ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

IT3160E
Introduction to Artificial Intelligence

Chapter 4 – Knowledge and inference

Part 2: Propositional logic

Lecturer:

Muriel VISANI

Department of Information Systems
School of Information and Communication Technology - HUST

Content of the course

- Chapter 1: Introduction
- Chapter 2: Intelligent agents
- Chapter 3: Problem Solving
    - Search algorithms, adversarial search
    - Constraint Satisfaction Problems
- Chapter 4: Knowledge and Inference
    - Knowledge representation
    - Propositional logic and first-order logic
- Chapter 5: Uncertain knowledge and reasoning
- Chapter 6: Advanced topics
    - Machine learning
    - Computer Vision

Outline

- Chapter 4 – part 1: Knowledge representation
- Chapter 4 – part 2: Propositional logic
  - Definitions
  - Syntax in propositional logic
  - Semantic in propositional logic
  - Logic inference
    - Some definitions
    - Inference in propositional logic
      - Introduction
      - Forward chaining

        Exercise
      - Backward chaining

        Exercise
  - Discussion
  - Homework

Goal of this Lecture

Goal	Description of the goal or output requirement	Output division/ Level (I/T/U)
M1	Understand basic concepts and techniques of AI	1.2

Propositional logic

Definitions

Recall: Knowledge-based Agents

- Know about the world
  - They maintain a collection of facts (sentences) about the world, their Knowledge Base, expressed in some formal language
- Reason about the world
  - They are able to derive new facts from those in the KB using some inference mechanism
- Act upon the world
  - They map percepts to actions by querying and updating the KB

What is Logic ?

- A logic is a triplet <L,S,R>
  - L, the language of the logic, is a class of sentences described by a precise **syntax**, usually a **formal grammar**
  - S, the logic's semantic, describes the meaning of elements in L
  - R, the logic's inference system, consisting of **derivation rules** over L
- Examples of logics:
  - Propositional, First Order, Higher Order, Temporal, Fuzzy, Modal, Linear, …

What is a Logic Language?

- All languages have a set of **symbols**, meanings and rules
  - For example, the letter ‘A’ is part of English, but not part of Chinese
  - For example, English is a Subject-Verb-Object language, while Arabic is Subject-Object-Verb.
  - For example, ‘snow’ means snow in English, but ‘schnee’ means snow in German
- The **syntax** of a language is the grammar of the language
- The **semantics** of a language is the meaning of the significant parts
- Propositional Logic (PL) is a **formal language** (not natural language like English)
- It is possible to translate sentences of most natural languages, (English, Vietnamese, French…) into PL

Propositional logic

Syntax in propositional logic

The Syntax of Propositional Logic

- PL is a language that focuses on a small set of expressions defined using **symbols:**
  - Propositional variables: A, B,…, P, Q,…
  - Logical constants: TRUE, FALSE
  - Propositional connectives and grouping symbols:
    - **Propositional connectives:**
      - ‘→’ arrow
      - ‘¬’ broken arrow
      - ‘≡’ triple bar
      - ‘∧’ carrot
      - ‘∨’ wedge
    - **Grouping Symbols:**
      - ‘(,)’ parentheses, and ‘[,]’ brackets

The Syntax of Propositional Logic

- **Symbols**
  - Propositional variables: A, B,…, P, Q,…
  - Logical constants: TRUE, FALSE
  - Propositional connectives and grouping symbols: (‘$\rightarrow$’, ‘$\neg$’, ‘$\equiv$’, ‘$\wedge$’, and ‘$\vee$’) and (‘(,)’, ‘[,]’)

- **Sentences**
  - Each propositional variable is a sentence (*a.k.a.* logical expression)
  - Each logical constant is a sentence
  - If **α** and **β** are sentences, then the following are also sentences: (**α**), $\neg$ **α**, **α** $\wedge$ **β**, **α** $\vee$ **β**, **α** $\rightarrow$ **β** and **α** $\equiv$ **β**

