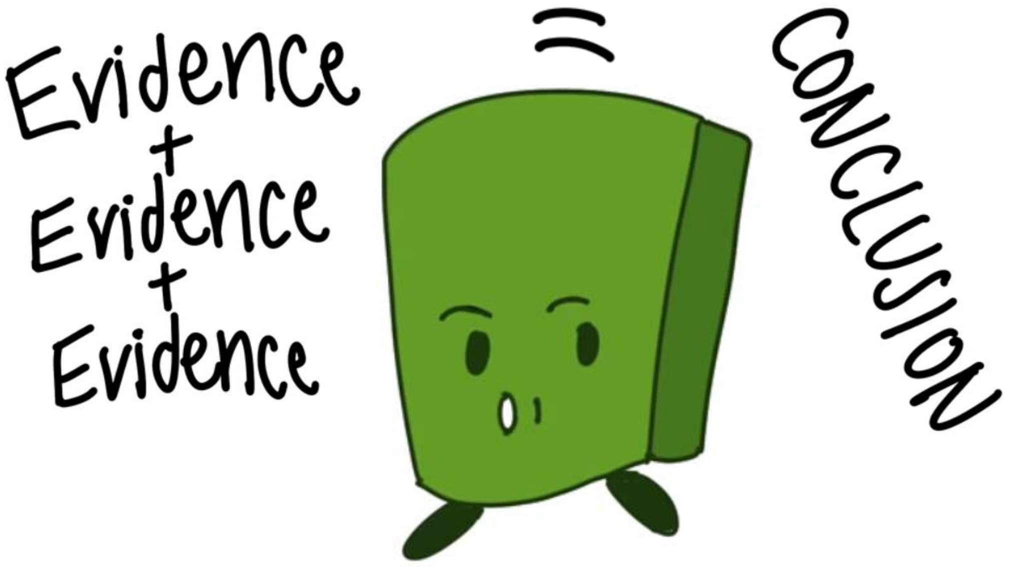
20IS603 Architecture of Intelligent Systems



Inference Chains

# Forward Chaining
# Backward Chaining
# Conflict resolution

Lecture #4

Inference chain

- In Rule-based Expert Systems, the inference engine:
  - Combines the facts with the knowledge contained in the knowledge base to come up with a recommendation.

  - Controls the order in which production rules are applied and resolves conflicts if more than one rule is applicable at a given time.

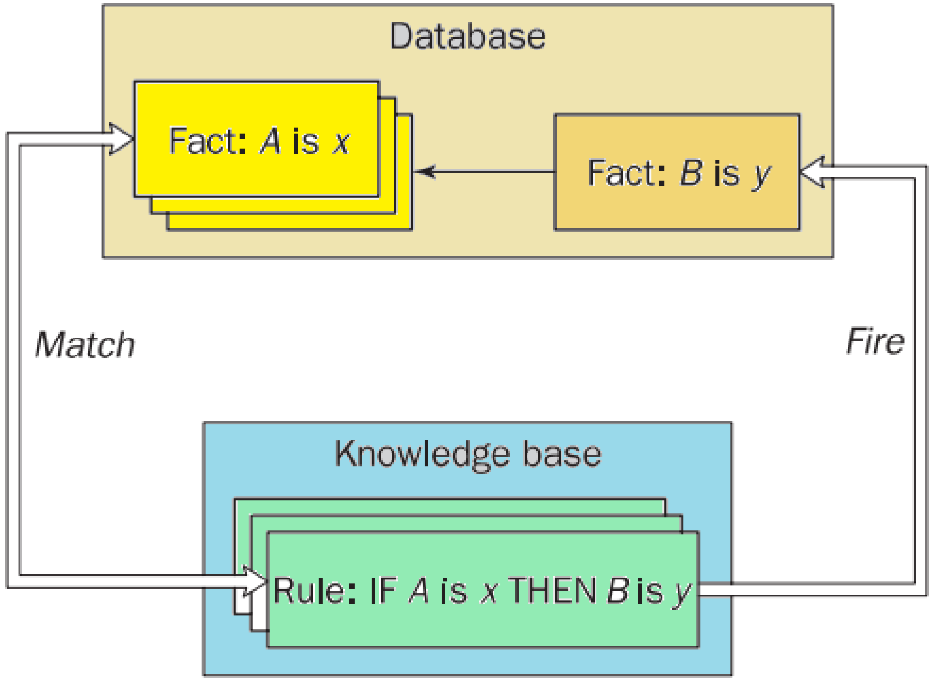
- The process of accessing the knowledge stored in the knowledge base in order to make conclusions – inferencing

- The Inference process consists of two parts
  - Single Inference - Process of applying inference rules to combine two pieces of knowledge to derive a new premise.

