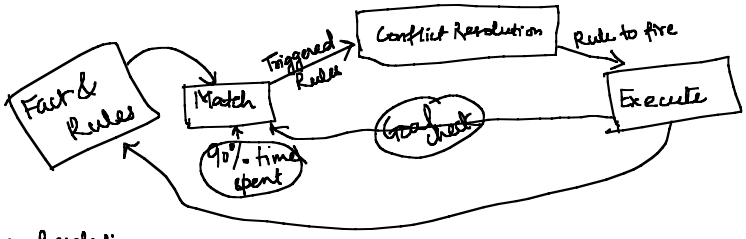
10/3/21



Conflict Resolution

→ The objective is to decide which triggered rules should be fired.

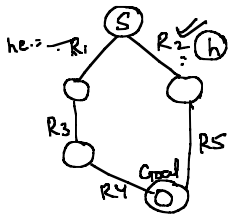
→ can use various strategies like -

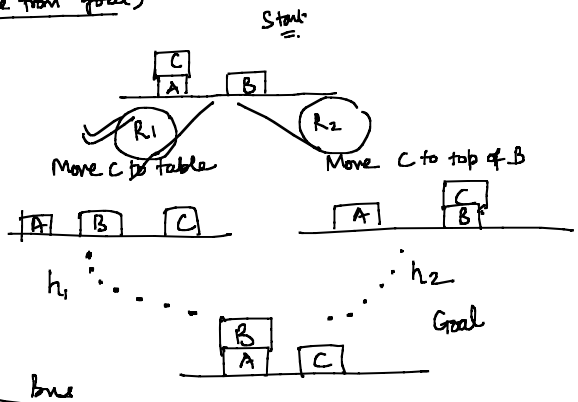
1. FCFS (first come first serve).
2. Specificity ordering.
3. Fire All
4. Heuristic measures (distance from goal)

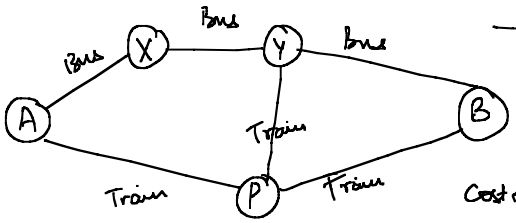
$P \rightarrow Q$: least specific

$P \wedge R \rightarrow Q$: more specific

✓ $P \wedge R \wedge S \rightarrow Q$: most specific





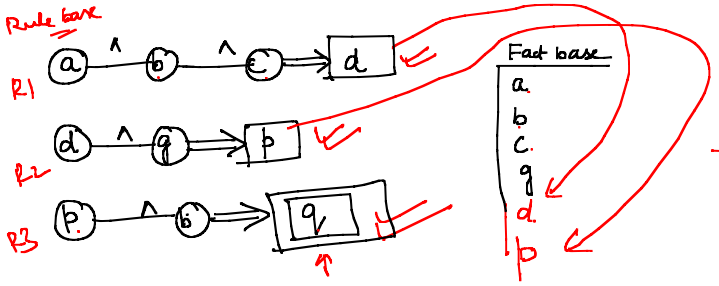


Cost func^n = $w_i(t) + w_j(c)$

{ t: time left.
{ c: cost to be paid

Other strategies :-

1. Refractoriness :- Rules once fired will not be fired again.
2. Meta - Rules :- Rules about Rules embedded within inference machine that provides the info^n about which rules apply under what conditions.



$P \rightarrow Q$ Modus Ponens.

P

Q

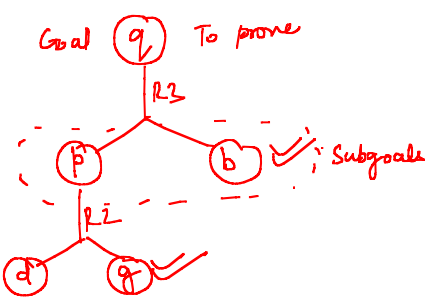
$a \wedge b \wedge c \rightarrow d$

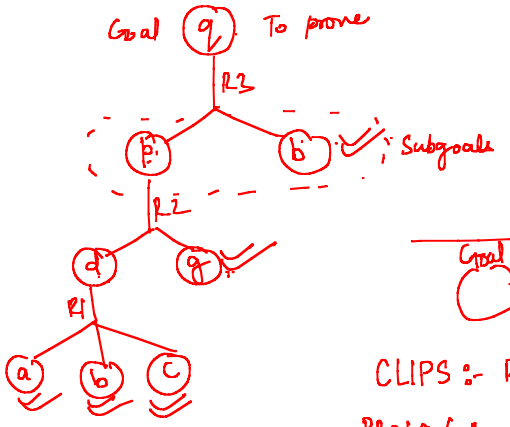
a

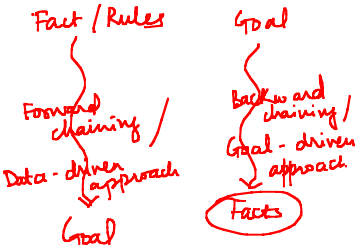
b

c

d







CLIPS :- Forward Chaining.

PROLOG :- Backward Chaining

Forward Chaining

- Based on Modus Ponen Rule.

  $P_1 \wedge \ldots \wedge P_n \rightarrow Q$

  $P_1 \ldots\ldots P_n$

  then we can conclude Q.

→ Starts from the given facts and try to reach the goal.

→ Data-driven search.

→ may lead to dead end and in such cases backtracking is required.

Chronological Intelligent

Backward Chaining

Starts from query and go backwards.

Starts from goal and try to prove goal using given facts.

→ Goal-driven Search.

R1: $a \wedge b \rightarrow c$

R2: $x \wedge y \rightarrow p$

R3: $p \wedge q \rightarrow$ goal

Fact.
a.
b
x
y
q