NOT COVERED IN DETAIL

**Production Systems**

(Rule-Based Systems, Expert Systems)

- Based on observation that people seem to use "rules of thumb" in solving problems;

  e.g., (in diagnosis)
  
  IF person x has a sore throat and x has a runny nose
  THEN x probably has a cold.

  e.g., (in symbolic integration)
  
  IF trying to find \( \int \frac{f(x)}{\sqrt{1-x^2}} \, dx \) (for some function f)
  THEN try to find \( \left[ f(\sin \theta) \, d\theta \right]_{\theta = \arcsin x} \)

- This approach to representing knowledge & "expertise" was originally proposed as a model of cognition in AI (Newell & Simon, 1960s & 70s) & later became a very successful technology for AI applications (medical diagnosis, organic chemistry, oil geology, etc.)

**Essential Ideas of Production/Rule-Based Systems**

- Production systems provide a syntax & operational semantics for sets of IF-THEN rules.

- The rules operate on a "blackboard" or "working memory" (WM) containing proposition-like attribute-value tuples called "working memory elements" (WMEs).

- The IF-parts of rules are matched against current WMEs (tuples), and the THEN-parts modify the WMEs (adding, changing, or deleting WMEs).
Syntax of Working Memory Elements (WMEs)  
as per Brachman & Levesque (more or less!)

\[
\langle \text{WME} \rangle ::= (\langle \text{type} \rangle, \langle \text{attribute}_1 \rangle; \langle \text{value}_1 \rangle, \ldots, \langle \text{attribute}_n \rangle; \langle \text{value}_n \rangle) \quad n \geq 0
\]

\[
\langle \text{type} \rangle ::= \text{person} \mid \text{goal} \mid \text{student} \mid \text{patient} \mid \text{bacterium} \mid \text{antibiotic} \mid \ldots
\]

\[\text{i.e., a monadic predicate constant}\]

\[
\langle \text{attribute} \rangle ::= \text{name} \mid \text{age} \mid \text{blood-type} \mid \text{morphology} \mid \ldots
\]

\[\text{i.e., a 1-place function constant}\]

\[
\langle \text{value} \rangle ::= \text{John} \mid johh \mid 23 \mid 0.8 \mid \text{csc244/444} \mid \ldots
\]

\[\text{i.e., an individual constant or numeric constant} \quad \& \text{maybe quoted entities like}\]

\[\text{"Joe Hill", } (\text{integral} (\sqrt{-1 (x \times x)})) x \quad ?\]

\[\text{e.g., } (\text{person } \text{age: } 23 \text{ occupation: student}) \quad \text{value}\]

\[\text{student name: Joe-Hill dept: csc} \quad \text{unique}\]

\[\text{goal wff: } (\text{at Robbie Door3 priority: 2})\]

\[\text{MEANING OF A WME IS AN EXISTENTIAL FORMULA}\]

\[\text{e.g., } \exists x [\text{person}(x) \land \text{age}(x) = 23 \land \text{occupation}(x) = \text{student}]\]

Syntax of Production Rules

\[
\langle \text{IF-THEN rule} \rangle ::= \text{IF} \langle \text{cond}_1 \rangle \ldots \langle \text{cond}_m \rangle \quad m, n \geq 1
\]

\[\text{THEN} \langle \text{action}_1 \rangle \ldots \langle \text{action}_n \rangle\]

\[
\langle \text{cond} \rangle ::= (\langle \text{type} \rangle, \langle \text{attribute}_1 \rangle; \langle \text{value-spec}_1 \rangle, \ldots, \langle \text{attribute}_k \rangle; \langle \text{value-spec}_k \rangle) \quad k \geq 0
\]

\[\text{only some of the attributes of } \langle \text{type} \rangle \text{ might be specified}\]

\[\langle \text{action} \rangle \text{ negated condition}\]

\[\text{e.g., } (\text{type}, \langle \text{attribute} \rangle \text{ defined as in WMEs}\]

\[\langle \text{value-spec} \rangle ::= (\langle \text{definite-spec} \rangle \mid \langle \text{test} \rangle)\]

\[\langle \text{definite-spec} \rangle ::= \langle \text{value} \rangle \mid \langle \text{variable} \rangle \mid \langle \text{evaluable-exp} \rangle\]

\[\langle \text{variable} \rangle ::= x \mid y \mid z \mid x \times 1 \ldots \quad \text{flagged as variables}\]

\[\langle \text{test} \rangle ::= \text{odd-integer} \mid \leq 5 \mid \leq x \mid \text{one-of (John Mary) } \ldots\]

\[\text{evaluable monadic predicates on values}\]

\[\langle \text{evaluable-exp} \rangle ::= x + y \mid \text{first-letter-of } x \mid \ldots\]

\[\text{i.e., an expression which (for a given}\]

\[\text{binding of the variables, if any)}\]

\[\text{evaluates to an attribute value}\]

\[
\langle \text{action} \rangle ::= \text{ADD} \langle \text{pattern} \rangle \mid \text{REMOVE} \langle \text{i} \rangle\]

\[
\text{MODIFY} \langle \text{attribute} \rangle \langle \text{definite-spec} \rangle \quad \text{(repeated & continued next page)}\]
Syntax of Production Rules (cont'd)

\[ (\text{action}) ::= \text{ADD} (\text{pattern}) \mid \text{REMOVE} i \mid \text{MODIFY} i (\langle \text{attribute}, \langle \text{definite-spec} \rangle \rangle) \]

\[ (\text{pattern}) ::= \langle \text{type} \rangle \langle \text{attribute} \rangle; \langle \text{definite-spec} \rangle; \ldots \]

\[ \langle \text{attribute} \rangle; \langle \text{definite-spec} \rangle \]

