Inference Examples

Forward Chaining

Rule 1: If on first floor
And button is pressed on first floor
Then open door.

Rule 2: If on first floor
And button is pressed on second floor
Then go to second floor.

Rule 3: If on first floor
And button is pressed on third floor
Then go to third floor.

Rule 4: If on second floor
And button is pressed on first floor
And already going to third floor
Then remember to go to first floor later

Let us imagine that we start with the following facts in our database:

Fact 1
On first floor

Fact 2
Button pressed on third floor

Fact 3
Today is Tuesday

Now the system examines the rules and finds that Facts 1 and 2 match the antecedents of Rule 3. Hence, Rule 3 fires, and its conclusion “Go to third floor” is added to the database of facts.

Presumably, this results in the elevator heading toward the third floor. Note that Fact 3 was ignored altogether because it did not match the antecedents of any of the rules.

Now let us imagine that the elevator is on its way to the third floor and has reached the second floor,when the button is pressed on the first floor. The fact “Button pressed on first floor” Is now added to the database, which results in Rule 4 firing.

Now let us imagine that later in the day the facts database contains the following information:

Fact 1

At first floor

Fact 2

Button pressed on second floor

Fact 3

Button pressed on third floor

In this case, two rules are triggered—Rules 2 and 3. In such cases where there is more than one possible conclusion, **conflict resolution** needs to be applied to decide which rule to fire.

Suppose we have developed the following rules for our weather forecasting system,

Rule I

If we suspect temperature is less than 20^{o}
AND there is humidity in the air
Then there are chances of rain

Rule II

If Sun is behind the clouds
AND air is very cool.
Then we suspect temperature is less than 20^{o}.

Rule III **If** air is very heavy
Then there is humidity in the air.

First Pass

Rule, premise	Status	Working Memory
1, 1 we suspect temperature is less than 20°	Unknown	a) Sun is behind the clouds. b) Air is very heavy and cool.
1, 2 there is humidity in the air	Unknown	a) Sun is behind the clouds. b) Air is very heavy and cool.
2, 1 Sun is behind the clouds	True	a) Sun is behind the clouds. b) Air is very heavy and cool.
2,2 air is very cool.	True, fire rule	a) Sun is behind the clouds. b) Air is very heavy and cool. c) We suspect temperature is less than 20°

Second Pass

Rule, premise	Status	Working Memory
1, 1 we suspect temperature is less than 20°	True	a) Sun is behind the clouds. b) Air is very heavy and cool. c) We suspect temperature is less than 20°
1, 2 there is humidity in the air	Unknown	a) Sun is behind the clouds. b) Air is very heavy and cool. c) We suspect temperature is less than 20°
3, 1 air is very heavy	True, fire rule	a) Sun is behind the clouds. b) Air is very heavy and cool. c) We suspect temperature is less than 20° d) there is humidity in the air

Third Pass

Rule, premise	Status	Working Memory
1, 1 we suspect temperature is less than 20°	True	a) Sun is behind the clouds. b) Air is very heavy and cool. c) We suspect temperature is less than 20° d) there is humidity in the air
1, 2 there is humidity in the air	True, fire rule	a) Sun is behind the clouds. b) Air is very heavy and cool. c) We suspect temperature is less than 20° d) there is humidity in the air **e) there are chances of rain**

So we have deduced there are chances of rain.

CONFLICT RESOLUTION

CONFLICT RESOLUTION

- Example:

  IF you're bored

  AND you've no cash

  THEN go to a friend's place.

  IF you're bored

  AND You've no cash

  THEN go to a park.

CONFLICT RESOLUTION STRATEGIES

- We have different resolution strategies:

1) Fire the first rule in sequence.
2) Assign rule priorities (by importance).
3) More specific rules are preferred over more general rules.(e.g. a rule having 5 IF's(handle more info) will be preferred over one having 3 IF's)
4) Prefer rules whose premises are added more recently (time stamping)
5) Parallel strategy (create view points)

EXAMPLE (Ben Coppin)

For example, consider the following set of rules:

IF it is cold

THEN wear a coat

IF it is cold

THEN stay at home

IF it is cold

THEN turn on the heat

If there is a single fact in the fact database, which is "it is cold," then clearly there are three conclusions that can be derived. In some cases, it might be fine to follow all three conclusions, but in many cases the conclusions are incompatible (for example, when prescribing medicines to patients).

- An alternative method is the **longest-matching strategy.** This method involves firing the conclusion that was derived from the longest rule.
- For example:

  IF patient has pain

  THEN prescribe painkiller

  IF patient has chest pain

  AND patient is over 60

  AND patient has history of heart conditions

  THEN take to emergency room

  Here, if all the antecedents of the second rule match, then this rule's conclusion should be fired rather than the conclusion of the first rule because it is a more specific match.

A further method for conflict resolution is to fire the rule that has matched the facts most recently added to the database. In each case, it may be that the system fires one rule and then stops (as in medical diagnosis), but in many cases, the system simply needs to choose a suitable ordering for the rules (as when controlling an elevator) because each rule that matches the facts needs to be fired at some point.

BACKWARD CHAINING

Suppose we have developed the following rules for our weather forecasting system,

Rule I

If we suspect temperature is less than 20^{o}
AND there is humidity in the air
Then there are chances of rain

Rule II

If Sun is behind the clouds
AND air is very cool.
Then we suspect temperature is less than 20^{o}.

Rule III **If** air is very heavy
Then there is humidity in the air.