  - Multiple Inference - The sequence or order of applying the single inference process to the entire knowledge base in order to derive final conclusions.

Inference chain

- A group of multiple inferences that connects a problem with its solution is called a chain.
- The matching of the rule IF parts to the facts produces inference chains.
- The inference chain indicates how an expert system applies the rules to reach a conclusion.
- When the IF (condition) part of the rule matches a fact, the rule is *fired* and its THEN (action) part is executed.
- When the IF part of the rule does not match a fact, the rule is said to *fail*
- Of the available rules, those whose conditions are satisfied are said to have been triggered and make up the conflict set



Conflict Resolution

- Method of choosing one rule to fire from those set of triggered rules known as the conflict set
- A decision is necessary as to which of the rules takes precedence.
- The conflict between these rules must be resolved.
  - First Come, First Served
  - Priority Values
  - Metarules

Conflict Resolution

- Highest priority rule – each rule has a priority associated with it and if several rules apply, the one with the highest priority is chosen.

- Highest priority conditions – each condition has a priority associated with it. For a rule to be chosen it must contain the highest priority conditions.

- Most recent – the rule whose condition has most recently been met is chosen.

- Most specific – the rule which has most conditions met is selected - also referred to as ‘longest matching’.

- Context limiting – rules are split into groups, only some of which are active at a certain time. To be chosen a rule must belong to an active group

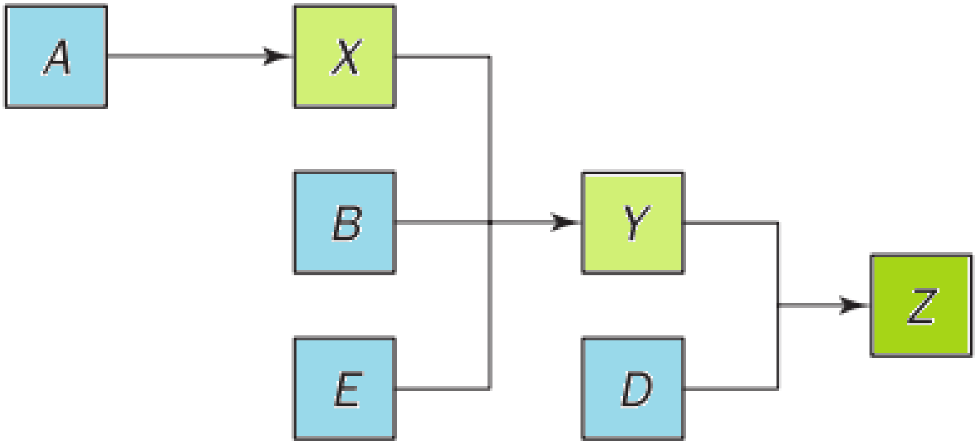
Inference chain - Example

- Example - Suppose the database initially includes facts A, B, C, D and E, and the knowledge base contains only three rules

```
Rule 1: IF Y is true
        AND D is true
        THEN Z is true

Rule 2: IF X is true
        AND B is true
        AND E is true
        THEN Y is true

Rule 3: IF A is true
        THEN X is true
```



Forward Chaining

- Using deduction to reach a conclusion from a set of antecedents.
- The system starts from a set of facts, and a set of rules, and tries to find a way of using those rules and facts to deduce a conclusion or come up with a suitable course of action.
- Data-driven reasoning because the reasoning starts from a set of data and ends up at the goal, which is the conclusion

Forward Chaining

Key points:

- Match the IF part of each rule against facts in database.
- If there is more than one rule that could be used select which one to apply by using conflict resolution
- Apply the rule. If new facts are obtained add them to database.
- Stop (or exit) when the conclusion is added to the database or if there is a rule which specifies to end the process.