**Operational Semantics:**

- Find all matches of the rule set to the WM
  (Each match to the IF-part of a rule uniquely binds its variables, if any, and the values can be used in the THEN-part.)
- Choose one or more of the matches (conflict resolution)
- Execute the THEN-parts of the chosen rules/matches

---

**Example:**

- Diagnosis of infections (with certainty factors, CF)

<table>
<thead>
<tr>
<th>IF</th>
<th>THEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>(organism1 entry-portal: GI-tract entry-portal: CF: x morphology: rod stain: gram-neg aerobicity: anaerobic)</td>
<td>MODIFY 1 (identity: bacteroides identity-CF: [1.7*(1+2*x)])</td>
</tr>
</tbody>
</table>

**Additional Examples**

1. **Forward Inference**

   Elaborating family relationships, detecting inconsistencies

   a. \( \forall x \forall y. \text{Siblings}(x,y) \Rightarrow \text{Siblings}(y,x) \)
   b. \( \forall x \forall y. [(\text{Siblings}(x,y) \land \text{Male}(x))] \equiv \text{Brother-of}(x,y) \)
   c. \( \forall x \forall y. [(\text{Siblings}(x,y) \land \text{Female}(y))] \Rightarrow \text{Sister-of}(x,y) \)
   d. \( \forall x \forall y. \text{Spouses}(x,y) \Rightarrow \text{Directly-related}(x,y) \)
   e. \( \forall x \forall y. \text{Spouses}(x,y) \Rightarrow \text{Spouses}(y,x) \)
   f. \( \forall x \forall y. \text{Siblings}(x,y) \Rightarrow \text{Directly-related}(x,y) \)
   g. \( \forall x \forall y. \text{Parent}(x,y) \Rightarrow \text{Directly-related}(x,y) \)
   h. \( \forall x \forall y. \text{Parent}(x,y) \Rightarrow [(\text{Father-of}(x,y) \lor \text{Mother-of}(x,y))] \)
   i. \( \forall x \forall y \forall z. [(\text{Parent}(x,y) \land \text{Parent}(x,z))] \Rightarrow \text{Siblings}(y,z) \)

   etc.

So for example we can see that the following are inconsistent (from (d), (f), (h), & (i)):

- Mother-of(Sandy, Dana), Mother-of(Sandy, Kim), Spouses(Dana, Kim).
Possible encoding of (i) as a production system:

**IF** (spouses name1: x name2: y)
**THEN ADD** (kinds name1: x name2: y yes-no: no)

**IF** (siblings name1: x name2: y)
**THEN ADD** (kinds name1: x name2: y yes-no: yes)

**IF** (kinds name1: x name2: y yes-no: yes)
(kinds name1: x name2: y yes-no: no)
**THEN ADD** (inconsistency name1: x name2: y)

**IF** (parent name1: x name2: y)
(parent name1: x name2: z)
**THEN ADD** (siblings name1: y name2: z)

**IF** (mother name1: x name2: y)
**THEN ADD** (parent name1: x name2: y)

N.B.: Saying that being a parent implies being a mother OR a father (forward direction of (N)) is somewhat difficult. We might use WMEs (parent name1: x, gend1: g, name2: y, gend2: g) instead of WMEs (mother,...), (father,...). Then if gend1, gend2 values can only be "male" or "female", the mother/father disjunction is implicit.

---

2. Goal-directed inference

Suppose we want to prove family relationships, such as that "Dana" & "Kim" are siblings (see previous examples).

**IF** (goal rel: siblings name1: x name2: y status: active)
(fact rel: mother name1: z name2: x)
**THEN ADD** (goal rel: mother name1: z name2: y status: active)
**MODIFY 1** (status: suspended)

**IF** (goal rel: r name1: x name2: y status: active)
(fact rel: r name1: name2: y)
**THEN MODIFY 1** (status: proved)

**IF** (goal status: proved)
(goal status: suspended)
- (goal status: active)
**THEN MODIFY 2** (status: proved)

etc.

---

(Hmm, I think we need goal ID's for this to work properly - unless we never have more than one goal at a time)
3. Action: Placing blocks in order of size...
   see text p.122-4

   3 blocks, named A, B, C, of sizes 10, 20, 30, originally loc. at "heap"
   are to be placed at positions 1, 2, 3 resp.

   $\begin{array}{c}
   \text{A} \\
   \text{B} \\
   \text{C}
   \end{array} \rightarrow
   \begin{array}{c}
   \text{B} \\
   \text{C} \\
   \text{A}
   \end{array}$

   Repeat: Place biggest block from "heap" at i
   and increment i (initial i = 1)

4. Determining if a given year is a leap year (has 366 days rather than 365)...
   see text p.125-6

   Note: The rough rule is that years 0, 4, 8, 12, ... are leap years; but centennial years (i.e., ending in 00)
   that are not multiples of 400 are an exception (i.e., not leap years, despite being divisible by 4).

   The text uses arithmetic modulo 400 & modulo 4 to solve the problem for a given year, e.g., year 2000, or 1980, etc

---

Conflict Resolution

- use in order of presentation (cf. prolog)
- most specific first (i.e., more attributes and/or more cond's)
- recency (breadth-1st or depth-1st)
- refractoriness (avoid repetition of rule instance)

Improving Efficiency

RETE algorithm (L. Forgy '74)

- use a network where each node tests
  for a type or an attribute value
  rules that have a condition matched by
  pass, new or modified WM's through
  the network as far as possible
- when they reach a terminal node, a rule is known to be instantiated

Applications

- MYCIN - infectious organism diagnosis & treatment 1976
- XCON - DEC systems configurations
- PROSPECTOR - petroleum prospecting
- HEURISTIC DENDRAL - spectrochem analysis
- SOAR (J. Laird)