- Suppose we have been given the following facts,

a) Sun is behind the clouds.

b) Air is very heavy and cool.

- Problem: Using Backward chaining try to conclude that there are chances of rain.

Step	Description	Working Memory
1	Goal "There are chances of rain." Not in Working Memory.	
2	Find rules with our goal "There are chances of rain" in conclusion: It is in Rule 1.	
3	Now see if Rule 1, premise 1 is known "we suspect temperature is less than 20^0".	
4	This is conclusion of rule 2. So going to Rule 2. The premise 1 of rule 2 is "Sun is behind the clouds".	
5	This is primitive. We ask from user Response: Yes	Sun is behind the clouds.

6	See if Rule 2, premise 2 is known "Air is very cool".	
7	This is also primitive. We ask its Response: Yes. Both conditions of Rule 2 are met so Fire rule 2	Sun is behind the clouds. Air is very cool. **We suspect temperature is less than 20^0.**
8	So Rule 1 premise 1 is in working memory, coming to Rule 1, premise 2 "There is humidity in the air"	Sun is behind the clouds. Air is very cool. **We suspect temperature is less than 20^0.**
9	This is conclusion of Rule 3. So see if Rule 3, premise 1 is known "Air is very heavy".	Sun is behind the clouds. Air is very cool. **We suspect temperature is less than 20^0.**

10	This is primitive so asking from user Response: Yes. Fire rule	Sun is behind the clouds. Air is very cool. We suspect temperature is less than 20^0. **There is humidity in the air.**
11	Now Rule 1 premise 1 and 2 both are in working memory so fire Rule 1.	Sun is behind the clouds. Air is very cool. Air is very heavy. We suspect temperature is less than 20^0. There is humidity in the air. **There are chances of rain.**

COMPARING FORWARD AND BACKWARD CHAINING

BEN COPPIN

Rules:

Rule 1	$A \wedge B \rightarrow C$
Rule 2	$A \rightarrow D$
Rule 3	$C \wedge D \rightarrow E$
Rule 4	$B \wedge E \wedge F \rightarrow G$
Rule 5	$A \wedge E \rightarrow H$
Rule 6	$D \wedge E \wedge H \rightarrow I$

Facts:

Fact 1	A
Fact 2	B
Fact 3	F

Goal:

Our goal is to prove H.

First let us use forward chaining. As our conflict resolution strategy, we will fire rules in the order they appear in the database, starting from Rule 1.

In the initial state, Rules 1 and 2 are both triggered. We will start by firing Rule 1, which means we add C to our fact database. Next, Rule 2 is fired, meaning we add D to our fact database.

We now have the facts A, B, C, D, F, but we have not yet reached our goal, which is G.

Now Rule 3 is triggered and fired, meaning that fact E is added to the database. As a result, Rules 4 and 5 are triggered. Rule 4 is fired first, resulting in Fact G being added to the database, and then Rule 5 is fired, and Fact H is added to the database. We have now proved our goal and do not need to go on any further.

This deduction is presented in the following table:

Facts	Rules triggered	Rule fired
A, B, F	1, 2	1
A, B, C, F	2	2
A, B, C, D, F	3	3
A, B, C, D, E, F	4, 5	4
A, B, C, D, E, F, G	5	5
A, B, C, D, E, F, G, H	6	STOP

Now we will consider the same problem using backward chaining. To do so, we will use a goals database in addition to the rule and fact databases. In this case, the goals database starts with just the conclusion, H, which we want to prove. We will now see which rules would need to fire to lead to this conclusion. Rule 5 is the only one that has H as a conclusion, so to prove H, we must prove the antecedents of Rule 5, which are A and E.

Fact A is already in the database, so we only need to prove the other antecedent, E. Therefore, E is added to the goal database. Once we have proved E, we now know that this is sufficient to prove H, so we can remove H from the goals database.

So now we attempt to prove Fact E. Rule 3 has E as its conclusion, so to prove E, we must prove the antecedents of Rule 3, which are C and D. Neither of these facts is in the fact database, so we need to prove both of them. They are both therefore added to the goals database. D is the conclusion of Rule 2 and Rule 2's antecedent, A, is already in the fact database, so we can conclude D and add it to the fact database.

Similarly, C is the conclusion of Rule 1, and Rule 1's antecedents, A and B, are both in the fact database. So, we have now proved all the goals in the goal database and have therefore proved H and can stop.

This process is represented in the table below:

Facts	Goals	Matching rules
A, B, F	H	5
A, B, F	E	3
A, B, F	C, D	1
A, B, C, F	D	2
A, B, C, D, F		STOP

In this case, backward chaining needed to use one fewer rule. If the rule database had had a large number of other rules that had A, B, and F as their antecedents, then forward chaining might well have been even more inefficient.

- Now let’s solve the same problem using resolution
  - First, convert to CNF
  - Negate the thing we are trying to prove and add it to the list
  - Resolve clauses to see if we can make the knowledge base unsatisfiable

- Either Heather attended the meeting or Heather was not invited. If the boss wanted Heather at the meeting, then she was invited. Heather did not attend the meeting. If the boss did not want Heather there, and the boss did not invite her there, then she is going to be fired. Use resolution to prove that Heather is going to be fired.

- If it rains, Joe brings his umbrella.
- If Joe has an umbrella, he doesn’t get wet.
- If it doesn’t rain, Joe doesn’t get wet.

Prove by resolution that:

Joe doesn’t get wet.