Forward Chaining

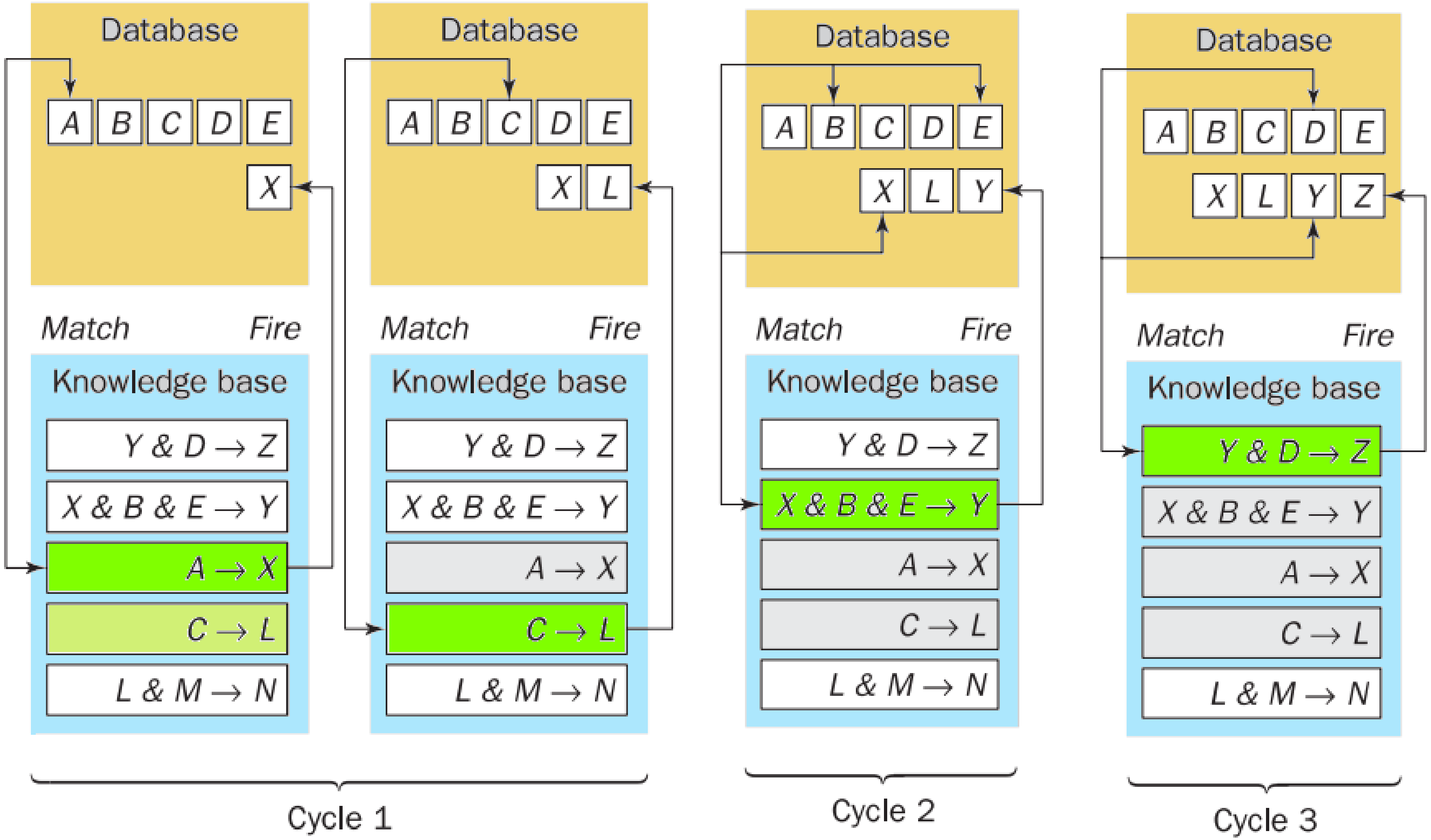
Rule 1: $Y \& D \rightarrow Z$

Rule 2: $X \& B \& E \rightarrow Y$

Rule 3: $A \rightarrow X$

Rule 4: $C \rightarrow L$

Rule 5: $L \& M \rightarrow N$



Forward Chaining – Exercise

- Consider the following expert systems whose database consists of the facts A, B, C, D, E and whose knowledge base is given by the rules below:

