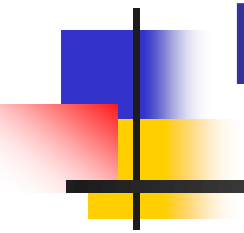
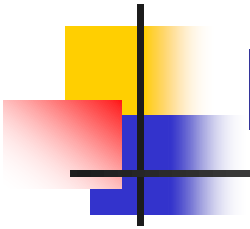


Knowledge-Based Systems

Russell Chap 9-3, 18-8

Luger Chap. 6, 8



Introduction

- Problem solving
  - Knowledge + Control
- Knowledge
  - <If…then…> rules → *Rule-based reasoning*
    - Prolog : Implication sentence
    - Jess : condition => action
  - Past experience → *Case-based reasoning*(CBR)
- Control (search)
  - Rule-based reasoning → Backward chaining, Production systems
  - Case-based reasoning → Find similar cases

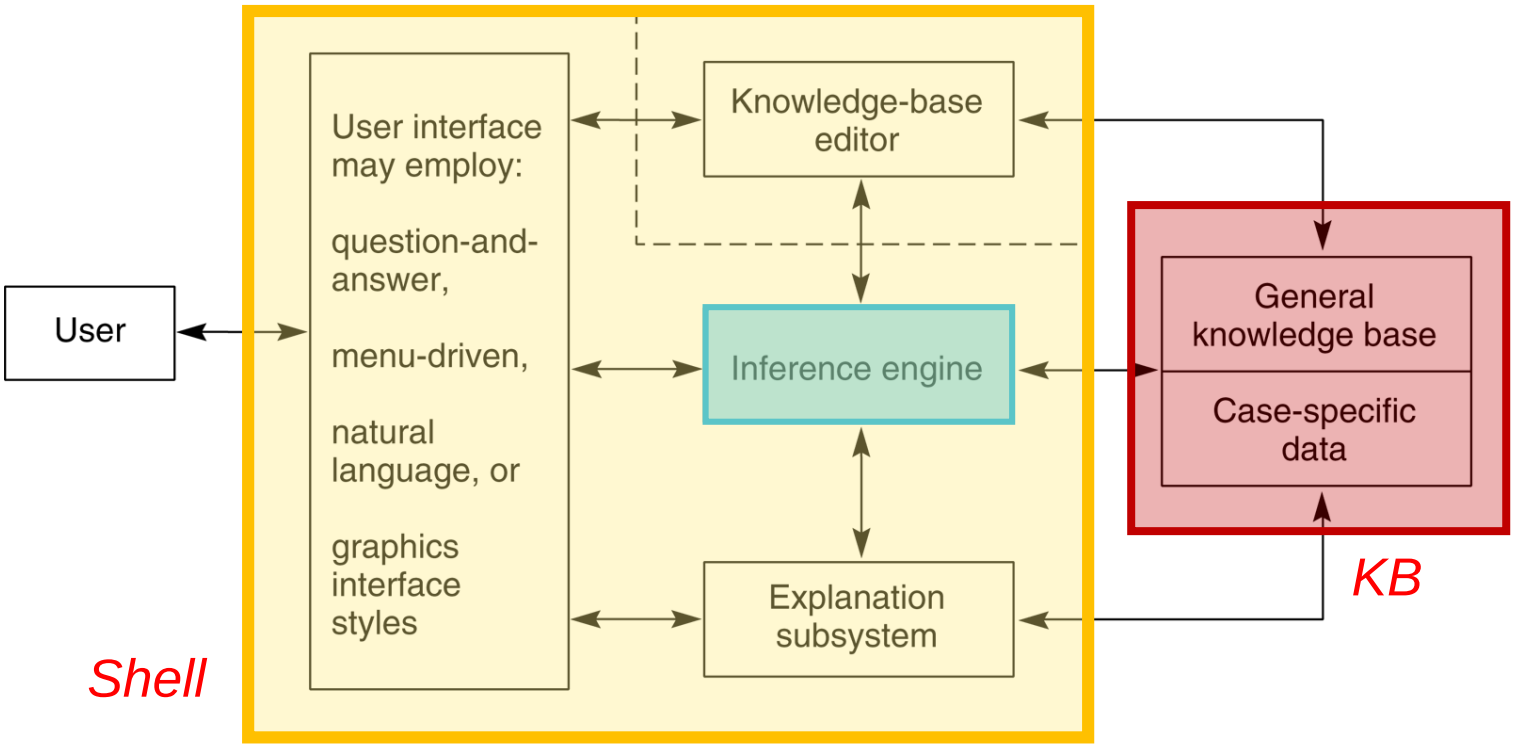
Expert Systems

- Expert system
  - Knowledge-based program
  - Provides answers to problems in specific domain
  - Problem Categories
    - Interpretation, prediction, diagnosis, design, …
- Separation of knowledge and control
  - Easy to represent knowledge
  - Easy to build and change KB
  - Same control and interface can be used for various systems

