INFERENCE AND CONTROL

Inference engine performs 2 major tasks:

1) examines existing facts and rules and adds new facts when possible

2) decides the order in which inferences are made.

We shall look at Inference and Control

INFERENCE:

Infer means " to derive as a conclusion from facts or premises".
There are 2 common rules for deriving new facts from rules and known facts.These are Modus Ponens and Modus Tollens

MODUS PONENS

*most common inference strategy
*simple ,reaoning based on it is easily understood.

The rule states that when A is known to be true and if a rule states " If A then B " it is valid to conclude that B is true.

MODUS TOLLENS

When B is false rule If A, then B
then A is false.

E.g:
Rule : IF Ahmet's CAR IS DIRTY
THEN Ahmet HAS BEEN DRIVING OUTSIDE ANKARA

Given fact : Ahmet has not been outside Ankara.
New rule : Ahmet car is not dirty.

This conclusion seems quite obvious but cannot be reached by most expert systems. Because they use modus ponens for deriving new facts from rules.

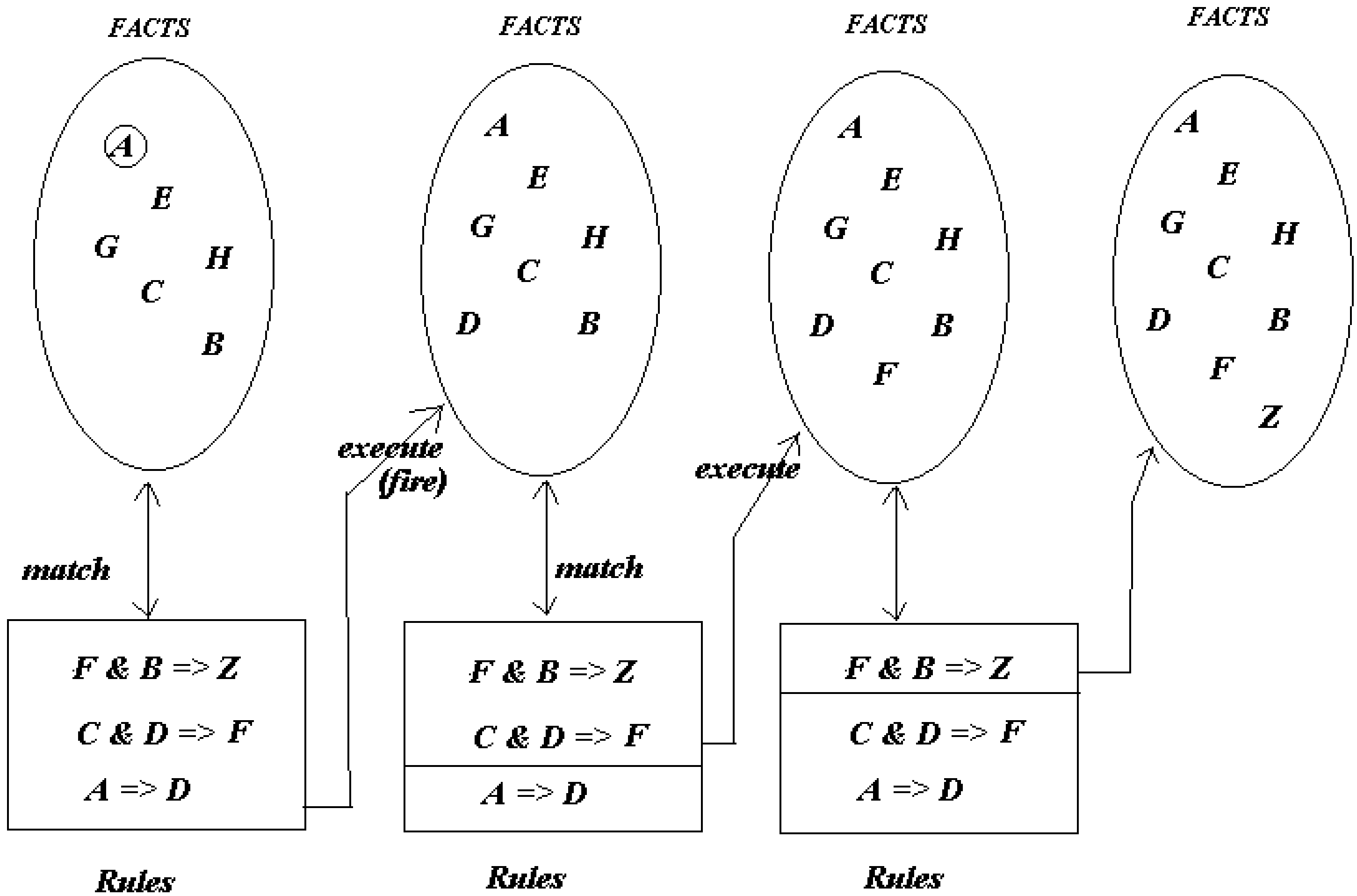
REASONING ABOUT UNCERTAINTY

An inference engine should be able to handle incomplete information. The degree of certainity is represented as a number of attached to a fact (certainty factor).

There are three inferencing methods. These are Forward,Backward and Mixed Chaining.

FORWARD CHAINING:

Forward Chaining(Example 1)



Problem: Does situation Z exists or not ?