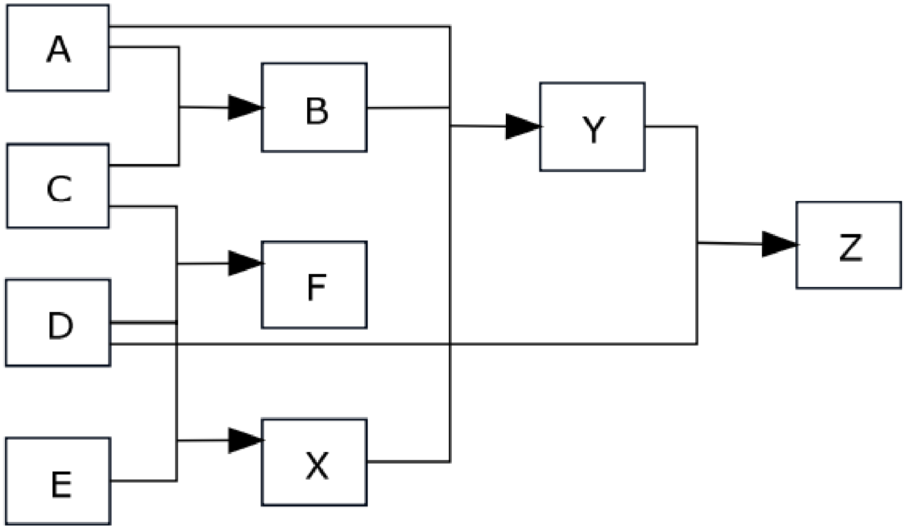
Rule 1: IF A is true
AND C is true
THEN B is true

Rule 2: IF C is true
AND D is true
THEN F is true

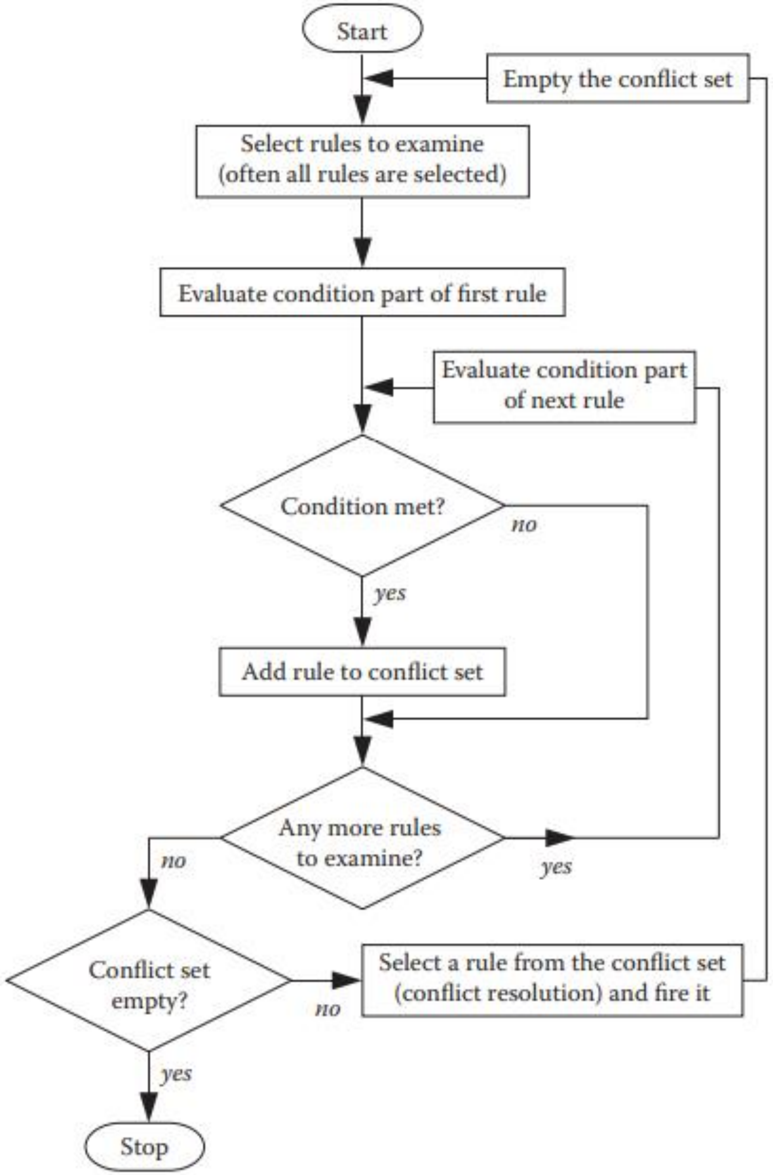
Rule 3: IF C is true
AND D is true
AND E is true
THEN X is true

