Al-Mustaqbal University

College of Science

Production Systems in Artificial Intelligence

Lecture 6

Assist Lecturer . Hadi salah

Lecture Outline

1. What is a Production System?
2. Main Components
3. How it Works
4. Conflict Set and Rule Firing
5. Example 1: String Reordering
6. Example 2: Number Transformation
7. Advantages and Limitations
8. Summary

What is a Production System?

A **production system** is a rule-based problem-solving method used in Artificial Intelligence.

It consists of:

- a set of **rules**,
- a **working memory** that stores the current state,
- and a **control strategy** that decides which rule to apply.

Each rule is usually written as:

IF condition THEN action

The system keeps applying rules until the goal is reached or no rule can be applied.

Main Components of a Production System

1. Rule Base

A collection of production rules.

2. Working Memory

Contains facts or the current problem state.

3. Inference Engine

Matches rules with the current state and selects one rule to fire.

4. Control Strategy

Resolves conflicts when more than one rule can be applied.

General Operation Cycle

1. Store the current state in working memory.
2. Compare the current state with all rules.
3. Find all applicable rules (**conflict set**).
4. Select one rule based on the control strategy.
5. Fire the selected rule.
6. Update the working memory.
7. Repeat until goal or halt.

Important Terms

Conflict Set: all rules that can be applied now.
Rule Fired: the rule chosen and executed.

Conflict Resolution Strategies

When several rules are applicable at the same time, the system must choose one.

Common strategies:

- **Rule ordering:** choose the first rule in the list.
- **Specificity:** choose the most specific rule.
- **Recency:** prefer the rule using the newest data.
- **Priority:** assign priority values to rules.

In many educational examples, we simply choose the **first matching rule**.

Example 1: String Reordering Problem

Goal: Rearrange the symbols until the string becomes ordered.

Production Rules:

$$\begin{aligned} R_1 &: cb \rightarrow bc \\ R_2 &: ba \rightarrow ab \\ R_3 &: ca \rightarrow ac \end{aligned}$$

Initial Working Memory:

$$aacbaab$$

Goal State:

$$aaaabbc$$

These rules move the symbols toward alphabetical order.

Example 1: Step-by-Step Table

Iteration	Working Memory	Conflict Set	Rule Fired
1	aacbaab	1, 2	1
2	aabc aab	3	3
3	aabacab	2, 3	2
4	aaabcab	3	3
5	aaabacb	2, 1	2
6	aaaabcb	1	1
7	aaaabbc	∅	halt

Note: Spaces are added only to make the changing part easier to read.

Example 1: Explanation

At each iteration, the system searches for patterns inside the current string:

- If it finds `cb`, it replaces it with `bc`.
- If it finds `ba`, it replaces it with `ab`.
- If it finds `ca`, it replaces it with `ac`.

So the characters gradually move into the correct order:

$$aacbaab \to aabcaab \to aabacab \to aaabcab \to aaabacb \to aaaabcb \to aaaabbc$$

This is a simple example of how a production system solves a problem by repeated rule application.

Example 1: Important Observation

Why does this work?

The rules are designed to reduce disorder in the string.

- ba → ab moves a left.
- ca → ac also moves a left.
- cb → bc moves b left relative to c.

Finally, all symbols become sorted:

$$aaaabbc$$

So the production system behaves like a rule-based sorting process.

Example 2: Number Transformation Problem

Goal: Transform a number until it reaches 1.

Production Rules:

$$R_1 : \text{If } n \text{ is even, then } n \to \frac{n}{2}$$
$$R_2 : \text{If } n \text{ is odd and } n > 1, \text{ then } n \to n - 1$$

Initial Working Memory:

$$n = 10$$

Goal State:

$$n = 1$$

This example shows production systems with numeric states instead of strings.

Example 2: Step-by-Step Table

Iteration	Working Memory	Conflict Set	Rule Fired
1	10	R_1	R_1
2	5	R_2	R_2
3	4	R_1	R_1
4	2	R_1	R_1
5	1	$\varnothing$	halt

Sequence:

$$10 \to 5 \to 4 \to 2 \to 1$$

Advantages of Production Systems

- Easy to understand and design.
- Rules are modular and can be added independently.
- Suitable for expert systems and decision systems.
- Good for representing human reasoning in IF–THEN form.
- Flexible for many AI problems.

Limitations of Production Systems

- Large rule sets may become hard to manage.
- Conflict resolution may be complex.
- Some problems need many iterations.
- Rules may interact in unexpected ways.
- Efficiency decreases when the number of rules becomes very large.

Applications of Production Systems

Production systems are used in:

- Expert systems
- Automated reasoning
- Diagnosis systems
- Game playing
- Planning and decision making
- Rule-based intelligent agents

Summary

- A production system is a rule-based AI model.
- It contains rules, working memory, and a control strategy.
- The system repeatedly matches, selects, and fires rules.
- The conflict set contains all applicable rules.
- We studied:
  - a string reordering example,
  - and a number transformation example.

Production systems solve problems step by step using rules.

Thank You

Questions?