    ⇨ *Expert System Shell*

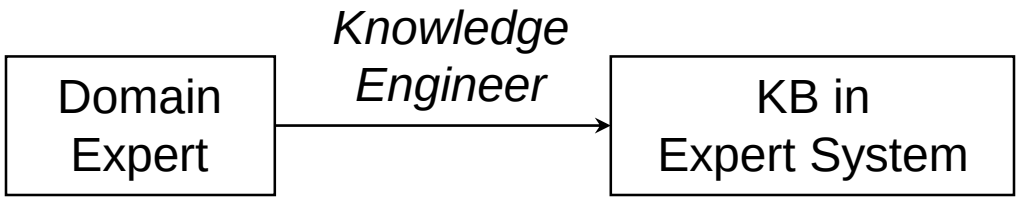
Expert Systems

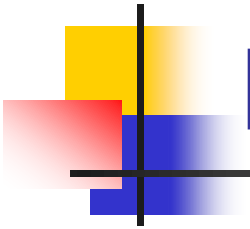
- Expert system architecture



Knowledge Engineering

- Knowledge acquisition
  - Building KB by acquiring knowledge from experts
    → Major bottleneck in building expert system
- Knowledge engineer
  - Select tools
  - Extract knowledge (interview)
  - Build efficient KB



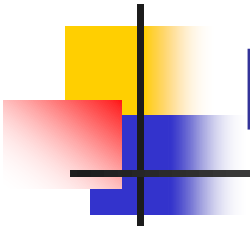


Rule-Based Reasoning

- Represent knowledge as explicit rules
  - Example rule (MYCIN)

```
If   ((infection is primary-bacteremia) AND
      (portal-of-entry is gastrointestinal-tract))
Then (conclude bacteroid 0.7)
```

  - Expert system as a prolog program
    - Knowledge base - Set of prolog sentences
    - Inference engine – Prolog interpreter (backward chaining)
  - Expert system as a production system
    - Knowledge base - Set of <if – then> rules
    - Inference engine - Production system (forward chaining)

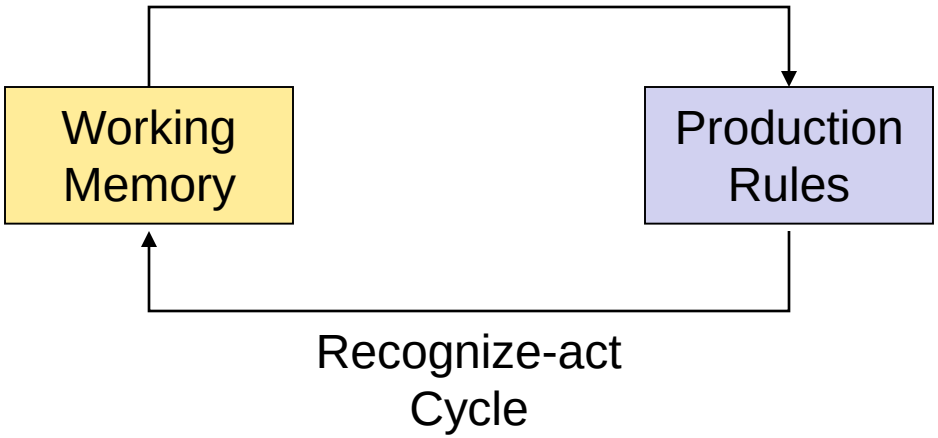


Rule-Based Reasoning

- Advantages
  - Use of expert knowledge
    → Good performance in limited domains
  - Separation of knowledge and control
  - Explanation is possible
- Disadvantages
  - Need knowledge acquisition
  - Rules are vulnerable to noisy data
    - Ex> If (temp > 39.0) Then …

Production Systems

- Production System
  - A computational model of problem solving process
  - Used for implementation of expert systems



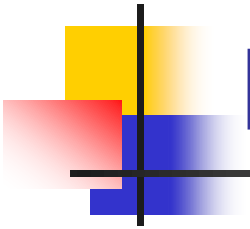
Production Systems

- Production rules
  - Problem solving knowledge
  - <condition> → <action>

```
Ex>     if     earning(X, unsteady)
       then   income(inadequate)
       if     clear(X)
       then   pickup(X)
```