The first rule that fires is A->D because A is already in the database. Next we infer D. Existence of C and D causes second rule to fire and as a consequence F is inferred and placed in the database. This in turn, causes

the third rule F?B->Z to fire, placing Z in the database.
This technique is called forward chaining.

A very simple Forward chaining Algorithm

Given m facts $F_1,F_2,....,F_m$? N RULES
$R_1,R_2,......R_n$

repeat
for i ?- 1 to n do
if one or more current facts match the antecedent of Ri then
1) add the new fact(s) define by the consequent
2) flag the rule that has been fired
3) increase m
until no new facts have been produced.

Forward Chaining (Example 2)

Rule 1
IF the car overheats , THEN the car will stall.
Rule 2
IF the car stalls
THEN it will cost me money
AND I will be late getting home

Now, the question is
How do you arrive at conclusion that this situation will cost money and cause you to be late ?
The condition that triggers the chain of events is the car overheating

BACKWARD CHAINING:

Backward Chaining (Example 1)

Rule 1
IF the car is not tuned AND the battery is weak
THEN not enough current will reach the starter.
Rule 2
IF not enough current reaches the starter
THEN the car will not start.

Given facts:
The car is not tuned
The battery is weak.
Now, the question is

How would you arrive at the conditions that have resulted in the car failing to start?

Backward Chaining(Example 2)