Rule 4: IF A is true
AND B is true
AND X is true
THEN Y is true

Rule 5: IF D is true
AND Y is true
THEN Z is true



Forward Chaining - Flowchart


Start
Empty the conflict set
Select rules to examine (often all rules are selected)
Evaluate condition part of first rule
Evaluate condition part of next rule
Condition met?
no
yes
Add rule to conflict set
Any more rules to examine?
no
yes
Conflict set empty?
no
Select a rule from the conflict set (conflict resolution) and fire it
yes
Stop

Backward Chaining

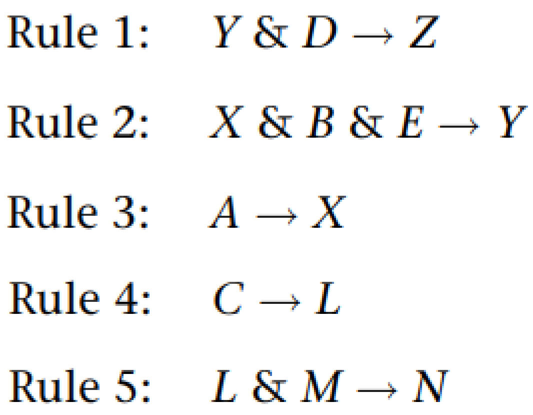
- In some cases, forward chaining can be inefficient because it may end up proving a number of conclusions that are not currently interesting.
- In such cases, where a single specific conclusion is to be proved, backward chaining is more appropriate.
- Starts from a conclusion (hypothesis) and show how that conclusion can be reached from the rules and facts in the database
- The conclusion aiming to prove is called a goal, and so reasoning in this way is known as goal-driven reasoning
- An expert system has the goal (a hypothetical solution) and the inference engine attempts to find the evidence to prove it.

Backward Chaining

Key points:

- First, the knowledge base is searched to find rules that might have the desired solution. Such rules must have the goal in their THEN (action) parts.
- If such a rule is found and its IF (condition) part matches data in the database, then the rule is fired and the goal is proved.
- If no match found then, the inference engine puts aside the rule it is working with (the rule is said to stack) and sets up a new goal, a sub-goal, to prove the IF part of this rule.
- Then the knowledge base is searched again for rules that can prove the sub-goal.
- The inference engine repeats the process of stacking the rules until no rules are found in the knowledge base to prove the current sub-goal.

Backward Chaining - Example



Rule 1: $Y \,\&\, D \rightarrow Z$

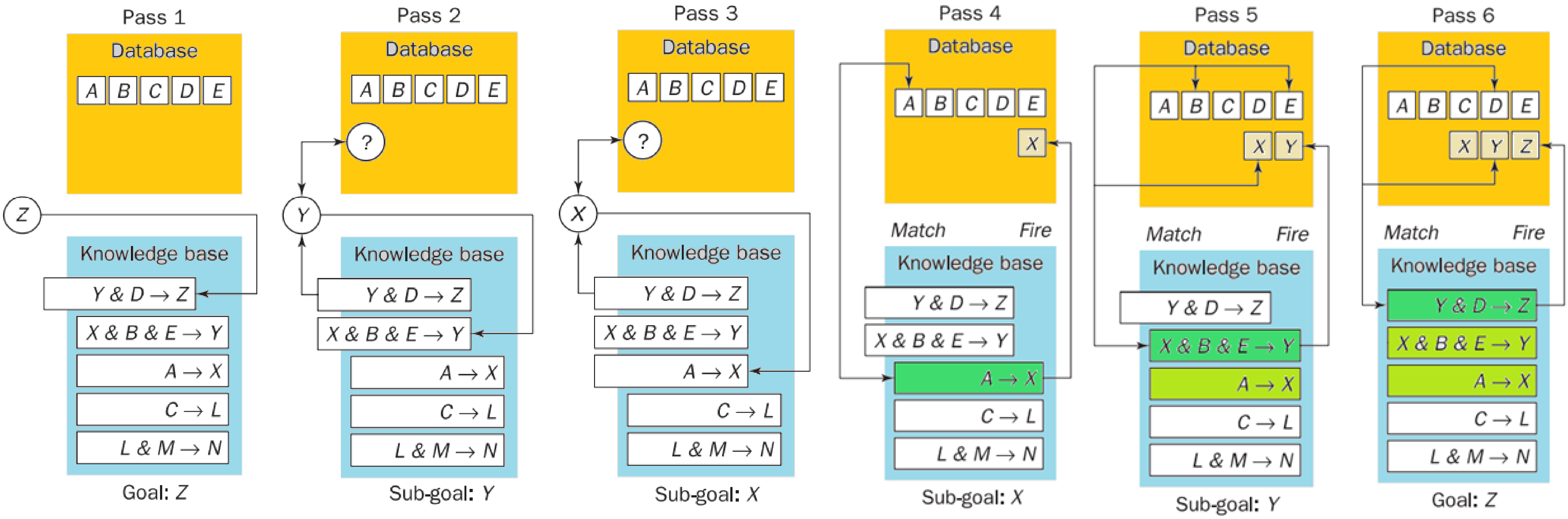
Rule 2: $X \,\&\, B \,\&\, E \rightarrow Y$