Formal Language of Propositional Logic

- **Symbols**
  - Propositional variables (sentences): A, B,…, P, Q,…
  - Logical constants: denoted T, F
  - Propositional connectives and grouping symbols: (‘$\rightarrow$’, ‘$\neg$’, ‘$\equiv$’, ‘$\wedge$’, and ‘$\vee$’) and (‘(,)’, ‘[,]’)
- **Formal Grammar**

  “simple” sentence “composite” sentence

  - Sentence : Asentence | Csentence
  - Asentence : TRUE | FALSE | A | B|…
  - Csentence : (Sentence) | $\neg$ Sentence | Sentence *Connective* Sentence
  - Where *Connective* : (‘$\rightarrow$’, ‘$\equiv$’, ‘$\wedge$’, or ‘$\vee$’)

Rules for Well-Formed Sentences

- Formal grammar (cont'd):

1. All propositional letters A….Z are atomic, well-formed sentences
2. If P and Q are well formed, then so are the following:
   I. $\neg P$
   II. $(P \equiv Q)$
   III. $(P \wedge Q)$
   IV. $(P \vee Q)$
   V. $(P \rightarrow Q)$
3. Nothing is a well-formed sentence, unless it derives from (1) or (2)

Examples

- Not well-formed

1. $P\neg$
2. QP
3. $\wedge R$
4. $A\rightarrow$
5. $\equiv\vee R)$

- Well-formed

1. $(P \rightarrow (Q \wedge R))$
2. $(V \equiv (\neg R \wedge S))$
3. $(Q \wedge (R \vee S))$
4. $(S \rightarrow (R \rightarrow T))$
5. $(P \rightarrow \neg(R \equiv S))$

Propositional logic

Semantics in propositional logic

The Semantics of PL

- PL is a language that only focuses on **propositional connectives** and operators. In English the main propositional connectives are ‘and’, ‘or’, ‘not’, ‘if…, then..’, and ‘if and only if’
  - Translation English to PL:
    - ‘and’ = ‘$\wedge$’
    - ‘or’ = ‘$\vee$’
    - ‘not’ = ‘$\neg$’
    - ‘if…, then..’ = ‘$\rightarrow$’
    - ‘if and only if’ = ‘$\equiv$’
- Since PL is only focused on these terms it only has a semantics for these terms (‘$\rightarrow$’, ‘$\neg$’, ‘$\equiv$’, ‘$\wedge$’, and ‘$\vee$’)

The Semantics of PL

- The semantics for PL is **binary** and **exclusive**
  - PL is about facts in the world
    - Examples: It's sunny, John is married
  - There are only two truth-values for these facts:
    - **TRUE** and **FALSE** (binary)
    - No statement is both **TRUE** and **FALSE** (exclusive)
- Propositional variables stand for basic facts
- Sentences are made of
  - Propositional variables (A, B,…)
  - Logical constants (TRUE, FALSE)
  - Propositional connectives ('$\rightarrow$', '$\neg$', '$\equiv$', '$\wedge$', and '$\vee$')
  - Possibly, grouping symbols: '(,)' and '[,]'

Semantic of Propositional Logics: Truth-Tables

- In order to define the propositional connectives, one needs to use **truth-tables**
- A truth-table is a table for visually displaying the distribution of truth and falsity across a compound formula given the basic inputs from the atomic letters (where T= TRUE and F=FALSE)

p	*q*	
T	T	
T	F	
F	T	
F	F	

Broken Arrow, Negation

- The definition of broken arrow is intended to capture the logical meaning of the word ‘not’, and the function of negation
- The output is the opposite of the input

p	¬ ***p***
T	F
F	T

Carrot, Conjunction

- The definition of carrot is intended to capture the logical meaning of the word ‘and’, and the function of conjunction
- The output is true only if both inputs are true

p	q	$(p \wedge q)$
T	T	T
T	F	F
F	T	F
F	F	F

Wedge, Disjunction

- The definition of wedge is intended to capture the logical meaning of the word ‘or’, and the function of disjunction
- The output is true as long at least one input is true

p	q	$(p \vee q)$
T	T	T
T	F	T
F	T	T
F	F	F

Arrow, Material Conditional

- The definition of the arrow is intended to capture the logical meaning of the phrase 'if…., then…', and the function of material conditional
- The output is false only when the first input is true, and the second input is false
  - Some definitions (in red)