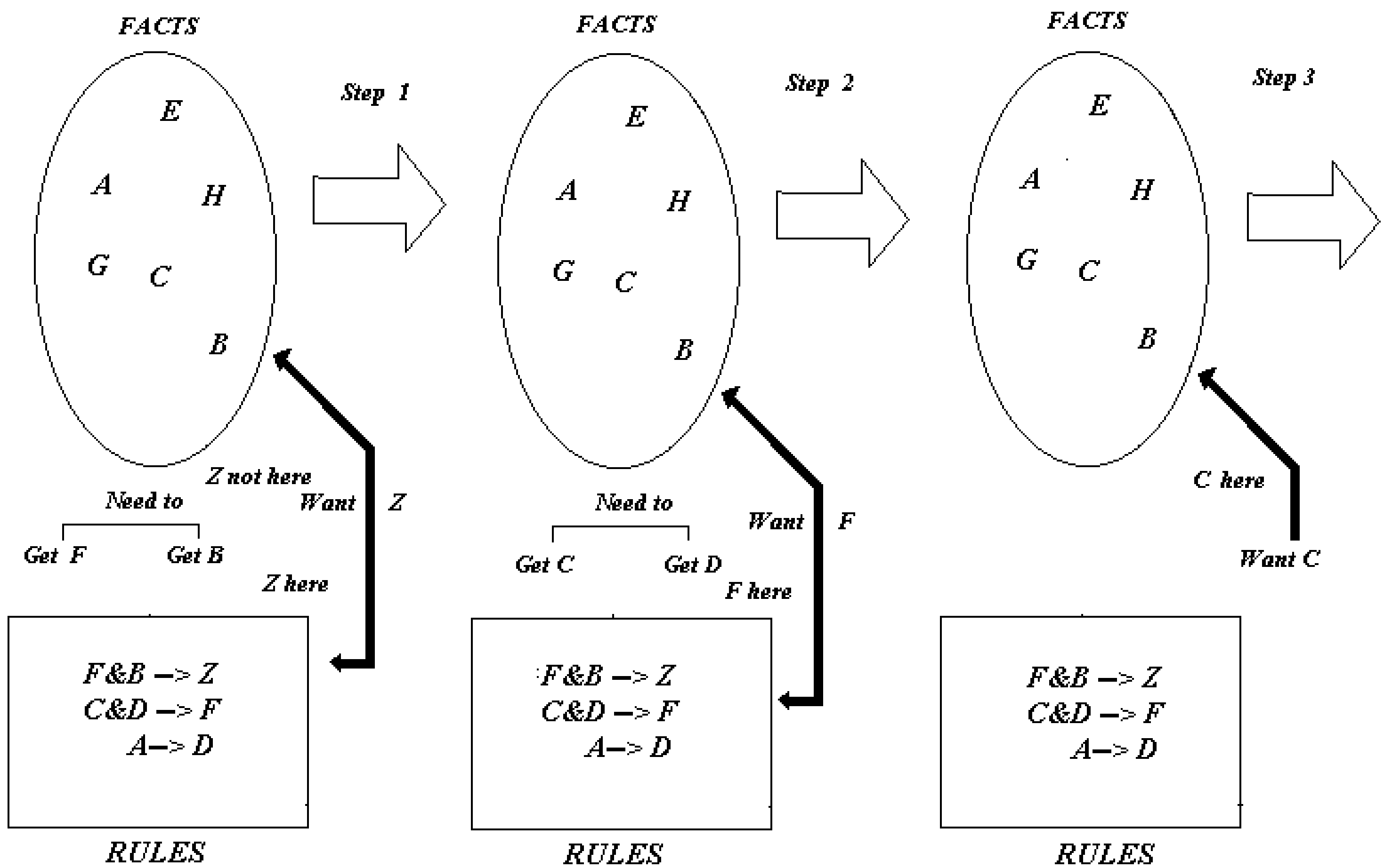
In such a situation backward chaining might be more cost-effective.With this inference method the system starts with what it wants to prove, e.g.,that situation Z exists, and only executes rules that are relavent to establishing it.Figure following shows how bacward chaining would work using the rules from the forward chaining example.

