AI Lecture. Prof. Carolina Ruiz. Worcester Polytechnic Institute

RULE–BASED SYSTEMS

(also known as production systems)

Knowledge Representation: Rules and assertions

Problem Solving: forward and backward chaining

```
-------------------------
|Inference Engine        |
|                        |
|- Forward Chaining, or  |
|- Backward Chaining     |
|-------------------------
|Knowledge Base:         |
|                        |
|- Rule base       (RB)  |
|- Working Memory  (WM)  |
-------------------------
```

KNOWLEDGE BASE

Rule:

IF list of antecedents THEN consequent
(IF LHS THEN RHS)

triggers: when all its antecedents are satisfied

fires: when its consequent is executed or added to the working memory.

Working Memory: Set of assertions.

Handling Variables:

Variable binding: a substitution of a variable for a constant

Instantiation: a (consistent) substitution of all occurrences of a variable in a rule by its binding.

FORWARD CHAINING

(Also called bottom-up approach)

Use known assertions and rules to deduce additional assertions.

```
-->Try to match ALL LHS's against WM
|
|  Any match?  (no) ---> fails
|
|    (yes)
|
|  select rule(s) from "conflict" set
|  ("triggered" rules)
|  using conflict resolution strategy
|
|
|  execute RHS ("fired" rules)
|
|
|  stop?     (yes) -----> done
|    |
----| (no)
```

BACKWARD CHAINING

(Also called top-down approach)

Work backward from the desired goal by finding rules that could deduce it; then try to deduce the premises of those rules.

DEDUCTIVE SYSTEMS

Rules: antecedents and consequent are assertions (logical statements)

Inference: Can use forward or backward chaining.

Example:

Rule Base: (RB)

D1. IF ?x has fever AND ?x has upset stomach
THEN ?x has flu

D2. IF ?x's temperature > 98 THEN ?x has fever

D3. IF ?x has flu THEN ?x needs rest

D4. IF ?x has fever AND ?x has itchy eyes
AND it is pollen season
THEN ?x has hay fever

Working Memory: (WM)

A1. It is pollen season

A2. John's temperature is 101

A3. Mary has fever

A4. Mary has upset stomach

REACTIVE SYSTEMS

Rules: consequents are actions and antecedents are conditions under which the actions can be executed.

Inference: forward chaining with a conflict resolution strategy.

Example:

Rule Base: (RB)

R1. IF ?x has fever AND ?x's age is ≤ 6
THEN hospitalize ?x

R2. IF ?x has fever THEN prescribe Tylenol

R3. IF ?X has fever AND ?x is pregnant
THEN prescribe Advil

R4. IF ?x is male THEN add(?x is not pregnant)

Working Memory: (WM)

A1. John's age is 28

A2. Mary is pregnant

A3. Mary's temperature is 99

SOME CONFLICT RESOLUTION

Random: use any rule

Rule ordering: use rules in the order given

Context limiting: group rules by contexts and use one in the present context

Specificity ordering: use the rule that handles the most specific situation

Size ordering: use rule with longest list of conditions

Data recency: use rule that uses most recently proven assertion

Rule recency: use the least recently used rule

IMPLEMENTING FORWARD CHAINING

Goals: We'll try two:

G1. Is Mary happy?

G2. Prove all you can.

Rule Base: (RB)

R1. IF ?x is rich THEN ?x is happy

R2. IF ?x loves ?y AND ?y loves ?x THEN ?x is happy

R3. IF ?x loves Jack THEN Jack loves ?x

Working Memory: (WM)

A1. Bob is rich

A2. Jack is rich

A3. Mary loves Jack

A4. Mary loves Bob

A5. Bob is happy

A6. Jack is happy

A6. Jack loves Mary

A8. Mary is happy

```
-------------------------------------------------------------------------
Using R1.
                                        ?x is rich
?x (Bob) is rich       ?x (Jack) is rich

succeeds            succeeds
WM U {Bob is happy} WM U {Jack is happy}

-------------------------------------------------------------------------
Using R2.
                                                        ?x loves ?y
?x (Mary) loves ?y (Jack)   ?x (Mary) loves ?y (Bob)
                                                        ?y loves ?x
?y (Jack) loves ?x (Mary)   ?y (Bob) loves ?x (Mary)

fails                       fail
-------------------------------------------------------------------------
Using R3.

                                        ?x loves Jack
?x (Mary) loves Jack

succeeds
WM U {Jack loves Mary}
-------------------------------------------------------------------------
Using R2.

                                                                ?x loves ?y

?x (Mary) loves ?y (Jack)  ?x (Jack) loves ?y (Mary) ?x (Mary) loves ?y (Bob)

                                                                ?y loves ?x

?y (Jack) loves ?x (Mary)  ?y (Mary) loves ?x (Jack) ?y (Bob) loves ?x (Mary)

succeeds                   succeeds                  fails
WM U {Mary is happy}       WM U {Jack is happy}
-------------------------------------------------------------------------
```

