Intro to AI,
Autumn, 2025

PROPOSITIONAL LOGIC *(Cont)*

Faculty of DS & AI
Autumn semester, 2025

Trong-Nghia Nguyen

Business AI Lab

2025-10

Content

- Representing problems using logic
- Propositional logic
- **Deduction in Propositional Logic - Prove using:**
    - ~~Truth table~~
    - ~~Inference rule~~
    - Robinson’s Method (Resolution Method)
    - Forward Chaining and backward Chaining

Propositional Logic (Cont)

Horn form

- An expression is in the Horn normal form if:
  - It is a conjunction (i.e., an AND combination) of clauses
  - Each clause is a disjunction (i.e., an OR combination) of literals and has at most 1 positive literal
  - Example: $(p \vee \neg q) \wedge (\neg p \vee \neg r \vee s)$
- Not all propositional expression can be converted to the Horn normal form!
- Representation of the set of premises KB in Horn normal form
  - **Rules**
    - $(\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n \vee q)$
    - Equivalent to the rule: $(p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow q)$
  - **Facts**
    - p, q
  - **Integrity constraints**
    - $(\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n)$
    - Equivalent to the rule: $(p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow \mathit{false})$

Propositional Logic (Cont)

Generalized Modus Ponens rule

$$\frac{(p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow q),\ p_1,\ p_2,\ \ldots,\ p_n}{q}$$

- The Modus Ponens rule is *sound* and *complete*, provided that propositional symbols and the set of premises KB are in Horn normal form
- The Modus Ponens rule can be used by both of the 2 reasoning approaches: *Forward reasoning* and *Backward reasoning*

Propositional Logic (Cont)

Literal, clause

- **Literal** is the smallest unit in propositional logic
  - **Positive literal:** An atomic proposition without negation (e.g., P, Q, e, f)
  - **Negative literal:** An atomic proposition with negation (e.g., ¬P, ¬Q, ¬e, ¬f)

- A **clause** is a disjunction (OR, ∨) of literals. In CNF, formulas are expressed as conjunctions (AND, ∧) of clauses

Unit Clauses

C1: e → "e is true" (fact)

C2: ¬w → "w is false" (negated goal)

Binary Clauses

C3: ¬e ∨ b → "If e then b" (e ⇒ b)

C4: ¬v ∨ w → "If v then w" (v ⇒ w)

Example 3: General Clauses

C5: ¬d ∨ ¬m ∨ ¬b ∨ w → "If d and m and b, then w"

C6: ¬u ∨ v ∨ ¬f → "If u then (v or not f)"

Empty Clause (⊥)

- Contains **no literals**
- Represents a **contradiction**
- Goal of refutation proof

Propositional Logic (Cont)

Resolution

- **Resolution** is an inference rule that combines two clauses containing complementary literals to produce a new clause (resolvent)

Resolution Rule (General Form):

Clause 1: $A_1 \lor \ldots \lor A_n \lor P$

Clause 2: $B_1 \lor \ldots \lor B_m \lor \neg P$

Resolvent: $A_1 \lor \ldots \lor A_n \lor B_1 \lor \ldots \lor B_m$

where P and ¬P are complementary literals

Deduction in propositional logic

Definition

- Deduction = deriving **conclusions** logically from known **premises**

Method	Approach	Characteristics
Truth Table	Exhaustive	Simple but exponential complexity
Inference Rules	Symbolic reasoning	More efficient and structured
Resolution (Robinson)	Contradiction-based	Basis for theorem proving

Deduction in propositional logic

Prove using Robinson method

- Proof by **contradiction** Method
- Assume the conclusion is false ($\neg c$), then ($p_1 \wedge p_2 \wedge \ldots \wedge p_n \wedge \neg c$) leads to a **contradiction**

Step 0: Add $\neg c$ into the set of premises.