Rule 3: $A \rightarrow X$

Rule 4: $C \rightarrow L$

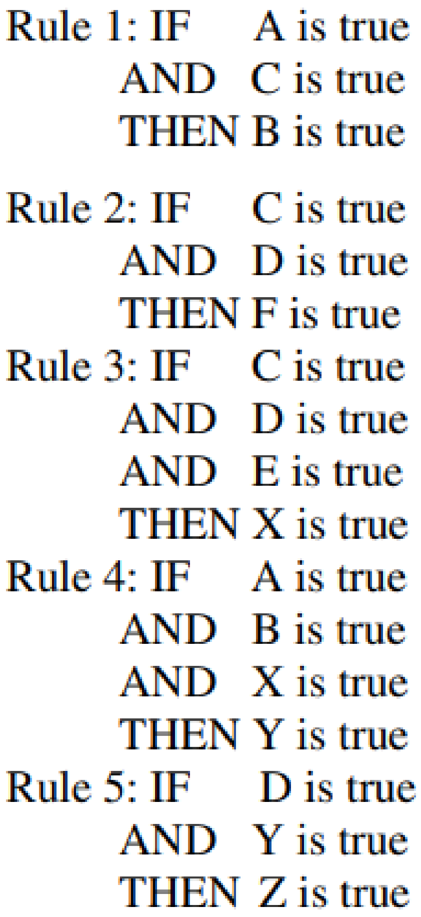
Rule 5: $L \,\&\, M \rightarrow N$

Backward Chaining - Example



Backward Chaining – Exercise

- Consider the following expert systems whose database consists of the facts A, B, C, D, E and whose knowledge base is given by the rules below:



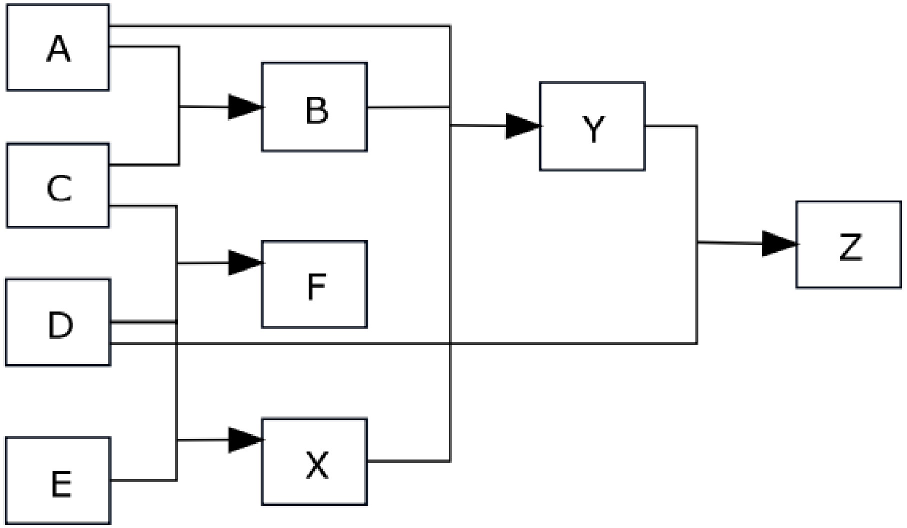
Rule 1: IF A is true
AND C is true
THEN B is true

