Chapter 40
Rule-Based Systems for Medical Diagnosis

V. S. Giridhar Akula
Methodist College of Engineering and Technology, India

ABSTRACT

A rule-based system is a set of "if-then" statements that uses a set of assertions, to which rules on how to act upon those assertions are created. Rule-based expert systems have played an important role in modern intelligent systems and their applications in strategic goal setting, planning, design, scheduling, fault monitoring, diagnosis, and so on. The theory of decision support system is explained in detail. This chapter explains how the concepts of fuzzy logic are used for forward and backward chaining. Patient data is analyzed with the help of inference rules.

INTRODUCTION

A rule-based system is a set of "if-then" like statements that uses a set of assertions. Rule based systems are used to preserve and utilize the knowledge. Rule based systems are also known as Knowledge-based expert systems. Knowledge is encoded in the form of rules and utilizes knowledge available from experts in a specified domain. Rule based expert systems (Achour, 2001) or in general expert systems were introduced in mid 1960s.

Rule-based systems differ from standard procedural or object-oriented programs in that there is no clear order in which code executes. Here, the knowledge of the expert is integrated in a set of rules, each of which encodes a small piece of the expert's knowledge. Each rule has a left hand side and a right hand side. The left hand side contains information about certain facts and objects. One of the rules on the agenda is picked, and its right hand side is executed, and then it is removed from the agenda. Through rule based systems, intelligent decisions can be made and can trace the decision making process.

Rule-based expert systems (Aikins, 1980), use human expert knowledge to solve real world problems that normally would require human intelligence. Expert knowledge is often represented in the form of rules as well as data within the computer.

DOI: 10.4018/978-1-5225-1908-9.ch040

Depending upon the problem requirement, these rules and data can be recalled to solve problems. Rule-based expert systems (Aczel & Sounderpandian, 2005; Aikins, 1983) have played an important role in modern intelligent systems and their applications in strategic goal setting, planning, design, scheduling, fault monitoring, diagnosis and so on.

As already discussed, a rule-based system consists of if-then rules, a bunch of facts, and an interpreter controlling the application of the rules, given the facts. These if-then rule statements are used to formulate the conditional statements that comprise the complete knowledge base. A single if-then rule assumes the form 'if x is A then y is B' and the if-part of the rule 'x is A' is called the antecedent or premise, while the then-part of the rule 'y is B' is called the consequent or conclusion. There are two broad kinds of inference engines used in rule-based systems: forward chaining and backward chaining systems. In a forward chaining system, the initial facts are processed first, and keep using the rules to draw new conclusions given those facts.

In a backward chaining system, the hypothesis (or solution/goal) which is tried to reach is processed first, and keep looking for rules that would allow concluding that hypothesis. As the processing progresses, new sub goals are also set for validation. Forward chaining systems are primarily data-driven, while backward chaining systems are goal-driven.

Consider an example with the following set of if-then rules:

Rule 1: If *A* and *C* then *Y*.
Rule 2: If *A* and *X* then *Z*.
Rule 3: If *B* then *X*.
Rule 4: If *Z* then *D*.

If the task is to prove that *D* is true, given *A* and *B* are true. According to forward chaining, start with Rule 1 and go on down ward till a rule that fires is found. Rule 3 is the only one that fires in the first iteration. After the first iteration, it can be concluded that *A, B*, and *X* are true. The second iteration uses this valuable information. After the second iteration, Rule 2 fires adding *Z* is true, which in turn helps Rule 4 to fire, proving that *D* is true.

Forward chaining strategy is especially appropriate in situations where data are expensive to collect, but few in quantity. However, special care is to be taken when these rules are constructed, with the preconditions specifying as precisely as possible when different rules should fire. In the backward chaining method, processing starts with the desired goal, and then attempts to find evidence for proving the goal. Having knowledge of the same example, the task to prove that *D* is true would be initiated by first finding a rule that proves *D*. Rule 4 does so, which also provides a sub goal to prove that *Z* is true. Now Rule 2 comes into play, and as it is already known that *A* is true, the new subgoal is to show that *X* is true. Rule 3 provides the next sub goal of proving that *B* is true. But that *B* is true is one of the given assertions. Therefore, it could be concluded that *X* is true, which implies that *Z* is true, which in turn also implies that *D* is true. Backward chaining is useful in situations where the quantity of data is potentially very large and where some specific characteristic of the system under consideration is of interest. If there is not much knowledge what the conclusion might be, or there is some specific hypothesis to test, forward chaining systems may be inefficient. In principle, the same set of rules are used for both forward and backward chaining. In the case of backward chaining, since the main concern is with matching the conclusion of a rule against some goal that is to be proved, the 'then' part of the rule is usually not expressed as an action to take but merely as a state, which will be true if the antecedent part(s) are true (Donald, 1986).

Related Content