$p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow c$

Deduction in propositional logic

Prove using Robinson method

Step 1: Normalize into Conjunctive Normal Form (CNF)

Use the following 7 transformations to normalize:

1. $a \rightarrow b \equiv (\neg a \vee b)$ is true => $-(-a \vee b) \equiv (a \wedge -b)$ is false
2. $(a \wedge b) \rightarrow c \equiv \neg a \vee \neg b \vee c$
3. $a \rightarrow (b \wedge c) \equiv (\neg a \vee b) \wedge (\neg a \vee c)$
4. $(a \vee b) \rightarrow c \equiv (\neg a \vee c) \wedge (\neg b \vee c)$
5. $a \rightarrow (b \vee c) \equiv \neg a \vee b \vee c$
6. $a \rightarrow (b \rightarrow c) \equiv \neg a \vee \neg b \vee c$
7. $(a \rightarrow b) \rightarrow c \equiv (a \vee c) \wedge (\neg b \vee c)$

Deduction in propositional logic

Prove using Robinson method

- **Step 2:** Split the normalized formulas into sub-lines (each sub-line is an expression that evaluates to true)
  $(p1 \vee \ldots \vee pn) \wedge \ldots \wedge (q1 \vee \ldots \vee qm)$ splits into:
  $(p1 \vee \ldots \vee pn)$
  …
  $(q1 \vee \ldots \vee qm)$
- **Remarks:**
  - Steps 1 and 2 are preprocessing.
  - Each problem is transformed into a finite set of normalized lines. Each line has only the union $\vee$ of propositions, each of which can be affirmative (p) or negative (¬p)

Deduction in propositional logic

Prove using Robinson method

Step 3: The solution process.

This is an iterative process, each iteration includes the following steps:

- 3.1. Select any two existing clauses that contain the following forms:
  - ~~p~~ v q (clause i)
  - ~~¬p~~ v r (clause j)
  - Selection mechanism: FIFO, LIFO, flexible
- 3.2. Combine those two clauses to form clause k
  - q v r
- Repeat until we have:
  - clause α: proposition u
  - clause β: proposition ¬u

From this, a contradiction is derived

Deduction in propositional logic

Prove using Robinson method

- Summary of the Robinson's proof method:
  $p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow c \equiv p_1 \wedge p_2 \wedge \ldots \wedge p_n \wedge \neg c \rightarrow \{u, \neg u\}$ (contradiction)
- **Robinson procedure**

Begin
1. Add ¬c to the set of premises G
2. Convert the expressions in G into Conjunctive Normal Form (CNF)
3. Split each conjunctive clause in **G** into its disjunctive sub-clauses
4. Repeat
   - 4.1. Choose 2 clauses A, B belonging to G
   - 4.2. If A and B are resolvable then
   - 4.2.1. Compute Res(A,B) (the resolvent of A and B)
   - 4.2.2. If Res(A,B) generates a new clause, add Res(A,B) to G

Until an empty clause is derived or no new clause can be generated
End

Deduction in propositional logic

Prove using Robinson method

Note when select clause: Unit Preference

- Always prioritize choosing **unit clause** (clause with 1 literal)
- Resolution with unit clause creates shorter clause
- Shorter clause → easier to find contradictions

Deduction in propositional logic

Exercise 1

Given that the following premises are true:

1. $v \Rightarrow w$
2. $(d \land m \land b) \Rightarrow w$
3. $u \Rightarrow (v \lor \neg f)$
4. $(b \land c \land a) \Rightarrow v$
5. $(u \land e) \Rightarrow (\neg m \lor a)$
6. $e \Rightarrow (f \land m)$
7. $(e \land f) \Rightarrow u$
8. $((c \lor e) \Rightarrow b) \land e$
9. $b \Rightarrow (d \Rightarrow a)$

Prove that w and b are also true by Robinson's solution method (arbitrary flexible selection mechanism)

Deduction in propositional logic

Exercise 1 Prove that **w** and **b** are true from the given **9 premises**.

A. Part 1: Proving b = TRUE

- **Step 0: Add ¬b to the premise set**
  - Assume b is false, add ¬b to the premise set G.

- **Step 1: Normalize to Conjunctive Normal Form (CNF)**

Original premises:

1. $v \Rightarrow w$
2. $(d \land m \land b) \Rightarrow w$
3. $u \Rightarrow (v \lor \neg f)$
4. $(b \land c \land a) \Rightarrow v$
5. $(u \land e) \Rightarrow (\neg m \lor a)$
6. $e \Rightarrow (f \land m)$
7. $(e \land f) \Rightarrow u$
8. $((c \lor e) \Rightarrow b) \land e$
9. $b \Rightarrow (d \Rightarrow a)$

CNF forms:

1. $\neg v \lor w$
2. $\neg d \lor \neg m \lor \neg b \lor w$
3. $\neg u \lor v \lor \neg f$
4. $\neg b \lor \neg c \lor \neg a \lor v$
5. $\neg u \lor \neg e \lor \neg m \lor a$
6. $(\neg e \lor f) \land (\neg e \lor m)$
7. $\neg e \lor \neg f \lor u$
8. $(\neg c \lor b) \land (\neg e \lor b) \land e$
9. $\neg b \lor \neg d \lor a$

Deduction in propositional logic

Exercise 1 Prove that **w** and **b** are true from the given **9 premises**.

