

# Reasoning

---

$$P \rightarrow Q$$

- *data-directed reasoning*. reasoning from P to Q. If we assert P, then we assert Q.
- *goal-directed reasoning*. reasoning from Q to P. Given the goal Q, we establish the goal P.

# Production Systems

---

A *production system* is a forward-chaining reasoning system that uses rules of a certain form called *production rules*.

A production system keeps an ongoing memory of assertions called *working memory* – WM. The WM is constantly changing.

A *Production Rule* has two parts an *antecedent* made up of a set of *conditions* and a *consequent* made up of a set of *actions* to be taken if the antecedent conditions are true.

IF *conditions* THEN *actions*]

# Cycle of Operations

---

1. **recognize** Which rules are applicable?
2. **resolve conflict** Which rules execute?
3. **act** Alter the working memory.

# Working Memory

---

## Working Memory Elements (WMEs)

*(typeattribute<sub>1</sub> : value<sub>1</sub> ... attribute<sub>n</sub> : value<sub>n</sub>)*

- (person age:30 home:RedBank)
- (goal task:putDown importance:5  
urgency:1)

$$\begin{aligned} \exists x [\text{TYPE}(x) \wedge \text{ATTRIBUTE}_1(x) = \text{value}_1 \\ \dots \wedge \text{ATTRIBUTE}_n(x) = \text{value}_n \end{aligned}$$

# Production Rules

---

Conditions

*(type attribute<sub>1</sub> : specification<sub>1</sub> ... attribute<sub>n</sub> : specification<sub>n</sub>)*

Each specification is one of the following:

- an atom
- a variable
- an evaluable expression within [ ]
- a test, within
- , the conjunction, disjunction, or negation of a specification

(person age:[n + 4] occupation: x)

# Production Rules (cont)

---

## Actions

- **ADD** *pattern* : Add the new WME specified by *pattern*.
- **REMOVE** *i* Remove the WME that matched the *i*th condition of the antecedent of the rule.
- **MODIFY** *i attribute specification* Modify the WME that matched the *i*th condition in the antecedent by replacing its value for *attribute* by *specification*

# Example

---

IF (student name:x) THEN ADD (person name: x)

IF (person age:x name:n) (birthday who:n)  
THEN MODIFY 1 (age[x + 1])  
REMOVE 2

# Another Example

---

(counter value:1)

(brick name:A size:10 position:heap)

(brick name:B size:30 position:heap)

(brick name:C size:20 position:heap)

1) IF (brick position:heap name:n size:s)

    -(brick position:heap size:{>s})

    -(brick position:hand)

    THEN MODIFY 1 (position hand)

2) IF (brick position:hand)

    (counter value:i)

    THEN MODIFY 1(position i)

        MODIFY 2(value[i+1])

# Conflict Resolution

---

In doing goal directed reasoning, if more than one rule can fire, how do we pick the rule to execute.

- *random*
- *order*
- *specificity*
- *recency*
- *refractoriness*

# Efficiency

---

- Matching is costly
- RETE algorithm

# Summary

---

- Expert Systems
- Psychological Modeling
- Modularity, Simple Constrol Structure, Transparency.
- Applications