Production System is a system based on IF…THEN…rules and consisting of three parts:

(Set of production rules, working memory (KB), control strategy)

Outline of this lecture

- Components of Production System.
- Recognize-Act Cycle.
- Control of search in production systems
- Forward and Backward Chinning.

Components of Production System

1. A ***set of rules*** of the form $C_i \rightarrow A_i$ where C_i is the condition part and A_i is the action part. The condition determines when a given rule is applied, and the action determines what happens when it is applied.
2. One or more ***knowledge databases*** that contain whatever information is relevant for the given problem. Some parts of the database may be permanent, while others may temporary and only exist during the solution of the current problem. The information in the databases may be structured in any appropriate manner.
3. A ***control strategy*** that determines the order in which the rules are applied to the database, and provides a way of resolving any conflicts that can arise when several rules match at once.

The kind of rules my use:

Deductive Inference Rule

Given "A" and "A implies B", we can conclude "B":

$$\frac{\begin{array}{l} A \\ A \Rightarrow B \end{array}}{B}$$

Example:

$$\frac{\begin{array}{l} \text{It is raining} \\ \text{If it is raining, the street is wet} \end{array}}{\text{The street is wet}}$$

Abductive Inference Rule

Given "B" and "A implies B", it might be reasonable to expect "A":

$$\frac{\begin{array}{l} B \\ A \Rightarrow B \end{array}}{A}$$

Example:

$$\frac{\begin{array}{l} \text{The street is wet} \\ \text{If it is raining, the street is wet} \end{array}}{\text{It is raining}}$$

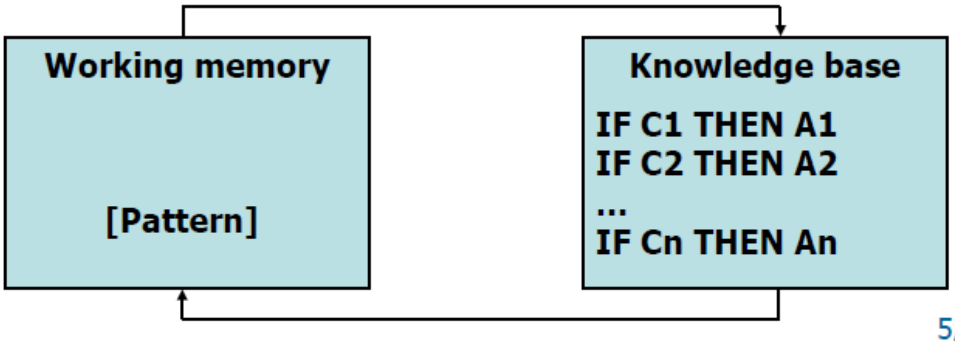
Recognize-Act Cycle.

Operation of the recognize-act cycle are based on ***the pattern-directed search***:

1. The working memory contains a pattern corresponding to the current state in the problem-solving process
2. The pattern is matched against IF or THEN part of rules (it depends on the direction of search)
3. If several rules match the pattern, ***the conflict set*** appears. The production rules in the conflict set are said to be ***enabled***. Selection of one rule from the conflict set is called ***conflict resolution***.
4. One of the enabled rules is fired and it changes the content of the working memory
5. The control cycle repeats with the modified working memory
6. The process terminates when no rules are matched by the content of the working memory

C-condition

A-action



5/16

Matching

The condition/premise patterns in the rules need to be ***matched*** with the known facts.

Consider, a typical rule (about the value of horses) that matches a set of facts:

Rule

IF:	x is a horse
	x is the parent of y
	y is fast
THEN:	x is valuable

Facts

Comet	is-a	horse
Prancer	is-a	horse
Comet	is-parent-of	Dasher
Comet	is-parent-of	Prancer
Prancer	is	fast
Dasher	is-parent-of	Thunder
Thunder	is	fast
Thunder	is-a	horse
Dasher	is-a	horse

We simply need to see which values can be assigned to the variables in the rule. For example, "Comet is-a horse" matches "x is-a horse", but "Comet is-a lion" would not.