CNF forms:

1. ¬v ∨ w
2. ¬d ∨ ¬m ∨ ¬b ∨ w
3. ¬u ∨ v ∨ ¬f
4. ¬b ∨ ¬c ∨ ¬a ∨ v
5. ¬u ∨ ¬e ∨ ¬m ∨ a
6. (¬e ∨ f) ∧ (¬e ∨ m)
7. ¬e ∨ ¬f ∨ u
8. (¬c ∨ b) ∧ (¬e ∨ b) ∧ e
9. ¬b ∨ ¬d ∨ a

Step 2: Split into sub-clauses
Clause set G:

- C1: ¬v ∨ w
- C2: ¬d ∨ ¬m ∨ ¬b ∨ w
- C3: ¬u ∨ v ∨ ¬f
- C4: ¬b ∨ ¬c ∨ ¬a ∨ v
- C5: ¬u ∨ ¬e ∨ ¬m ∨ a
- C6a: ¬e ∨ f
- C6b: ¬e ∨ m
- C7: ¬e ∨ ¬f ∨ u
- C8a: ¬c ∨ b
- C8b: ¬e ∨ b
- C8c: e (fact from premise 8)
- C9: ¬b ∨ ¬d ∨ a
- C10: ¬b (negated conclusion - proof by contradiction)

Deduction in propositional logic

Exercise 1 Prove that **w** and **b** are true from the given **9 premises**.

Step 2: Split into sub-clauses
Clause set G:
- C1: ¬v ∨ w
- C2: ¬d ∨ ¬m ∨ ¬b ∨ w
- C3: ¬u ∨ v ∨ ¬f
- C4: ¬b ∨ ¬c ∨ ¬a ∨ v
- C5: ¬u ∨ ¬e ∨ ¬m ∨ a
- C6a: ¬e ∨ f
- C6b: ¬e ∨ m
- C7: ¬e ∨ ¬f ∨ u
- C8a: ¬c ∨ b
- C8b: ¬e ∨ b
- C8c: e (fact from premise 8)
- C9: ¬b ∨ ¬d ∨ a
- C10: ¬b (negated conclusion - proof by contradiction)

Step 3: Resolution process
- Iteration 1: Select clauses C8c and C8b (flexible selection mechanism)
  - C8c: e
  - C8b: ¬e ∨ b
  - Res(C8c, C8b) = C11: b ✓ (new clause generated)
- Iteration 2: Select clauses C11 and C10
  - C11: b
  - C10: ¬b
  - Res(C11, C10) = ⊥(empty clause - CONTRADICTION!)

Conduct similar steps part 2: w

Deduction in propositional logic

Exercise 2

1. Tet is coming
2. There is a bonus
3. Tet is coming and there is a bonus → fun
4. Fun → play
5. Play → visit beautiful and new places
6. Visit beautiful and new places → visit My Dinh
7. Visit My Dinh ∧ My Dinh near my Maternal Grandmother'house → visit Maternal Grandmother
8. Visit Maternal Grandmother → visit Paternal Grandmother
9. My Dinh near my Maternal Grandmother'house

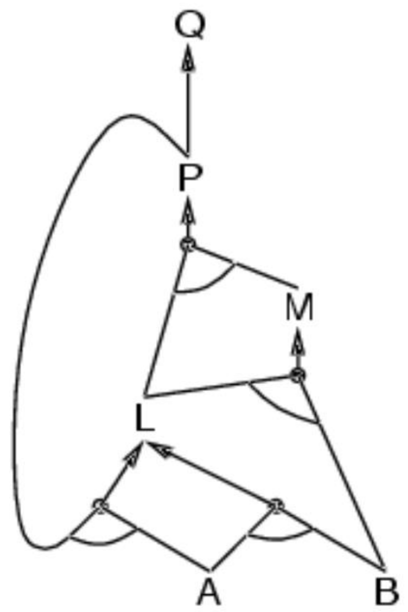
Proof of visit to My Dinh ∧ Paternal Grandmother∧ Maternal Grandmother