Rule 2: IF C is true
AND D is true
THEN F is true

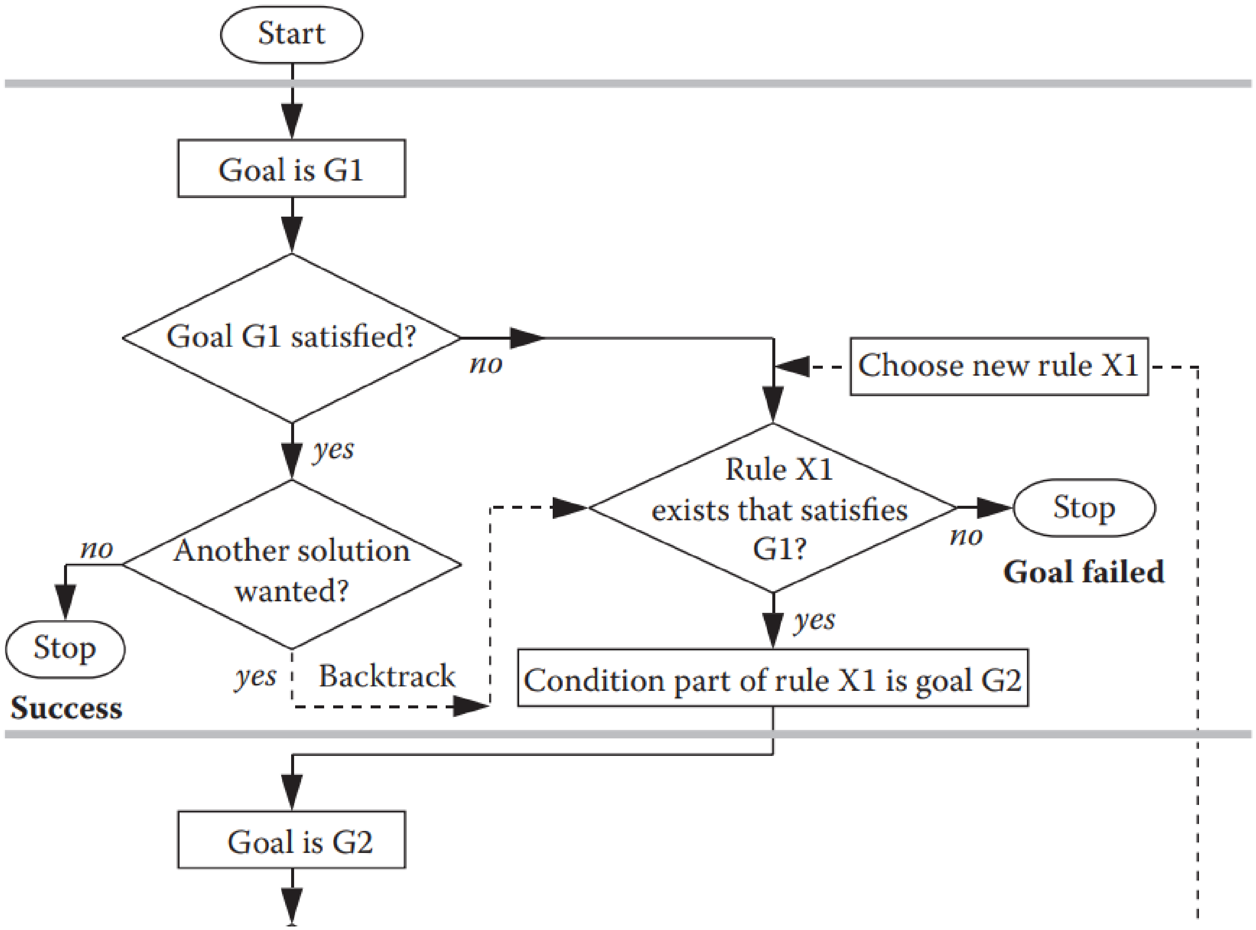
Rule 3: IF C is true
AND D is true
AND E is true
THEN X is true

