldots \land p_m \Rightarrow p_i \), \( 1 \leq i \leq m \), i.e., the result of the rule appears in the preconditions. It is obviously that a trivial rule always holds.

Let \( R \) be a rule base, and \( r \) is a rule in \( R \). If \( R-\{r\} \models r \), we call \( r \) an implication redundancy rule for \( R \), i.e., \( R \)'s function is equivalent to that of \( R-\{r\} \), for \( r \) can be replaced by some other rule(s) in \( R-\{r\} \). Hence, \( r \) can be removed from \( R \).

Implication redundancy can be caused by many factors such as equivalency, transitivity and subsumption, which have been studied by many researchers [1, 3, 4, 6, 8, 12-16]. Complicated implication redundancy also exists, for instance, if \( p \rightarrow q_1, p \rightarrow q_2, q_1 \land q_2 \rightarrow w, p \rightarrow w \) are four rules in a rule base, then \( p \rightarrow w \) is a redundant rule.

All implication redundant rules we discuss above are due to the logical implication relationship among rules. In order to decide whether a rule \( r \) is redundant or not for a rule base \( R \), we should determine whether \( r \) could be derived from \( R-\{r\} \) or not.

2.2 Equivalence among Rule Bases and Minimal Cover of a Rule Base

The set of all rules implied by a rule base \( R \) is called the closure of \( R \), denoted \( R' = \{ r \mid R \models r \} \). Obviously, \( R' \) is unique for a given rule base \( R \).

We say a rule base \( R \) is equivalent to another one \( S \) if \( R = S' \). In this case, \( R \) can be replaced with \( S \). It is obvious that the equivalence among rule bases is reflexive, symmetric, and transitive.

Lemma 1. A rule base \( R \) is equivalent to \( S \) if and only if \( R \subseteq S' \) and \( S \subseteq R' \).