⚠ Interesting result : $(p \to q) \equiv \neg p \vee q$

p: premise

q: conclusion

p	***q***	**$(p \to q)$**
T	T	T
T	F	F
F	T	T
F	F	T

Logical rule

Triple Bar, Biconditional

- The definition of triple bar is intended to capture the logical meaning of the phrase ‘if and only if’, and the function of biconditional
- The output is true just in cases the inputs are the same

p	q	$(p \equiv q)$
T	T	T
T	F	F
F	T	F
F	F	T

Composition of propositional connectives

- Note: More refined logical connectives can be derived from propositional connectives
  - For instance, we can derive the XOR (exclusive OR) connective such as:
    - p XOR q: $(\neg p \wedge q) \vee (p \wedge \neg q)$
    - Exercise: Prove that, using two truth-tables

p	***q***	$(\neg p \wedge q) \vee (p \wedge \neg q)$
T	T	F
T	F	T
F	T	T
F	F	F

Semantic of Propositional Logic

- The meaning of TRUE is always T, the meaning of FALSE is always F
- The meaning of a propositional variable is either T or F
  - Depends on the interpretation
    - assignment of Boolean values to propositional variables
- The meaning of a sentence is either T or F
  - Depends on the interpretation

Semantic of Propositional Logic

- Satisfiability
  - A sentence is **satisfiable** if it is true under some interpretation
  - Ex: $P \lor H$ is **satisfiable**

    $P \land \neg P$ is **unsatisfiable** (not satisfiable)
- Validity
  - A sentence is **valid** if it is true in every interpretation
  - Ex: $((P \land H) \land \neg A) \rightarrow P$ is **valid**

    $P \land H$ is **not valid**

Semantic of Propositional Logic

- Entailment
  - Given
    - A set of sentences Γ
    - A sentence ψ
  - We write

$$\Gamma \models \psi$$

    if and only if every interpretation that makes all sentences in Γ true also makes ψ true

  - We said that Γ entails ψ

Semantic of Propositional Logic

- Difference between entailment and implication

Implication

Implication ($\rightarrow$) is a function on statements / sentences P and Q that can be true or false

$(P \rightarrow Q)$ is true iff $(\neg P \vee Q)$ is true

As a result, $(P \rightarrow Q)$ is false only if P is true, and Q is false (it is true otherwise)

Example:

P: I am writing these words

Q: I am human

In **this course's** interpretation, $P \rightarrow Q$

(not always in Facebook's interpretation ☺)

Entailment (logical consequence)

Entailment ($\vDash$) is a relation between sets of statements Γ and a statement ψ

$(\Gamma \vDash \psi)$ is true iff every interpretation that makes all $\varphi \in \Gamma$ true, makes ψ true

As a result, $(\Gamma \vDash \psi)$ is true in the case where it's *impossible* to make all of Γ true and ψ false

Example:

Γ: the president was assassinated

ψ: the president is dead

In **every interpretation**, where all sentences in Γ are true, then ψ is true

Therefore, $(\Gamma \vDash \psi)$

Semantic of Propositional Logic

- Satisfiability vs. Validity vs. Entailment
  - A sentence is **valid** if it is true in every interpretation
    - ψ is valid iff TRUE $\models \psi$ (also written $\models \psi$)
  - A sentence is **unsatisfiable** if there exists no interpretation in which it is true
    - ψ is unsatisfiable iff $\psi \models$ False
    - $\Gamma \models \psi$ iff $\Gamma \wedge \{\neg \psi\}$ is unsatisfiable

Propositional logic

Logical inference:

Some definitions

Logical Inference

- **Logical inference problem:**
  - Given:
    - a **knowledge base** KB (a set of sentences) and
    - a sentence α (called a theorem)
  - Does KB **semantically entail** α (KB |= α)?

- **In other words:** In all interpretations where all sentences in the KB are true, is also α true?

- Proving this is the objective of logical inference
  - But first, let's give some more definitions!