- Working memory
  - Current states of the world

```
Ex>     earning(1500, unsteady)
       clear(a)
```

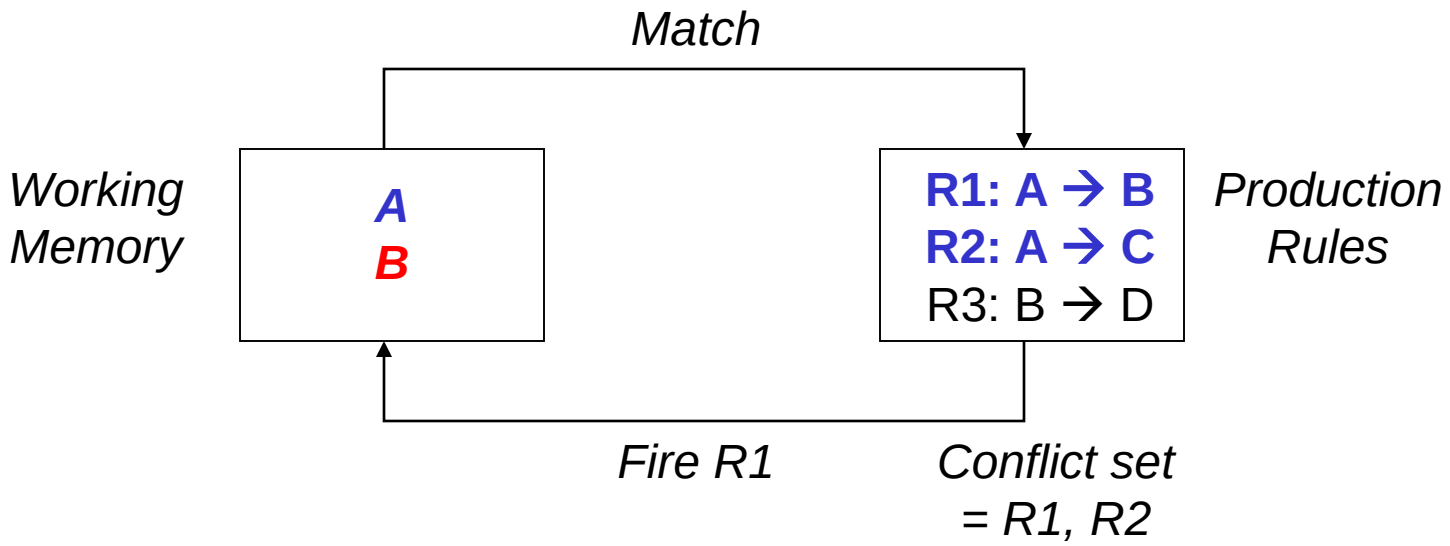


Production Systems

- Recognize-act cycle
  - Match
    - Patterns in working memory are matched against rules
      → Conflict set
  - Conflict resolution
    - One of the rules in the conflict set is selected
  - Fire
    - Action is performed. Working memory is updated

Production Systems

- Example

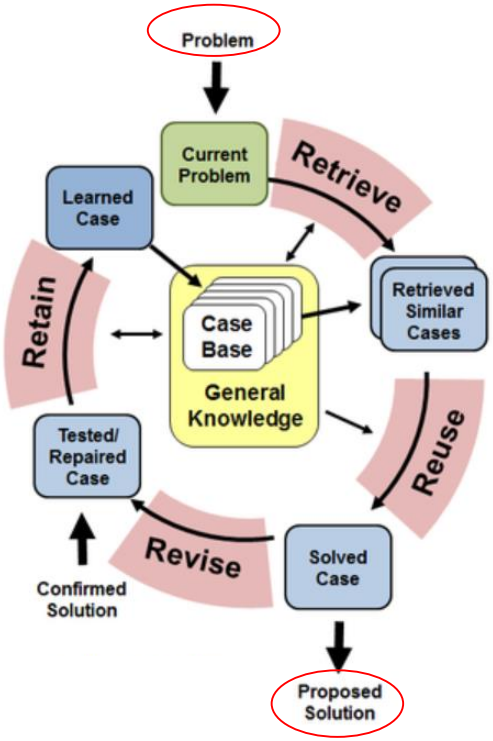


Case-Based Reasoning

- Reasoning from cases
  - Case: examples from past
    - Ex> Law cases
  - Reasoning procedure

    For a new problem,

    1. Retrieve similar case (based on common feature)
    2. Reuse the case: modify the retrieved case to get the solution
    3. Revise the case
    4. Retain: save the new case



Case-Based Reasoning

- Example

customer	sex	age	action?(f)
c1	M	40	N
c2	M	20	Y
c3	F	30	N
c4	M	30	Y

c5: F, 20 → Y or N?