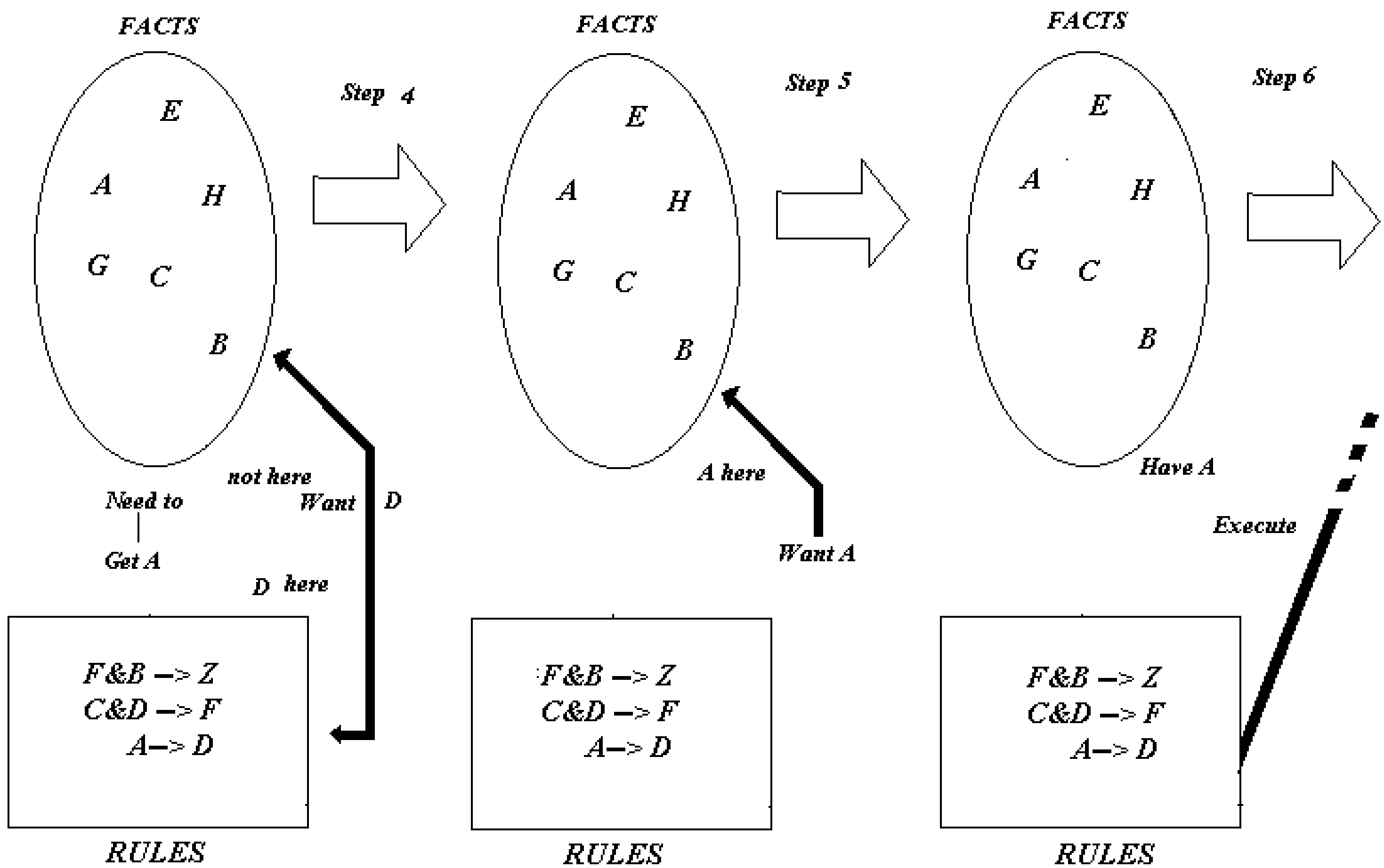
In step 1 the system is told to establish (if it can) that situation Z exists,It first checks the data base for Z,and when that fails, searches for rules that conclude Z,i.e., have Z on the right side of the arrow.It finds the rule F?B->Z, and decides that it must establish F and B in order to conclude Z.

