CPS331 Lecture: Rule-Based Systems last revised September 4, 2018

Objectives:

1. To introduce condition-action rules
2. To introduce forward and backward chaining

Materials:

1. Projectable of Cawsey figure 2.6
2. Projectable of roman numeral rules
3. Projectable of roman numeral rules extended to handle numbers up to 99
4. Projectable of Bagger rules (from Winston)
5. Projectable of animals rules (example from Winston)

I. Introduction

A. The knowledge-representation schemes we considered earlier were aimed at representing factual knowledge (knowledge that). Another kind of knowledge we need to be able to represent is procedural knowledge (knowledge of how-to).

B. One approach that is typical in symbolic systems is the use of "if-then" or condition-action rules.

C. A rule-based system has the following general structure

PROJECT: Cawsey figure 2.6

1. The rules have the general form

if *some condition(s)*
then *some action*(s)

a) The "if" part of the rule specifies the conditions under which the rule is applicable.

(1)This part of the rule is sometimes called the antecedent.

(2)If the antecedent a rule matches the current state of the problem, the rule is said to be triggered.

(3)When a triggered rule is applied - i.e. when it's action is carried out - it is said to fire.

(4)Typically, more than one rule is triggered. In this case, some strategy is used to select a specific rule to fire. This is something we will discuss more fully later in the lecture, but for now we will just assume that the rules are considered in the order written, and the first triggered rule will fire. (The method used to decide which rule to fire if multiple rules are triggered is called the conflict-resolution policy of the rule interpreter)

(5)In any case, when a rule fires the current state of the problem is updated and then the search for an applicable rule begins all over from the start.

(6)When no rule is triggered, then the computation is complete.

b) The "then " part of the rule specifies the actions to be performed when the rule fires. It is sometimes called the consequent.

c) In general, both the antecedent and the consequent can be composite. If the antecedent is composite, all parts of it must be true; if the consequent is composite, all the things it specifies are done.

2. The database of facts can be represented using any of the knowledge representation schemes used to represent facts (e.g. predicate calculus)

a) Initially, it represents the starting point of the problem being solved

b) As the various rules fire, it is updated.

D. A simple example is a set of rules for producing the roman numeral representation for a number less than 40.

PROJECT. Note that, in this particular case, the antecedents all involve a single test, but most of the consequents are composite

1. Let's work an example. Convert 24 to Roman numerals.

a) Initially, N (the current state of the problem) is 24 - i.e. the database of facts is something like

N = 24

b) Right away, all but the first are triggered. Following the conflict resolution strategy we just discussed, R2 fires.

This causes 'X' to be printed, and N is reduced to 14.

c) Now, the same rule fires again.

This causes another 'X' to be printed, and N is reduced to 4.

d) At this point the R5 fires.

This causes ''IV' to be printed, and N is reduced to 0.

e) Now, no more rules are triggered, so we're done, having printed "XXIV".

2. Let's work another example (class exercise): 18

3. One of the advantages of a rule based system is that they are easy to modify. Let's extend the system to handle numbers up to 99.

DEVELOP WITH CLASS. (Note importance of ordering. Work through examples as appropriate)

Change the first rule to "if N is greater than 99 ..."

Add:
```
if      N is greater than 89
then    print 'XC'
        subtract 90 from N

if      N is greater than 49
then    print 'L'
        subtract 50 from N

if      N is greater than 39
then    print "XL"
        subtract 40 from N
```

PROJECT modified rules.

Class Exercise: 79

II. The Control-Scheme (Interpreter or Engine)

A. The control-scheme, or engine, is the part of the system that applies the rules to the current problem.

B. There are two general types of engine:

1. The example we have just looked at uses a forward-chaining interpreter.

2. We will look at backward-chaining engines shortly.

C. For either type of engine, one key issue is conflict resolution in the case where multiple rules are triggered.

1. The example we just did simply chose the first triggered rules. Although this is the simplest approach, it does mean rule order is very important.

Example: What would our original Roman numeral system do if R6 were first?

ASK

(It would translate any number into something like IIIIII ....)