Deduction in propositional logic

Forward reasoning (chaining)

- Given a set of premises (knowledge base) *KB*, it requires to prove the expression *Q*
- Idea: Repeat the following 2 steps until inferring the expression
  - Apply a rule whose condition (IF) part is satisfied in *KB*
  - Add the applied rule's conclusion (THEN) part to *KB*

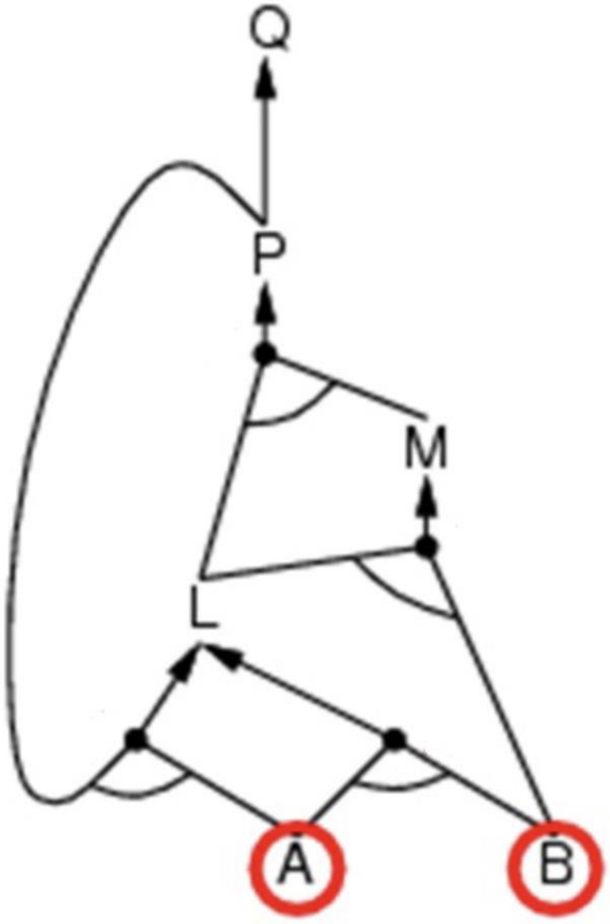
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Deduction in propositional logic

Forward reasoning (chaining)

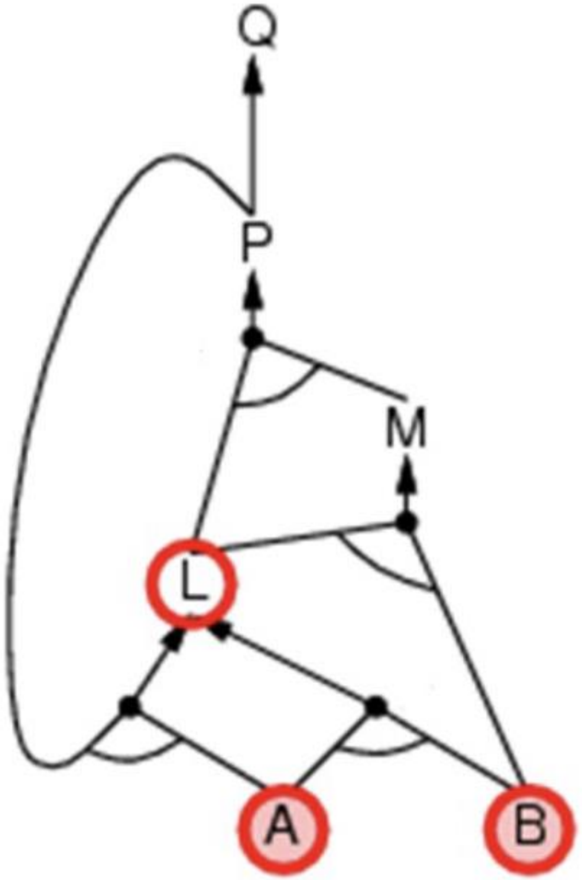
$$P \Rightarrow Q$$
$$L \land M \Rightarrow P$$
$$B \land L \Rightarrow M$$
$$A \land P \Rightarrow L$$
$$A \land B \Rightarrow L$$
$$A$$
$$B$$