  - Rule-based
    - Rule: if (sex=F) then N. Therefore N
  - Case-based
    - C3 is most similar to c5. Therefore N

Case-Based Reasoning

- k-Nearest Neighbor method
  1. Define distance functions
  - Example
    - $d_{sex}(A, B) = |A - B|$ (female:0, male:1)
    - $d_{age}(A, B) = |A - B|$ / max difference
    - $d = d_{sex} + d_{age}$

    - $d_{sum}(c5, c1) = |0 - 1| + |20 - 40| / 20 = 2.0$
    - $d_{sum}(c5, c2) = |0 - 1| + |20 - 20| / 20 = 1.0$
    - $d_{sum}(c5, c3) = |0 - 0| + |20 - 30| / 20 = 0.5$
    - $d_{sum}(c5, c4) = |0 - 1| + |20 - 30| / 20 = 1.5$

Case-Based Reasoning

2. Predict value from neighbors
   - Weighted average of neighbor values f_i (Y = +1, N = -1)

$$Let \quad w_i = \frac{1}{d_{ij}}, \quad W = \sum w_i$$

$$f_j = \sum f_i \cdot \frac{w_i}{W}$$

- Example
  - 3-NN → c3, c2, c4

    c3: f_3 = -1(N), d_{35} = 0.5, w_3 = 2.0

    c2: f_2 = +1(Y), d_{25} = 1.0, w_2 = 1.0

    c4: f_4 = +1(Y), d_{45} = 1.5, w_4 = 0.67
  - f_5 = [(-1*2.0) + (1*1.0) + (1*0.67)] / 3.67 = -0.09 → N

Case-Based Reasoning

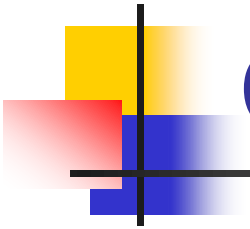
- Example: breast cancer classification
  - Case: represented by 9 attributes
  - Class: recurrence / no-recurrence

Viewer

Relation: breast-cancer

No.	age Nominal	menopause Nominal	tumor-size Nominal	inv-nodes Nominal	node-caps Nominal	deg-malig Nominal	breast Nominal	breast-quad Nominal	irradiat Nominal	Class Nominal
1	40-49	premeno	15-19	0-2	yes	3	right	left_up	no	recurrence-events
2	50-59	ge40	15-19	0-2	no	1	right	central	no	no-recurrence-events
3	50-59	ge40	35-39	0-2	no	2	left	left_low	no	recurrence-events
4	40-49	premeno	35-39	0-2	yes	3	right	left_low	yes	no-recurrence-events
5	40-49	premeno	30-34	3-5	yes	2	left	right_up	no	recurrence-events
6	50-59	premeno	25-29	3-5	no	2	right	left_up	yes	no-recurrence-events
7	50-59	ge40	40-44	0-2	no	3	left	left_up	no	no-recurrence-events
8	40-49	premeno	10-14	0-2	no	2	left	left_up	no	no-recurrence-events
9	40-49	premeno	0-4	0-2	no	2	right	right_low	no	no-recurrence-events
10	40-49	ge40	40-44	15-17	yes	2	right	left_up	yes	no-recurrence-events
11	50-59	premeno	25-29	0-2	no	2	left	left_low	no	no-recurrence-events
12	60-69	ge40	15-19	0-2	no	2	right	left_up	no	no-recurrence-events
13	50-59	ge40	30-34	0-2	no	1	right	central	no	no-recurrence-events
14	50-59	ge40	25-29	0-2	no	2	right	left_up	no	no-recurrence-events
15	40-49	premeno	25-29	0-2	no	2	left	left_low	yes	recurrence-events
16	30-39	premeno	20-24	0-2	no	3	left	central	no	no-recurrence-events
17	50-59	premeno	10-14	3-5	no	1	right	left_up	no	no-recurrence-events
18	60-69	ge40	15-19	0-2	no	2	right	left_up	no	no-recurrence-events
19	50-59	premeno	40-44	0-2	no	2	left	left_up	no	no-recurrence-events
20	50-59	ge40	20-24	0-2	no	3	left	left_up	no	no-recurrence-events

Undo OK Cancel



Case-Based Reasoning

- Issues
  - Selection of features
  - Computing similarity
- Advantages
  - Knowledge acquisition can be simplified
  - Provide learning capability to expert systems
- Disadvantages
  - Lack of explanation capability