Bindings for "x is-a horse"	Bindings for "y is fast"	Bindings for "x is a parent of y"
x = Comet	y = Prancer	x = Comet , y = Dasher
x = Prancer	y = Thunder	x = Comet , y = Prancer
x = Thunder		x = Dasher , y = Thunder
x = Dasher		

From these we can deduce that there are two possible bindings applicable to the rule:

"x = Comet and y = Prancer" , "x = Dasher and y = Thunder"

The rule then tells us "x is valuable", i.e. "Comet is valuable" and "Dasher is valuable".

Control of search in production systems

For real world problems a knowledge base typically includes a huge number of IF...THEN... rules. In order to make search more effective, developers of intelligent systems must use heuristic control of search.

Control through choice of search strategy

It is possible to choose a direction of search in production systems:

1. Data-driven search (forward chaining)
2. Goal-driven search (backward chaining)

Data-driven search:

1. Begins with a problem description (a pattern) added to the working memory
2. The recognize-act cycle compares matching of the pattern with IF part of rules in the knowledge base
3. Firing a rule, its THEN part is added to the working memory and the process continues
4. Search stops when the goal is found

7/16

Example: Data-driven search

Knowledge base:

1. **IF** John is a student (A) **THEN** John enjoys student's life (B)
2. **IF** John enjoys student's life (B) **THEN** John meets friends (C) **AND** John participates in university's events (D)
3. **IF** John meets friends (C) **THEN** John needs money (E)
4. **IF** John needs money (E) **THEN** John has a job (F)
5. **IF** John meets friends (C) **AND** John participates in university's events (D) **THEN** John has little free time (G)
6. **IF** John has little free time (G) **AND** John has a job (F) **THEN** John is not successful in studies (H) **AND** John does not receive scholarship (I)

Example: Data-driven search

Knowledge base:

1. **IF** A **THEN** B
2. **IF** B **THEN** C **AND** D
3. **IF** C **THEN** E
4. **IF** E **THEN** F
5. **IF** C **AND** D **THEN** G
6. **IF** G **AND** F **THEN** H **AND** I

Start - A
Goal - I

Iteration	Working memory	Conflict set	Fired rule
0	A	1	1
1	A, B	1, 2	2
2	A, B, C, D	1, 2, 3, 5	5
3	A, B, C, D, G	1, 2, 3, 5	3
4	A, B, C, D, G, E	1, 2, 3, 5, 4	4
5	A, B, C, D, G, E, F	1, 2, 3, 5, 4, 6	6
6	A, B, C, D, G, E, F, H, I	1, 2, 3, 5, 4, 6	goal

Goal-driven search:

1. A goal (a pattern) is added to the working memory
2. The recognize-act cycle compares matching of the pattern with THEN part of rules in the knowledge base
3. Firing a rule, its IF part is added to the working memory and the process continues
4. Search stops when facts on problem are found

Example:

Goal-driven search

Knowledge base:

1. **IF** p **AND** q **THEN** goal
2. **IF** r **AND** s **THEN** p
3. **IF** w **AND** r **THEN** q
4. **IF** t **AND** u **THEN** q
5. **IF** v **THEN** s
6. **IF** start **THEN** v **AND** r **AND** q

Iteration	Working memory	Conflict set	Fired rule
0	goal	1	1
1	goal, p, q	1, 2, 3, 4	2
2	goal, p, q, r, s	1, 2, 3, 4, 5	3
3	goal, p, q, r, s, w	1, 2, 3, 4, 5	4
4	goal, p, q, r, s, w, t, u	1, 2, 3, 4, 5	5
5	goal, p, q, r, s, w, t, u, v	1, 2, 3, 4, 5, 6	6
6	goal, p, q, r, s, w, t, u, v, start	1, 2, 3, 4, 5, 6	stop