Search for Matching Rules

1

Example (Using slightly non-python notation)

- Each time a rule is applied, a new depth-first search is started, to find all the ways its antecedents match facts in working memory, i.e.,
- The search is exploring possible bindings for the variables in the rule
- Once these are found, the bindings are plugged into the consequent of the rule and the result added to working memory
- States have two parts:
  - antecedents still to be matched
  - a substitution (from how prior antecedents were matched)
- Search is depth-first: For a rule IF A B C then D, when a candidate binding is found for A, the search pursues candidate bindings for B before considering alternative bindings for A

2

- Consider R=
  if ((patient ?x) (spouse ?x ?y) (fever ?x high))
  then (quarantine ?x)(quarantine ?y)

WM = ((patient sue) (patient joe) (spouse sue fred)(fever sue high))

When starting a new rule, all antecedents need to be matched:

(((patient ?x) (spouse ?x ?y) (fever ?x high))

Initially, bindings are empty:

()

Thus initial state is: (((patient ?x) (spouse ?x ?y) (fever ?x high)) ()))

3

- First antecedent (patient) unifies with two items in WM, Sue and Joe
- Resulting states:

(antecedents-list	**bindings-list)**
(((spouse ?x ?y) (fever ?x high))	((patient sue)))
(((spouse ?x ?y) (fever ?x high))	((patient joe)))

- For depth-first search, these are placed at front of queue of states to explore

4