Production Systems

Suppose, for example, that a mathematician says:

> If x and y are both greater than zero,
> so is the product of x and y.

In first-order predicate calculus:

$$\forall x \forall s\ [\ G(x, 0)\ \wedge G(x, 0)\ ]\ \Rightarrow G(time(x,y), 0)$$

However, the following equivalent formulation could be obtained:

$$\forall x \forall s\ [\ G(x, 0)\ \wedge \neg G(time(x,y), 0)\ ]\ \Rightarrow \neg G(y, 0)$$

In general, the sentence (A V B V C) is logically equivalent to any of the following implications:

$\neg A \wedge \neg B \Rightarrow C$
$\neg A \wedge \neg C \Rightarrow B$
$\neg B \wedge \neg C \Rightarrow A$
$\neg A \Rightarrow B \vee C$
$\neg B \Rightarrow A \vee C$
$\neg C \Rightarrow A \vee B$

Production Systems

Productions (rules) consist of situation-action pairs.

> The problem state is what is currently known about the problem.
>
> The situation is a description of problem states in which the rule is applicable.
>
> The action is a modification of the problem state.

Example:

IF	smoke is coming from the television
THEN	the television is broken.

Using intermidiate inferences:

IF	smoke is coming from the television
THEN	there is an electrical short.
IF	there is an electrical short
THEN	the television is broken.

Production Systems

A basic productions system contains three distinct components:

Context:

Data base of currently relevant facts.
Starts out with initial data.
Modifiable by production rules.

Rule base:

Conditions -

Predicates which must be satisfied by rule to apply.

Predicates defined with respect to the context.

Actions -

Modifications to the context.
Additions/deletions of relevant facts.

Interpreter:

Chooses specific rule(s) to apply.
Implements actions.

Production Systems

Simple productions systems operate in a sequence of three-phase cycles:

Matching - Determine which rules might apply.
Conflict resolution - Choose a single rule when more than one rule applies.
Action - Implement the chosen rule.

Example:

R1: IF On-CL(green) **THEN** Put-On-CL(produce)
R2: IF On-CL(packed in small container)
THEN Put-On-CL(delicacy)
R3: IF On-CL(refrigerated) **OR** ON-CL(produce)
THEN Put-On-CL(perishable)
R4: IF On-CL(weighs 15 lbs) **AND** ON-CL(inexpensive)
AND ON-CL(perishable) **THEN** Put-On-CL(staple)
R5: IF ON-CL(perishable) **AND** On-CL(weighs 15 lbs)
THEN Put-On-CL(turkey)
R6: IF ON-CL(weighs 15 lbs) **AND** On-CL(produce)
THEN Put-On-CL(watermelon)

Interpreter:

1. Find all productions whose condition parts are TRUE and make them applicable.
2. If more than one production is applicable, then deactivate any production whose action adds a duplicate symbol to the CL.
3. Execute the action of the lowest numbered applicable production. If no productions are applicable, then quit.
4. Reset the applicability of all productions and return to 1.

Production Systems

Initial conditions:

CL = (Green, weighs 15 lbs.)

Cycle 1:

R1 applies.
CL = (Produce, Green, weighs 15 lbs.)

Cycle 2:

R1, R3 and R6 apply.
R1 eliminated because of duplicate symbol.
R3 chosen over R6 because of lower number.
CL = (Perishable, Produce, Green, weighs 15 lbs.)

Cycle 3:

R1, R3, R5 and R6 apply.
R1 and R3 are redundant.
R5 chosen over R6 because of lower number.
CL = (Turkey, Perishable, Produce, Green, weighs 15 lbs.)

Cycle 4:

R6 applied.
CL = (Watermelon, Turkey, Perishable, Produce, Green, weighs 15 lbs.)

Cycle 5:

No rule applicable.

Conflict-resolution strategy

- **Specificity ordering** - Suppose the conditions of one triggering rule are a superset of the conditions of another triggering rule. Use the rule with the superset on the ground that it is more specialized to the current situation.

- **Rule ordering** - Arrange all rules in one long priority list. The triggering rule appearing earliest in the list has the highest priority. The others are ignored.

- **Data ordering** - Arrange all possible aspects of the situation in one long priority list. The triggering rule having the highest priority condition has the highest priority.

- **Size ordering** - Assign the highest priority to the triggering rule with the toughest requirements, where toughest means the longest list of constraining conditions.

- **Recency ordering** - Consider the most recently used rule to have the highest priority.

- **Context limiting** - Reduce the likelihood of conflict by separating the rules into groups, only some of which are active at any time. Have a procedure that activates and deactivates groups.

Forward and Backward Chaining

- Forward chaining:
  - — It is a data-driven procedure.
  - — It starts with the sentences in the KB and generates new conclusions that in turn can allow more inferences to be made.
  - — The inference processes are not directed towards solving any particular problem.
  - — It is usually used when a new fact is added to the KB and we want to generate its consequences.
- Backward chaining:
  - — It is a goal-driven procedure.
  - — It starts with something we want to prove, finds implication sentences that would allow us to conclude it, and then attempts to establish their premises in turn.
  - — It is usually used when there is a goal to be proved.
  - — It finds all answers to a question posed to the KB.

Backward Chaining Systems

Example of backward chaining:

R1: IF (hoofs, t) **THEN** (subclass, mammal)
R2: IF (subclass, mammal) **AND** (eats, cub)
AND (antlers, none) **THEN** (order, ungulate)
R3: IF (order, ungulate) **AND** (neck, long) **AND** (legs, long)
AND (spots, dark) **THEN** (type, giraffe)

Hypothesis:

(type, giraffe)

Known information:

(eats, cub) (antlers, none) (neck, long) (legs, long) (spots, dark)
(hoofs, t)

Work backwards from hypothesis:

R3 might apply.

Three of the conditions (neck, long) (legs, long) (spots, dark) are satisfied.

Need to show (order, ungulate)

R2 might apply.

Two of the conditions (eats, cub) and (antlers, none) are satisfied.

Need to show (subclass, mammal)

R1 might apply.

Advantages and disadvanges of Production System

Advantages:

Modularity

Changes to one rule do not directly affect other rules.
Rules communicate only through the context.

Uniformity

Highly constrained representational structure.
Easier to understand individual rules.
Possible to build general purpose rule interpreters.

Naturalness

Statements about what to do in predetermined situations is easily encoded into production rules.
Experts often explain their reasoning strategies using rule-like descriptions.

Disadvantages:

Inefficiency

Difficult to focus effort in productive directions.
Difficult to organize rules hierarchically.
Difficult to take large step in direction of solution.

Opacity

Hard to follow control flow.
No easy way to represent sequence of problem solving steps.

Production systems

Appropriate problem domains for production systems:

- Domains in which knowledge is diffuse, consisting of many facts (e.g. medicine), as opposed to domains in which there is a concise, unified theory (e.g. physics).
- Domain in which process can be represented as a set of approximately independent actions (e.g. a patient monitoring system), as opposed to domains with dependent subprocesses (e.g. payroll processing).
- Domain in which knowledge can be easily separated from the manner in which it is used (e.g. a taxonomic system), as opposed to cases in which representation and control are merged (e.g. a recipe).