Forward Chaining Inference Algorithm

1 Query and Knowledge Base

Given the query Q, we aim to determine if the knowledge base KB entails the query Q, denoted by $KB \models Q$.

1.1 Knowledge Base (KB)

The knowledge base consists of the following clauses:

1. $P \Rightarrow Q$
2. $L \wedge M \Rightarrow P$
3. $B \wedge L \Rightarrow M$
4. $A \wedge P \Rightarrow L$
5. $A \wedge B \Rightarrow L$
6. A (fact)
7. B (fact)

1.2 Count Table

We create a count table that tracks the number of symbols in each clause's premise that need to be inferred before the conclusion can be inferred:

Clause	Premise	Count
1.	$P \Rightarrow Q$	1 (symbol: P)
2.	$L \wedge M \Rightarrow P$	2 (symbols: L, M)
3.	$B \wedge L \Rightarrow M$	2 (symbols: B, L)
4.	$A \wedge P \Rightarrow L$	2 (symbols: A, P)
5.	$A \wedge B \Rightarrow L$	2 (symbols: A, B)

2 Inferred Table

Initialize an inferred table where each symbol starts as false, except for the given facts:

$$\begin{aligned}
\text{Inferred}[Q] &= \text{false} \\
\text{Inferred}[P] &= \text{false} \\
\text{Inferred}[L] &= \text{false} \\
\text{Inferred}[M] &= \text{false} \\
\text{Inferred}[A] &= \text{true} \quad \text{(fact)} \\
\text{Inferred}[B] &= \text{true} \quad \text{(fact)}
\end{aligned}$$

3 Queue Initialization

Initialize the queue with symbols that are known to be true in the knowledge base:

$$\text{queue} = [A, B]$$

4 Main Loop Execution

4.1 Iteration 1

Pop from the queue: $p = A$
Check if $p = Q$: No, so continue.
Since Inferred[A] = true, continue with the next symbol in the queue.
For each clause where A is a premise:

- Clause 4: $A \wedge P \Rightarrow L$
  Count[4] remains 1 as P is not yet inferred.
- Clause 5: $A \wedge B \Rightarrow L$
  Decrement Count[5] $\rightarrow$ 1.

4.2 Iteration 2

Pop from the queue: $p = B$
Check if $p = Q$: No, so continue.
Since Inferred[B] = true, continue with the next symbol in the queue.
For each clause where B is a premise:

- Clause 3: $B \wedge L \Rightarrow M$
  Count[3] remains 1 as L is not yet inferred.
- Clause 5: $A \wedge B \Rightarrow L$
  Decrement Count[5] $\rightarrow$ 0.

Since Count[5] = 0, add conclusion L to the queue.

4.3 Iteration 3

Pop from the queue: $p = L$
Check if $p = Q$: No, so continue.
Update Inferred$[L]$ = true.
For each clause where L is a premise:

- Clause 2: $L \wedge M \Rightarrow P$
  Decrement Count[2] $\rightarrow 1$.

- Clause 3: $B \wedge L \Rightarrow M$
  Decrement Count[3] $\rightarrow 0$.

Since Count[3] $= 0$, add conclusion M to the queue.

4.4 Iteration 4

Pop from the queue: $p = M$
Check if $p = Q$: No, so continue.
Update Inferred$[M]$ = true.
For each clause where M is a premise:

- Clause 2: $L \wedge M \Rightarrow P$
  Decrement Count[2] $\rightarrow 0$.

Since Count[2] $= 0$, add conclusion P to the queue.

4.5 Iteration 5

Pop from the queue: $p = P$
Check if $p = Q$: No, so continue.
Update Inferred$[P]$ = true.
For each clause where P is a premise:

- Clause 1: $P \Rightarrow Q$
  Decrement Count[1] $\rightarrow 0$.

Since Count[1] $= 0$, add conclusion Q to the queue.

4.6 Iteration 6

Pop from the queue: $p = Q$
Check if $p = Q$: Yes, so return true.
Hence, $KB \models Q$.