Deduction in propositional logic

Forward reasoning (chaining)

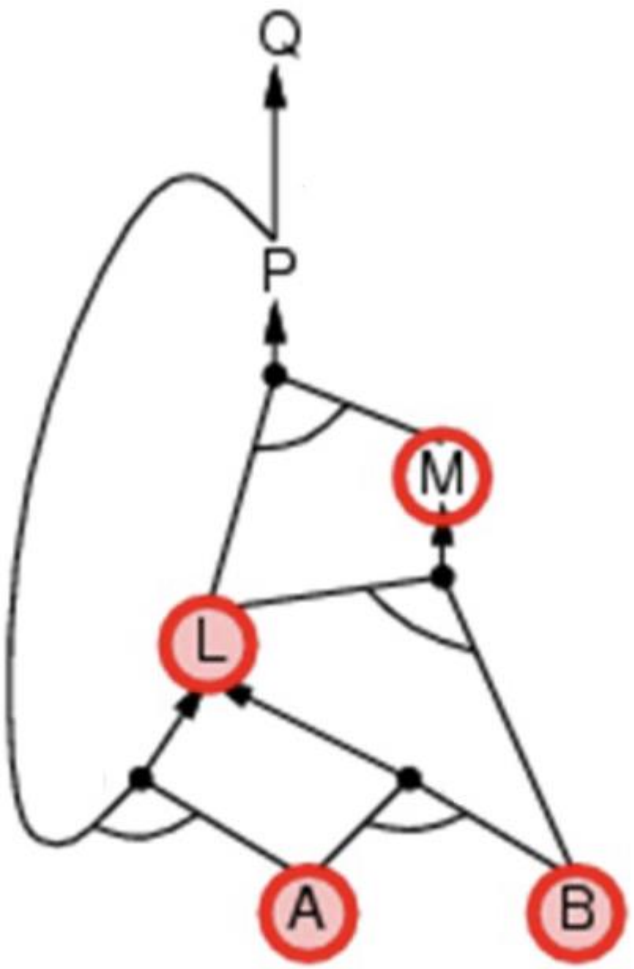
$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B



Deduction in propositional logic

Forward reasoning (chaining)

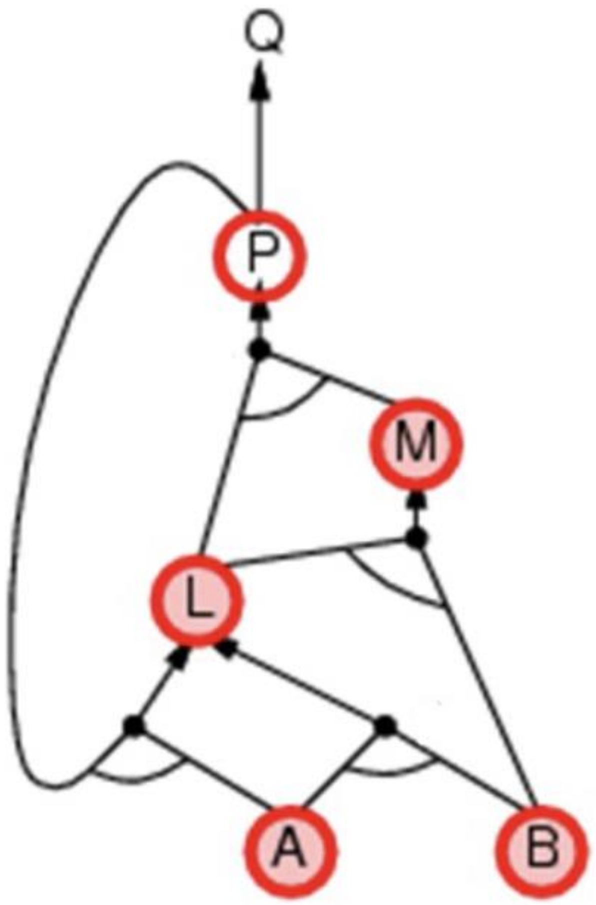
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Deduction in propositional logic

Forward reasoning (chaining)