Positive and negative literals

- In mathematical logic, a **literal** is either:
  - an atomic formula (positive literal), *e.g.* P
  - or its negation (negative literal), *e.g.* $\neg P$

- In propositional logic, a literal is a propositional variable P, Q, R, … Z (or its negation)

Horn clauses

- A **clause** is a disjunction of literals
  - Positives or negative literals connected by $\vee$
- A **Horn clause** is a clause with **at most 1 positive** literal:

$$(\neg r_1 \vee \neg r_2 \vee \ldots \vee \neg r_n) \vee h, \text{ where } n \in \mathbb{N}$$

  - Therefore, there are three types of Horn clauses:
    - **1- Strict Horn clauses**, that contain 1 positive literal and $n \geq 1$ negative literals:

$$(\neg r_1 \vee \neg r_2 \vee \ldots \vee \neg r_n) \vee h, \quad \text{where } n \geq 1$$

    - Intuitively, they represent if… then rules
    - They enable to **deduce** new facts from existing facts:

$$(r_1 \wedge r_2 \wedge \ldots \wedge r_n) \rightarrow h$$

Horn clauses

- **Exercise**
  - Using truth table, show that a **strict Horn clause** with 2 negative litterals
    - $\neg r1 \vee \neg r2 \vee h$
  - Is strictly equivalent to
    - $r1 \wedge r2 \rightarrow h$

Horn clauses

- **Exercise**
  - Using truth table, show that a **strict Horn clause** with 2 negative litterals
    - $\neg r1 \vee \neg r2 \vee h$
  - Is strictly equivalent to
    - $r1 \wedge r2 \rightarrow h$

Horn clauses

- A **clause** is a disjunction of literals
  - Positives or negative literals connected by $\vee$
- A **Horn clause** is a clause with **at most 1 positive** literal
  - Therefore, there are three types of Horn clauses:
    - **1- Strict Horn clauses**, that contain 1 positive literal and $n \geq 1$ negative literals
    - **2- Positive Horn clauses**, that contain 1 positive literal h and $n = 0$ negative literals:
    - They are **facts**
      - Propositional variables (atomic propositions) that can be either True or False

Fact $\neq$ True

Horn clauses

- A **clause** is a disjunction of literals
  - Positives or negative literals connected by $\vee$
- A **Horn clause** is a clause with **at most 1 positive** literal
  - Therefore, there are three types of Horn clauses:
    - **1- Strict Horn clauses**, that contain 1 positive literal and $n \geq 1$ negative literals
    - **2- Positive Horn clauses**, that contain 1 positive literal and $n = 0$ negative literals
    - **3- Negative Horn clauses**, that **only** negative literals

$$(\neg r_1 \vee \neg r_2 \vee \ldots \vee \neg r_n) \vee \neg q$$

- Represent questions to answer, where q is the question
- To prove q, *i.e.* $\{r_1, r_2, \ldots r_n\} \models q$, one can prove that:

$$\{r_1, r_2, \ldots r_n, \neg q\} \models \emptyset$$

Horn form

- In **logical programming**, composites of Horn clauses are a particular case for which we know effective resolution algorithms
- A KB follows a **Horn (normal) form** if it is a conjunction of Horn clauses
  - **Note**: the following Horn form

$$(A \vee \neg B) \wedge (\neg A \vee \neg C \vee D)$$

can also be written as (with implications)

$$(B \rightarrow A) \wedge ((A \wedge C) \rightarrow D)$$

Propositional logic

Inference in Propositional Logic:

Introduction

Inference in propositional logic

- In propositional logic, inference is based on the modus ponens:

$$\frac{p \qquad p \rightarrow q}{q}, \quad \equiv \quad \frac{p \qquad (\neg p \vee q)}{q}$$

