PRODUCTION SYSTEMS

(RULE-BASED SYSTEMS, EXPERT SYSTEMS)

eg. SOAR (Laird Rosenbloom)

- 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 $\left[\int f(\sin\theta)\,d\theta\right]_{\theta=\arcsin x}$.

- 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 <u>syntax</u> & <u>operational semantics</u> 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!)

⟨WME⟩ ::= (⟨type⟩ ⟨attribute⟩$_1$: ⟨value⟩$_1$ ... ⟨attribute⟩$_n$: ⟨value⟩$_n$) $n \geq 0$

⟨type⟩ ::= person | goal | student | patient | bacterium | antibiotic | ...
i.e., a monadic predicate constant

⟨attribute⟩ ::= name | age | blood-type | morphology | ...
i.e., a 1-place function constant

⟨value⟩ ::= John | john | 23 | 0.8 | CSC244/444 | ...
i.e., an individual constant or numeric constant (& maybe quoted entities like "Joe Hill", '(integral (/ 1 (sqrt (- 1 (* x x)))) x) ?)

e.g., (person age: 23 occupation: student) — value
predicate — (student name: Joe-Hill dept: CSC) ... unique
(goal wff: '(at Robbie Door3 priority: 2)

MEANING OF A WME IS AN EXISTENTIAL FORMULA

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

Syntax of Production Rules

These generally specify different WME types & so match different WMEs

⟨IF-THEN rule⟩ ::= IF ⟨cond⟩$_1$ ... ⟨cond⟩$_m$ THEN ⟨action⟩$_1$ ... ⟨action⟩$_n$ $m,n \geq 1$

⟨cond⟩ ::= (⟨type⟩ ⟨attribute⟩$_1$: ⟨value-spec⟩$_1$ ... ⟨attribute⟩$_k$: ⟨value-spec⟩$_k$) $k \geq 0$ |
−(⟨cond⟩) negated condition

only some of the attributes of ⟨type⟩ might be specified

⟨type⟩, ⟨attribute⟩ defined as in WMEs

⟨value-spec⟩ ::= ⟨definite-spec⟩ | {⟨test⟩}

⟨definite-spec⟩ ::= ⟨value⟩ | ⟨variable⟩ | [⟨evaluable-expr⟩]

1st match to a value-spec fixes the value

⟨variable⟩ ::= x | y | z | x1 | ... flagged as variables

⟨test⟩ ::= odd-integer | ≤5 | ≤x | one-of (John Mary) | ... | ¬⟨test⟩ | ⟨test⟩∧⟨test⟩ | ⟨test⟩∨⟨test⟩

evaluable monadic predicates on values

⟨evaluable-expr⟩ ::= x+y | first-letter-of x | ...
i.e., an expression which (for a given binding of the variables, if any) evaluates to an attribute value

⟨action⟩ ::= ADD ⟨pattern⟩ | REMOVE i |
MODIFY i (⟨attribute⟩ ⟨definite-spec⟩)

(repeated & continued next page)

Syntax of Production Rules (cont'd)

⟨action⟩ ::= ADD ⟨pattern⟩ | REMOVE i | MODIFY i (⟨attribute⟩ ⟨definite-spec⟩)

remove WME matching ⟨cond⟩$_i$

change specified attribute of ⟨cond⟩$_i$

⟨pattern⟩ ::= (⟨type⟩ ⟨attribute⟩$_1$: ⟨definite-spec⟩$_1$ ... ⟨attribute⟩$_k$: ⟨definite-spec⟩$_k$)

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

— * * * * —

e.g., diagnosis of infections (with certainty factors - CFs)

IF (organism1 entry-portal: GI-tract entry-portal-CF: x
morphology: rod stain: gram-neg
aerobicity: anaerobic)
(culture-site-org1 type: {= sterile V = blood})
THEN MODIFY 1 (identity: bacteroides
identity-CF: [.7 * (1 + .2 * x)])

Additional Examples

1. Forward Inference

Elaborating family relationships, detecting inconsistencies

a. $\forall x \forall y.\ Siblings(x,y) \Rightarrow Siblings(y,x)$

b. $\forall x \forall y.\ [Siblings(x,y) \wedge Male(x)] \Leftrightarrow Brother\text{-}of(x,y)$

c. $\forall x \forall y.\ [Siblings(x,y) \wedge Female(x)] \Leftrightarrow Sister\text{-}of(x,y)$

d. $\forall x \forall y.\ Spouses(x,y) \Rightarrow \neg Directly\text{-}related(x,y)$

e. $\forall x \forall y.\ Spouses(x,y) \Rightarrow Spouses(y,x)$

f. $\forall x \forall y.\ Siblings(x,y) \Rightarrow Directly\text{-}related(x,y)$

g. $\forall x \forall y.\ Parent(x,y) \Rightarrow Directly\text{-}related(x,y)$

h. $\forall x \forall y.\ Parent(x,y) \Leftrightarrow [Father\text{-}of(x,y) \vee Mother\text{-}of(x,y)]$

i. $\forall x \forall y \forall z.\ [Parent(x,y) \wedge Parent(x,z)] \Rightarrow Siblings(y,z)$

... 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:

(d)
```
IF (spouses name1: x name2: y)
THEN ADD (kins name1: x name2: y yes-no: no)
```

(f)
```
IF (siblings name1: x name2: y)
THEN ADD (kins name1: x name2: y yes-no: yes)
```

```
IF (kins name1: x name2: y yes-no: yes)
   (kins name1: x name2: y yes-no: no)
THEN ADD (inconsistency name1: x name2: y)
```

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

part of (h)
```
IF (mother name1: x name2: y)
THEN ADD (parent name1: x name2: y)
```
...etc.

N.B.: Saying that being a parent implies being a mother OR a father (forward direction of (h)) is somewhat difficult. We might use WMEs (parent name1: x_1 gend1: g_1 name2: x_2 gend2: g_2) 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: x name2: y)
THEN MODIFY 1 (status: proved)
```

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

no more unproven subgoals

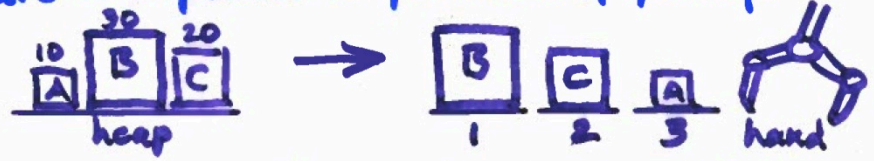
declare all suspended goals proven.

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 3, 1, 2 resp.



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 1900, etc

Conflict Resolution p126-7

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

Improving Eficiency p127-8

RETE algorithm (C. Forgy '74)

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

Applications 130-132

SOAR (J. Laird)

MYCIN - infectious organism diagnosis & treatment 1976

XCON - DEC systems configurations

PROSPECTOR - petroleum prospecting

HEURISTIC DENDRAL - spectrochem. analysis

etc