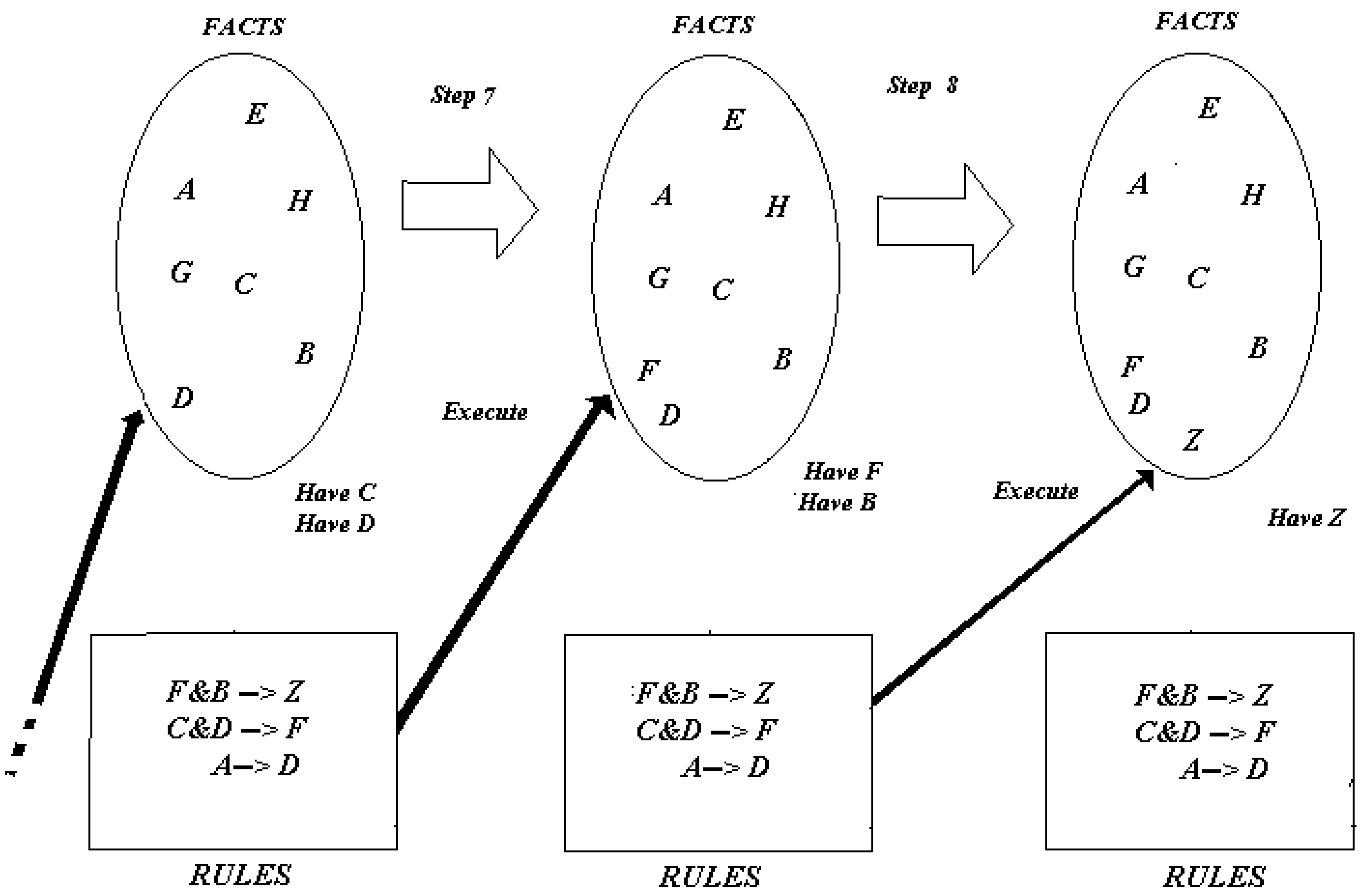
In step 2 the system tries to establish F, first checking the data base and then finding a rule that concludes F. From this rule, C?D->F, the system decides it must establish C and D to conclude F.

In steps 3 through 5 the system finds C in the data base but decides it must establish A before it can conclude D. It then finds A in the data base.

In steps 6 through 8 the system executes the third rule to establish D, then executes the second rule to establish the original goal, Z.The infenece chain created here is identical to the one created by forward chaining. The difference in two approaches hinges on the method in which data and rules are searched.


FACTS
E
A
H
G
C
B
Step 1
Z not here
Need to
Want Z
Get F
Get B
Z here
F&B --> Z
C&D --> F
A--> D
RULES
FACTS
E
A
H
G
C
B
Step 2
Need to
Want F
Get C
Get D
F here
F&B --> Z
C&D --> F
A--> D
RULES
FACTS
E
A
H
G
C
B
Step 3
C here
Want C
F&B --> Z
C&D --> F
A--> D
RULES


FACTS
E
A
H
G
C
B
Step 4
not here
Want D
Need to
Get A
D here
F&B –> Z
C&D –> F
A–> D
RULES
FACTS
E
A
H
G
C
B
Step 5
A here
Want A
F&B –> Z
C&D –> F
A–> D
RULES
FACTS
E
A
H
G
C
B
Step 6
Have A
Execute
F&B –> Z
C&D –> F
A–> D
RULES



. . .