  - *From p and p -> q, one can deduce q*

- Resolution techniques are **complete** for KBs in the Horn form, *e.g.*

$$(A \vee \neg B) \wedge (\neg A \vee \neg C \vee D)$$

- Resolution in propositional logic

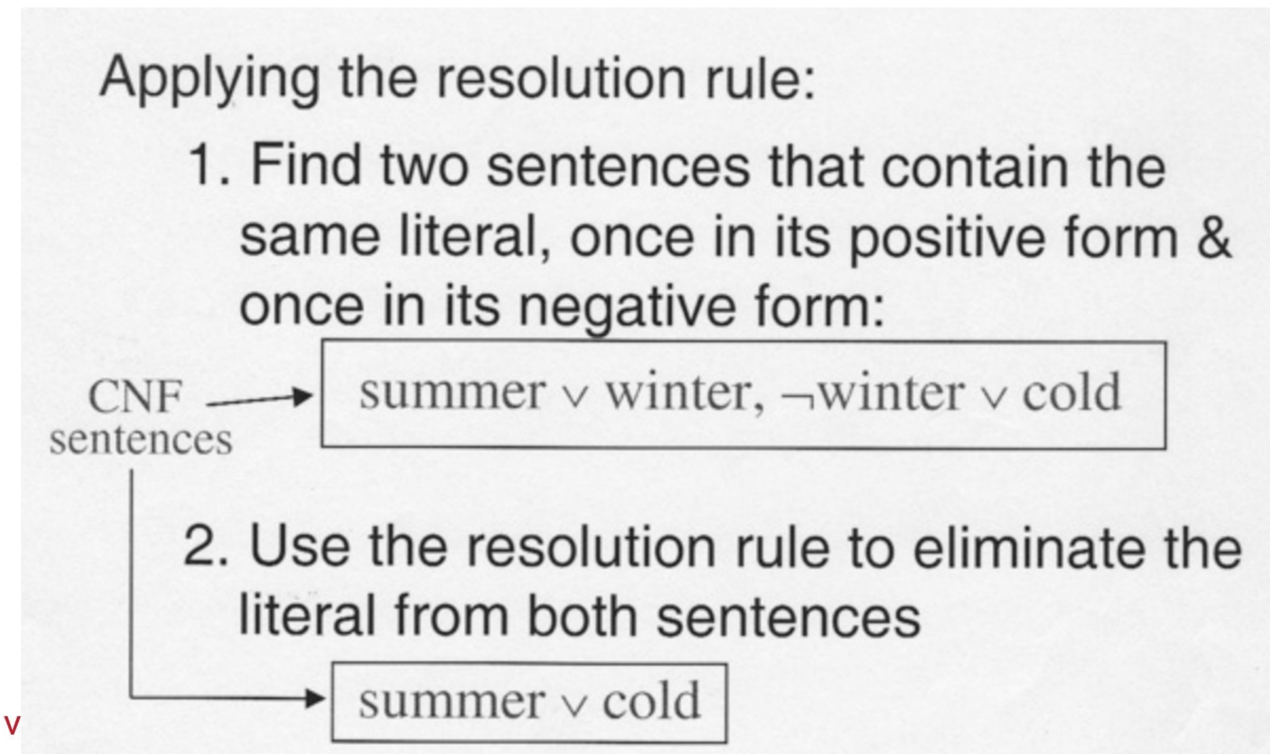
$$\frac{(p \vee L_1 \vee \dots L_n) \qquad (\neg p \vee M_1 \vee \dots \vee M_k)}{(L_1 \vee \dots L_n \vee M_1 \vee \dots \vee M_k)}$$

Resolvent of $(p \vee L_1 \vee \dots L_n)$ and $(\neg p \vee M_1 \vee \dots \vee M_k)$

  - Permuting the literals does not change anything

Inference in propositional logic

- In “easier” words: the resolution rule for propositional logic
  - In Boolean logic, a formula is in conjunctive normal form (CNF) if it is a **conjunction** of one or more clauses, where a clause is a **disjunction** of literals: it is an AND of ORs.