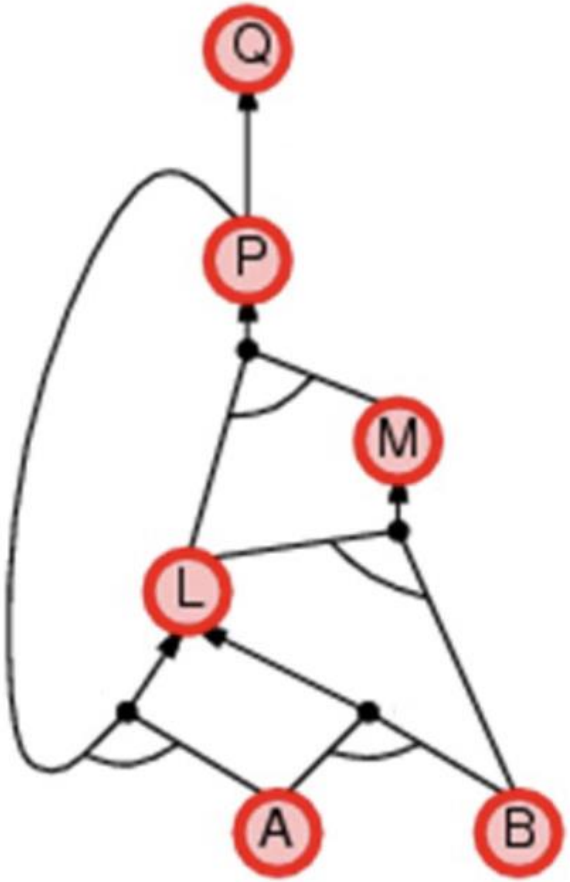
$$
\begin{aligned}
&P \Rightarrow Q \\
&L \wedge M \Rightarrow P \\
&B \wedge L \Rightarrow M \\
&A \wedge P \Rightarrow L \\
&A \wedge B \Rightarrow L \\
&A \\
&B
\end{aligned}
$$



Deduction in propositional logic

Forward reasoning (chaining)

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Deduction in propositional logic

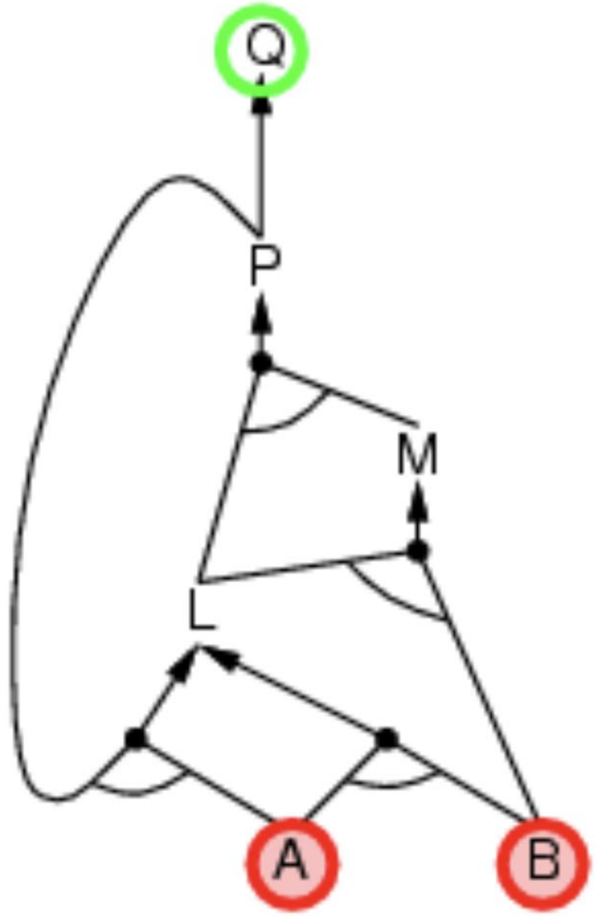
Backward reasoning (chaining)

- Idea: The reasoning process starts from the conclusion *Q*
- To prove *Q* by the set of premises (i.e., knowledge base) *KB*
  - Check if Q has been proven by *KB*,
  - If not yet, continue proving all the conditions of a rule (in *KB*) whose conclusion is *Q*
- Avoid loops
  - Check if the new expressions have been included in the list of expressions to prove? – If yes, then do not include them again!
- Avoid proving again to an expression
  - Has previously been proven true
  - Has previously been proven unsatisfiable (i.e., false) in *KB*