2. The book discussed some alternatives. What are they?

ASK

a) Prefer rules that involve facts most recently added to the knowledge base

b) Prefer rules that involve more specific conditions

c) Include a priority as part of each rule. Fire the highest priority triggered rule. (The simple scheme we just used can actually be thought of as a case of this approach, in which the priorities of the rules are their order of appearance. If explicit priorities are used, ties are broken by order of appearance).

This is, in a way, the strategy used by Weizenbaum's ELIZA system, where each pattern could be thought of as a rule that is triggered if the input sentence matches it.

d) Fire all applicable rules at once.

D. Another example of a forward-chaining system: a rule-based grocery bagger (from Winston)

PROJECT example rules - note use of notion of a "step" to segregate rules.

E. The alternative to a forward-chaining system is a backward chaining or goal-directed system.

1. In a goal-directed system, one always has a goal (which may be simple or composite).

2. Rules are triggered if their consequent matches something in the goal that is not currently satisfied.

   a) When a rule fires, any part of its antecedent that is not satisfied is added to the goal, and becomes part of what we need to prove.

   b) If anything in a rule's antecedent cannot be proved after attempting to do so, but another rule was triggered at the same time, the other rule fires instead. This is known as backtracking.

   c) While rules for backward-chaining systems can have composite antecedents, they need to have a single-item consequent in order to support rule selection based on the goal

3. Example: an animal identifier system (adapted from Patrick Henry Winston *Artificial Intelligence* Addison-Wesley 1984)

   a) Background: Winston introduces this system this way:

   "Suppose that Robbie, our robot, wants to spend a day at the zoo. Plainly Robbie will enjoy that visit more if he can recognize various animals. Assume Robbie can see basic features like color and size, but he does not know how to combine such facts into conclusions like, say, this is a zebra, or that is a tiger". (Winston *Artificial Intelligence* 2e p. 177)

   b) PROJECT rules.

   c) Initial goal is always identify animal as _____ given various facts.

d) Antecedents marked * can only be satisfied by the initial facts, since there is no rule with them a consequent

e) Conflict resolution strategy will be first rule that is triggered for most recently added antecedent, with backtracking.

f) Work through example given tawny color, dark spots, has hair, chews cut, long legs, long neck

(1) I9 fires - add to "carnivore" to goal

(2) I5 fires (first rule for most recently-added-antecedent) - fails

(3) I6 fires (alternative for carnivore) - fails

(4) I10 fires (alternative for original goal) - fails on black stripes

(5) I11 fires (alternative for original goal) - add "ungulate" to goal

(6) I7 (first rule for most recently-added-antecedent) fails on hoofs

(7) I8 (alterative for ungulate) - adds "mammal" to goal

(8) I1 (first rule for most-recently-added antecedent) succeeds;
therefore I8 succeeds
therefore I11 succeeds

(9) Animal is identified as a giraffe

F. The decision whether to use a forward-chaining or backward-chaining approach depends in large measure on the nature of the problem.

1. The Roman-numeral problem was a natural for forward chaining; it would be hard to formulate a backward-chaining version

2. The book gave an example of a set of rules used both ways, but the backward-chaining approach was a bit forced. (Once again, the problem was more appropriate for forward chaining)

3. In general

   a) Forward chaining is used for systems that want to discover the consequences of a given set of facts (e.g. the building is on fire or the Roman representation of a number)

   b) Backward chaining is used for systems where there is a single initial goal that may give rise to others (e.g. identifying an animal)

III. What Can Rule-Following Accomplish?

A. What Can a Rule-Based System Accomplish?

Think back to Newell and Simon's claim. In essence, a Physical Symbol System is a rule-based system. Recall the following from their article:

"A physical symbol system consists of a set of entities, called symbols, ... that can occur as components of another type of entity called an expression (or symbol structure). ... Besides these structures, the system also contains a collection of processes that operate on expressions to produce other expressions: processes of creation, modification, reproduction and destruction."

The "processes" they refer to are, in essence, rules.

B. If the PSSH is true, then he infrastructure of intelligence is, in fact, a sort of rule-based system, and achieving general intelligence would entail formulating an appropriate - presumably very complex - set of rules.

C. However, whether or not general intelligence is achievable in this way (which I doubt), it remains the case that rule-based systems can be developed to solve specific problems - though even here they have limitations.

D. We next consider expert systems, which are a form of rule-based system addressing specific problems - a form of weak AI.