7.5 MIXED CHAINING

Mixed Chaining (Example)

R₁. IF F and H then K } Suppose R_1-R_3 are
R₂. IF E and A then K } backward chaining
R₃. IF E and B then H }

R₄. IF A and G then B } R_4-R_8
R₅. IF B and D then H } are forward chaining
R₆. IF G and D then E }
R₇. IF A and B then D }

***R_8*.** IF A and C then G }

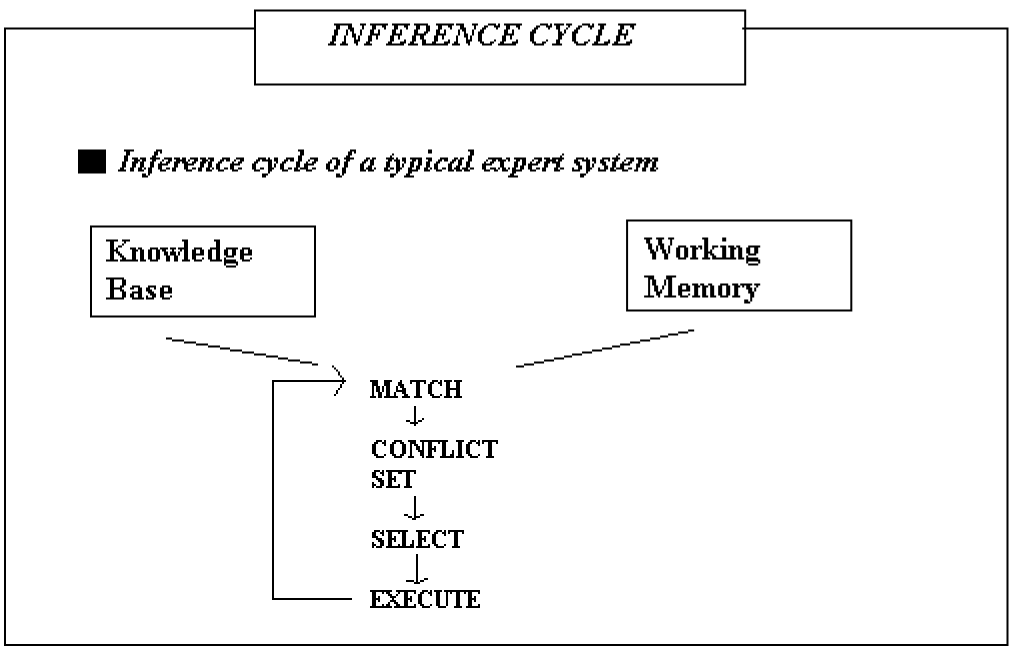
1) <u>Mixed Chaining with priority to backward chaining</u>
only resort to forward chaining when unable to backward chaining
Assume working memory has {A,C} ?goal to determine K.

B B B
R_1,R_3,R_8,R_4,R_7,R_5,R_6,R_2

2) <u>Mixed Chaining with priority to forward chaining</u>
some rule -set and goal and facts

R_8,R_4,R_7,R_5,R_6,R_2(B)

Result: 9 steps v.s +steps



CONTROL

There are two problems addressed by the inference engine:

1) It must have a way to decide where to start.Rules and facts reside in a static knowledge base. There must be a way for the reasining process to begin.

2) The inference engine must resolve conflicts that occur when alternative links of reasining emerge,The system may reach a point where there are more than a few rules ready to fire. The inference engine must choose which rule to examine next.

CONFLICT RESOLUTION STRATEGIES (A PARTIAL LIST)

- **Refractoriness-** Once a given rule fires then that same rule will be disallowed for subsequent firing.(Avoid applying a rule more than once for the same situation)
- **Recency** **-** Rules that apply to the most recently working elements are chosen in preference to thosewhich apply to older working elements.

- most complicated than others
- because it requires that each fact in the fact set is supplemented with a time tag , (or stamp)
- a unique number indicating the "time" the fact was derived.

Ex :

Consider the following fact set

t_1 : x=a,

t_2 : x=b,

t_3 : y=c,

t_4 : z=d

x occurs twice at time t_1 has taken value a and at time t_2 obtained the value b.

- **Specificity** **-** Rules which have more conditions on the left hand side are chosen in preference to those with fewer conditions.

A rulebase can easily be enlarged by adding new, more specific rules to it without worrying too much about older ones, because more specific Production Rules will prevail over more general ones.

For example, a person encountering a friend in the street will not be inclined to think this person is a mammal , but instead think of the person by name , just applying the most specific knowledge.

Give priority to rules with more specific antecedents.

A general rule : If B And D Then P And Q And L.

A specific rule : If A and B And C And D Then J And K And L.