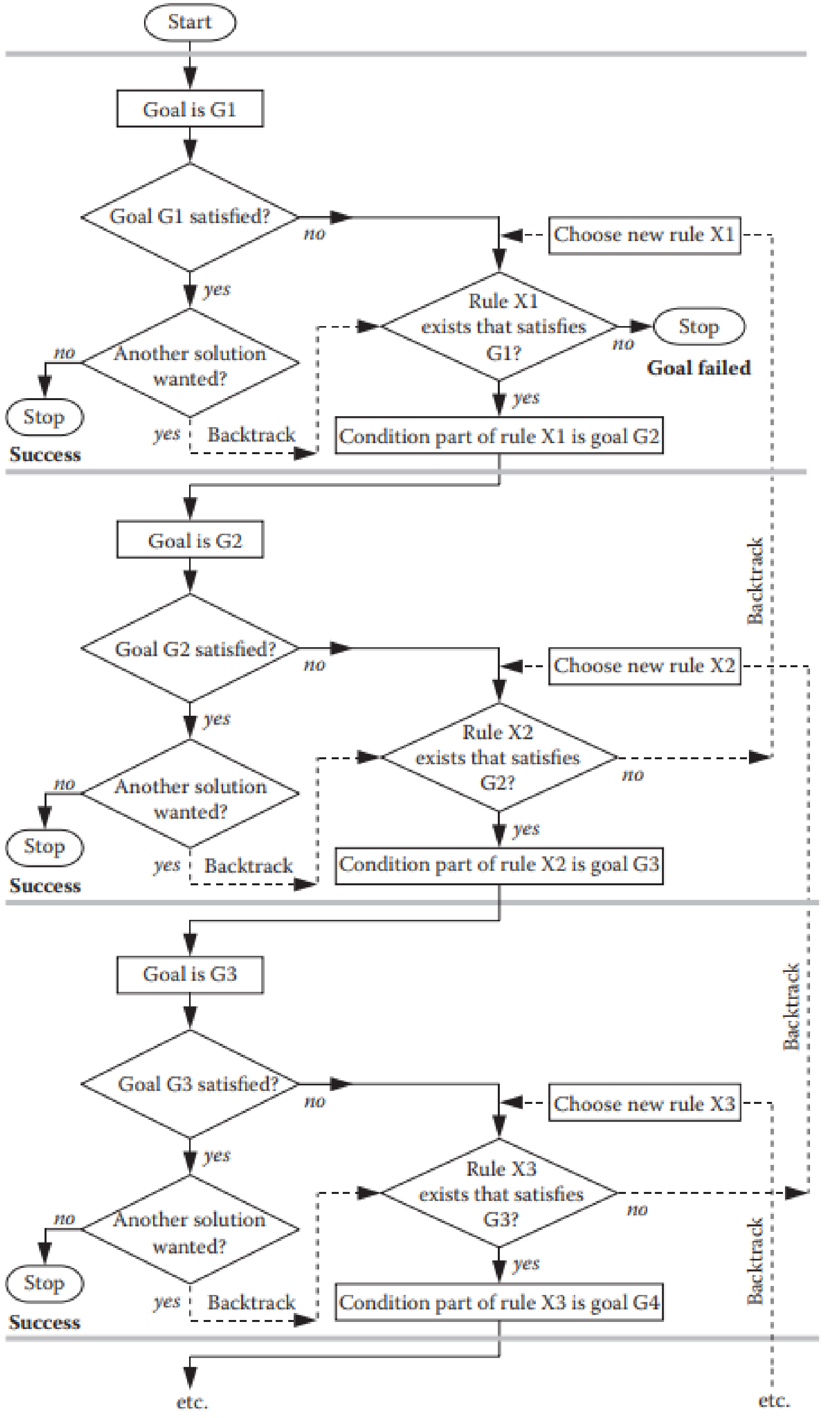
Rule 4: IF A is true
AND B is true
AND X is true
THEN Y is true

Rule 5: IF D is true
AND Y is true
THEN Z is true



Backward Chaining - Flowchart


Start
Goal is G1
Goal G1 satisfied?
no
Choose new rule X1
yes
Rule X1
exists that satisfies
G1?
no
Stop
Goal failed
no
Another solution
wanted?
Stop
Success
yes
Backtrack
yes
Condition part of rule X1 is goal G2
Goal is G2


Start
Goal is G1
Goal G1 satisfied?
no
Choose new rule X1
yes
Rule X1
exists that satisfies
G1?
no
Stop
Goal failed
no
Another solution
wanted?
Stop
Success
yes
Backtrack
yes
Condition part of rule X1 is goal G2
Backtrack
Goal is G2
Goal G2 satisfied?
no
Choose new rule X2
yes
Rule X2
exists that satisfies
G2?
no
no
Another solution
wanted?
Stop
Success
yes
Backtrack
yes
Condition part of rule X2 is goal G3
Backtrack
Goal is G3
Goal G3 satisfied?
no
Choose new rule X3
yes
Rule X3
exists that satisfies
G3?
no
no
Another solution
wanted?
Stop
Success
yes
Backtrack
yes
Condition part of rule X3 is goal G4
Backtrack
etc.
etc.