Deduction in propositional logic

Backward reasoning (chaining)

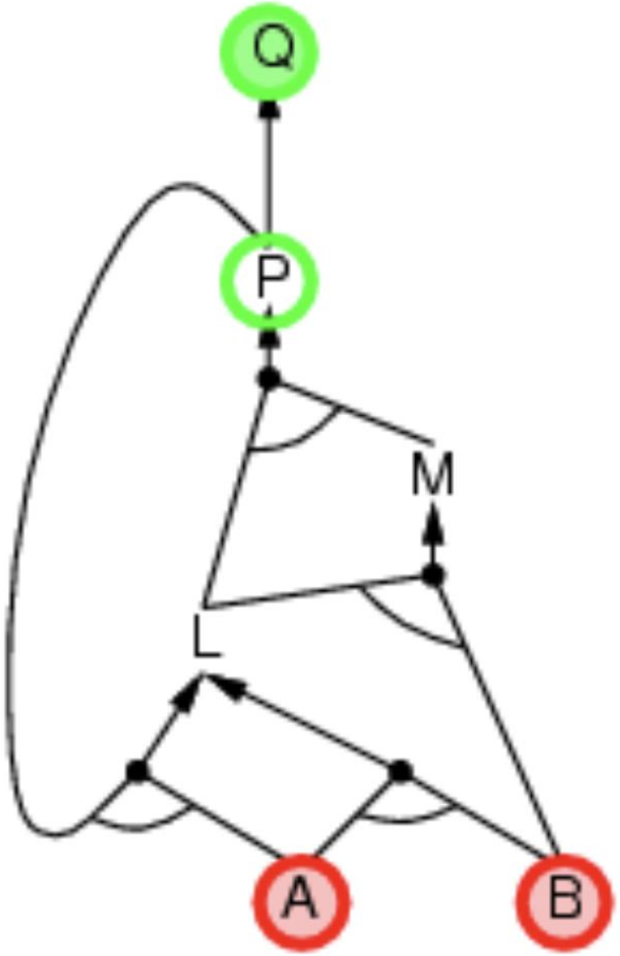
$$
\begin{aligned}
&P \Rightarrow Q \\
&L \wedge M \Rightarrow P \\
&B \wedge L \Rightarrow M \\
&A \wedge P \Rightarrow L \\
&A \wedge B \Rightarrow L \\
&A \\
&B
\end{aligned}
$$



Deduction in propositional logic

Backward reasoning (chaining)

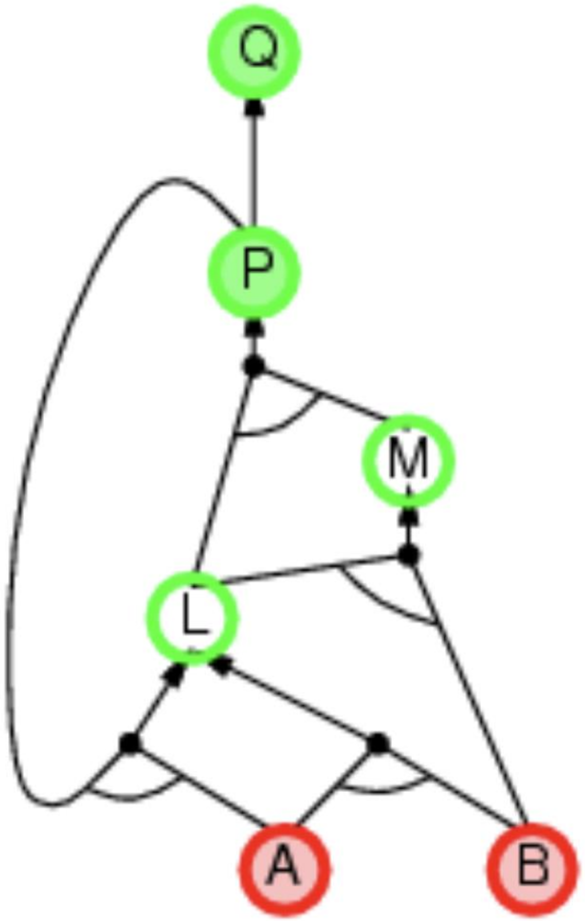
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Deduction in propositional logic

Backward reasoning (chaining)

