

Chapter 2. Knowledge and Reasoning [Knowledge-based / Logical Agent]

Topic 2.1 Knowledge representation and reasoning with **Propositional Logic (PL)**

a) PL

- PL is a language for representing Knowledge Bases (KBs)
- It is also used for reasoning/ inference / answering queries to KBs.
- It is one of the simplest languages for the purposes.

b) Inference in PL

- To infer means to derive a new sentence from some old ones.
- Forward and backward chaining algorithms are used for easy inference.

c) Assumptions for working with PL

- KB contains only Horn clauses (definite clauses)
- Clause: disjunction of literals
- Horn clause: clause with at most one positive literal

d) Example of executing the forward chaining algorithm

KB:

A

B

C

D

$P \Rightarrow Q$

$C \wedge L \Rightarrow P$

$D \wedge M \Rightarrow P$

$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

$A \wedge D \Rightarrow G$

$G \wedge B \Rightarrow C$

Query: $KB \models M ?$

Execution:

A
B
C
D
P
C
D
B
A
A
A
G

Execution:

$$\text{ii) } \frac{\begin{array}{c} A \\ B \end{array}}{A \wedge B \Rightarrow L} \vdash L, [\text{Added to KB}]$$
$$\text{iii) } \begin{array}{l} A \\ D \\ A \wedge D \Rightarrow G \end{array} \models G, [\text{Added to KB}]$$
$$\text{iv) } \begin{array}{l} \text{B} \\ \text{L} \\ \text{B} \wedge \text{L} \Rightarrow \text{M} \end{array} \models \text{M, [Added to KB]}$$

v) M is found in KB.

A
B
C
D
P
C
D
B
A
A
A
G

Execution:

i) M is not found in KB.

$$\text{ii) } \frac{\begin{array}{c} A \\ B \end{array}}{A \wedge B \Rightarrow L} \models L, [\text{Added to KB}]$$

.....

- ❖ It is called a data driven algorithm
- ❖ Query → data search → premise → conclusion → conclusion added to KB → data search ... until found or KB exhausted

f) Example of executing the backward chaining algorithm

KB (same):

A	$A \wedge P \Rightarrow L$
B	$A \wedge B \Rightarrow L$
C	$A \wedge D \Rightarrow G$
D	$G \wedge B \Rightarrow C$
$P \Rightarrow Q$	
$C \wedge L \Rightarrow P$	
$D \wedge M \Rightarrow P$	
$B \wedge L \Rightarrow M$	

Query: $KB \models M ?$

□ Characteristics of a typical **backward chaining** algorithm

- ❖ A backward chaining algorithm is called a **goal oriented** algorithm.
- ❖ Query \rightarrow goal \rightarrow conclusion \rightarrow premise \rightarrow subgoal \rightarrow conclusion \rightarrow premise \rightarrow subgoal \rightarrow backtracking \rightarrow ... until proved or KB exhausted

Example:

KB:

1. A
2. B
3. C
4. D
5. $P \Rightarrow Q$
6. $C \wedge L \Rightarrow P$
7. $D \wedge M \Rightarrow P$
8. $B \wedge L \Rightarrow M$
9. $A \wedge P \Rightarrow L$
10. $A \wedge B \Rightarrow L$
11. $A \wedge D \Rightarrow G$
12. $G \wedge B \Rightarrow C$

Query: $KB \models M ?$

1. ? M
2. $B \wedge L \Rightarrow M$, Found in KB
3. ? $B \wedge L$
4. ? B ? L
5. B, Found in KB
6. ? L
7. $A \wedge P \Rightarrow L$, Found in KB; **Alternatives are there**
8. ? $A \wedge P$
9. ? A ? P
10. A, Found in KB
11. ? P
12. $C \wedge L \Rightarrow P$, Found in KB; **Alternatives are there**
13. ? $C \wedge L$
14. ? C ? L
15. C, Found in KB
16. ? L **[Backtrack needed; See step 6]**

Example:

KB:

1. A
2. B
3. C
4. D
5. $P \Rightarrow Q$
6. $C \wedge L \Rightarrow P$
7. $D \wedge M \Rightarrow P$
8. $B \wedge L \Rightarrow M$
9. $A \wedge P \Rightarrow L$
10. $A \wedge B \Rightarrow L$
11. $A \wedge D \Rightarrow G$
12. $G \wedge B \Rightarrow C$

Query: $KB \models M ?$

12₁. $D \wedge M \Rightarrow P$, Found in KB

13₁. ? $D \wedge M$

14₁. ? D ? M

15₁. D, Found in KB

16₁. ? M **[Backtrack needed; See step 1]**

7₁. $A \wedge B \Rightarrow L$, Found in KB

8₁. ? $A \wedge B$

9₁. ?A ?B

10₁. A, Found in KB

11₁. B, Found in KB

[Proved]