OBSERVATIONS

- Each path corresponds to a set of biding commitments
- Each binding for an antecedent produces a branch
- Depth of tree = # of antecedents in the rule
- Inference mechanism is a search! Here we explore the tree in a depth 1st manner (other ways may work too.)

IMPLEMENTING BACKWARD CHAINING

Goals: We'll try two:

G1. Is Mary happy? (conjecture)

G2. Who is happy?

Rule Base: (RB)

R1. IF ?x is rich THEN ?x is happy

R2. IF ?x loves ?y AND ?y loves ?x THEN ?x is happy

R3. IF ?x loves Jack THEN Jack loves ?x

Working Memory: (WM)

A1. Bob is rich

A2. Jack is rich

A3. Mary loves Jack

A4. Mary loves Bob

A5. Jack loves Mary

A6. Mary is happy

```
-----------------------------------------------------------------------
G1. Is Mary happy?                                       (using R1)

                                        ?x is happy
?x (Mary) is happy
                                        ?x is rich
?x (Mary) is rich

fails
.........................................................................
                                                        (using R2)

                                                        ?x is happy
                  ?x (Mary) is happy
                                                        ?x loves ?y

?x (Mary) loves ?y (Jack)   ?x (Mary) loves ?y (Bob)
                            (this branch is not even tried)
                                                        ?y loves ?x

?y (Jack) loves ?x (Mary)   ?y (Bob) loves ?x (Mary)

                            fails
.........................................................................
                                                        (using R3)

                                                         ?x loves Jack

?x (Mary) loves Jack

succeeds! Hence Mary is happy!

WM U {Jack loves Mary, Mary is happy}

-----------------------------------------------------------------------
```

```
G2.  Who is happy?
                                                 ?x is happy

                     ?x is happy

...............................................................................
(using R1)                                                      (using R2)

      ?x is rich                                ?x loves ?y

?x(Bob) is rich ?x(Jack) is rich ?x(Mary) loves ?y(Jack) ?x(Mary) loves ?y(Bob)

(1)              (2)              ?y(Jack) loves ?x(Mary) ?y(Bob) loves ?x(Mary)
Succeeds         Succeeds
Bob is happy!    Jack is happy!                           fails (4)
                                 ..............................................
                                                                (using R3)
                                 ?x(Mary) loves ?y(Jack)

                                 (3)
                                 Succeeds
                                 Mary is happy!

-------------------------------------------------------------------------
```

ADVANTAGES OF RBS

- RBS have good explanation capabilities: How and why questions (good because people will trust AI–based systems)
- Modularity:

  Rules are units that work in concert (chaining) but are easy to update, delete, or insert.
- Model to some extent how humans reason Using rules to model cognition
- Rules encode problem reduction strategy

COGNITIVE MODELING

DEFAULTS AND EXCEPTIONS:

```
IF ?x is rich THEN ?x is happy
              PROVIDED ?x is a person

IF ?x is rich THEN ?x is happy
              UNLESS ?x is ill
```

Modes	PROVIDING assumptions	UNLESS assumptions
Show-me	prove	
Ask-questions later	ignore	
Decision-maker	assume true	assume false (except -><- WM)
Trusting-Skeptic	assume true	assume true
	(except when -><- 1 rule)	
Progressive-Reliability	ignore	ignore
	(explore as time permits)	

COGNITIVE MODELING (cont.)

UNCERTAINTY REASONING

Certainty Factors (values between 0 and 1) measure how reliable an answer is

```
IF ?x's temperature > 98
THEN ?x has fever
with CF =1 (true 100% of the time)

IF ?x has fever AND ?x has upset stomach
THEN ?x has flu
with CF = 0.6 (true 60% of the time)
```

DISADVANTAGES OF RBS

- May behave like idiot savants:
  They do their task well but do not reason behind their rules.

KNOWLEDGE ACQUISITION

How to obtain rules

- from human experts: Knowledge Engineering
- from non-human experts: (inductive) machine learning