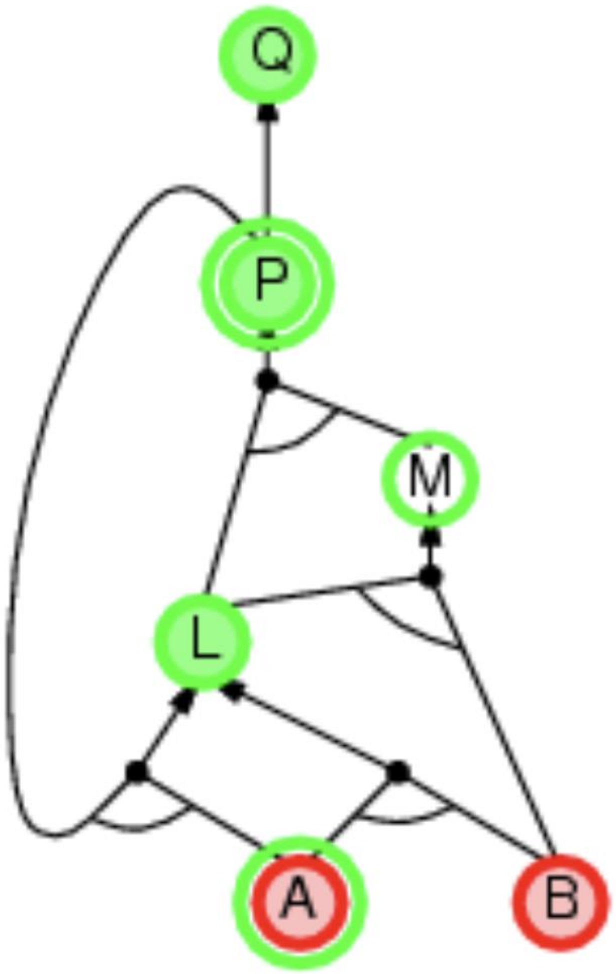
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Deduction in propositional logic

Backward reasoning (chaining)

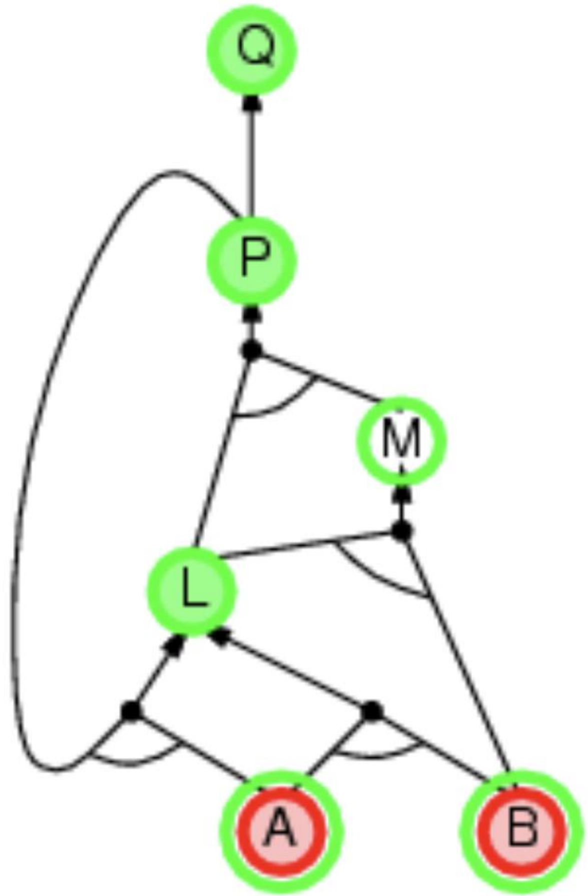
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Deduction in propositional logic

Backward reasoning (chaining)

$$
\begin{aligned}
&P \Rightarrow Q \\
&L \land M \Rightarrow P \\
&B \land L \Rightarrow M \\
&A \land P \Rightarrow L \\
&A \land B \Rightarrow L \\
&A \\
&B
\end{aligned}
$$



Deduction in propositional logic

Forward Chaining and backward chaining?

- Forward reasoning is a data-driven process
  - Example: object recognition, decision making
- Forward reasoning may perform many redundant inference steps – irrelevant (unnecessary) to the proving goal
- Backward reasoning is a goal-driven process, suitable for problem solving

Thank you!

You're now ready to explore the exciting world of AI!