Inference in propositional logic

- Example: $$\frac{p \qquad \neg p}{\perp}$$ Contradiction symbol
- Exercise:

$$\frac{(a \vee \neg b \vee \neg c) \qquad (b \vee e \vee \neg f)}{???}$$

Inference in propositional logic

- Now, we are going to study two resolution techniques for inference in propositional logic:
  - Forward chaining and backward chaining
  - Basically, defines in which order to consider the propositions during inference

Propositional logic

Inference in Propositional Logic:

Forward chaining

Forward Chaining

- **Forward chaining** is a form of reasoning which start with **atomic sentences** in the KB…
- … then applies **inference rules** in the forward direction…
- … until the **goal** *q* (**theorem**) is reached

Forward Chaining

- Example:
  - Goal: Proving that Q is True, given the following KB:
    - Question: does this KB follow a Horn form?
      - Answer:
      - Prove it!

$$\text{rules}\left\{\begin{array}{l} P \rightarrow Q \\ L \wedge M \rightarrow P \\ B \wedge L \rightarrow M \\ A \wedge P \rightarrow L \\ A \wedge B \rightarrow L \end{array}\right.$$

$$\text{facts}\left\{\begin{array}{l} A \\ B \end{array}\right.$$

Forward Chaining

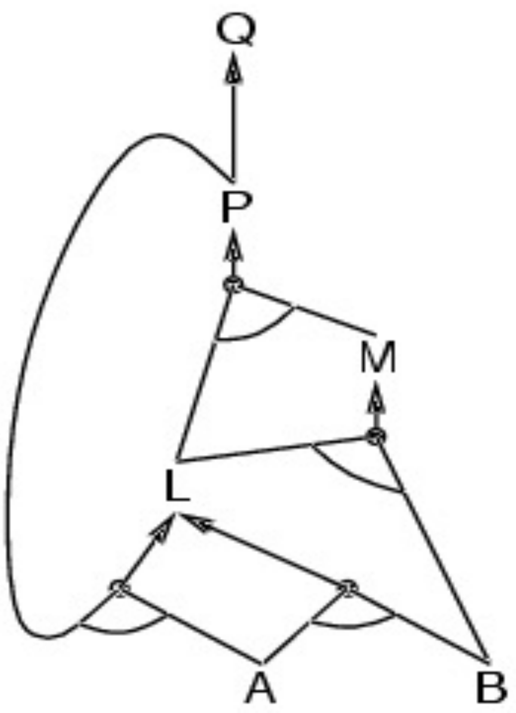
- Example:
  - Goal: Proving that Q is True, given the following KB:

rules

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

facts

$$A$$
$$B$$



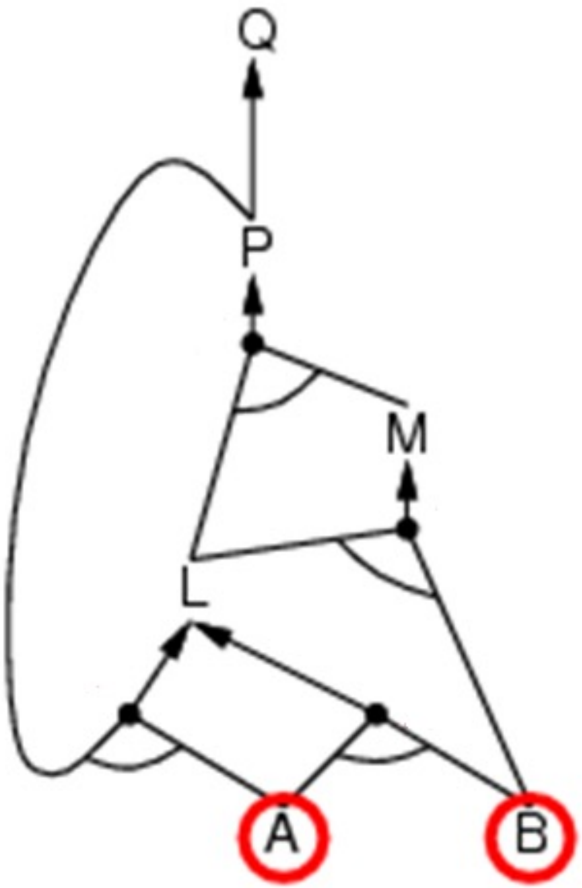
Forward Chaining

rules

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

facts

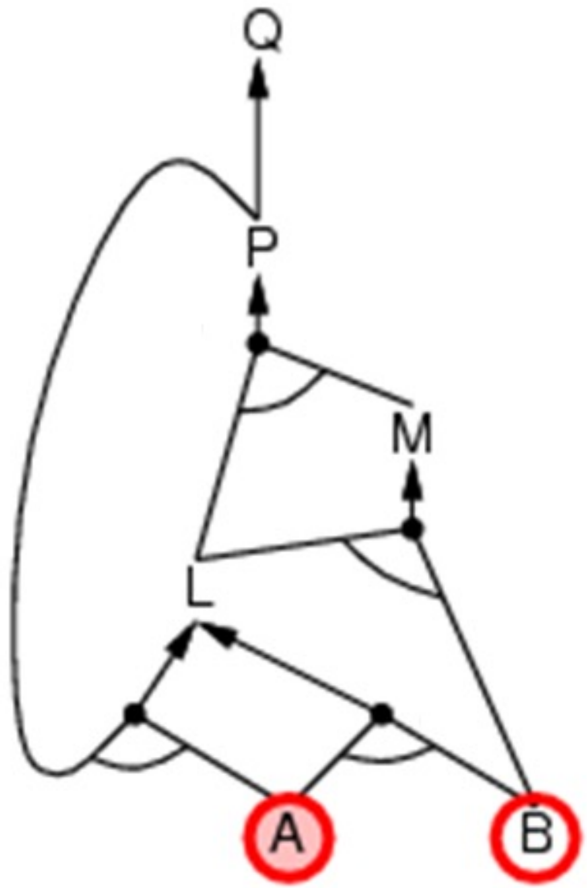
$$A$$
$$B$$



Forward Chaining

rules
$$P \to Q$$
$$L \wedge M \to P$$
$$B \wedge L \to M$$
$$A \wedge P \to L$$
$$A \wedge B \to L$$

facts
$$A$$
$$B$$



Forward Chaining

Modus ponens: $\dfrac{A \wedge B \qquad \neg (A \wedge B) \vee L}{L}$

rules

$P \rightarrow Q$

$L \wedge M \rightarrow P$

$B \wedge L \rightarrow M$

$A \wedge P \rightarrow L$

$A \wedge B \rightarrow L$

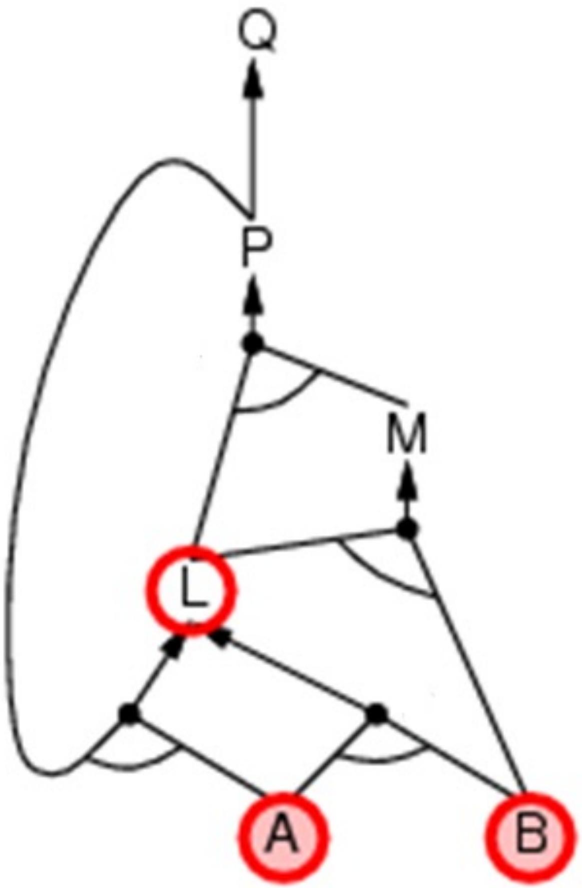
facts

A

B

Inferred facts

L



Forward Chaining

Modus ponens: $\dfrac{B \wedge L \qquad \neg(B \wedge L) \vee M}{M}$

rules

$P \to Q$

$L \wedge M \to P$

$B \wedge L \to M$

$A \wedge P \to L$

$A \wedge B \to L$

facts

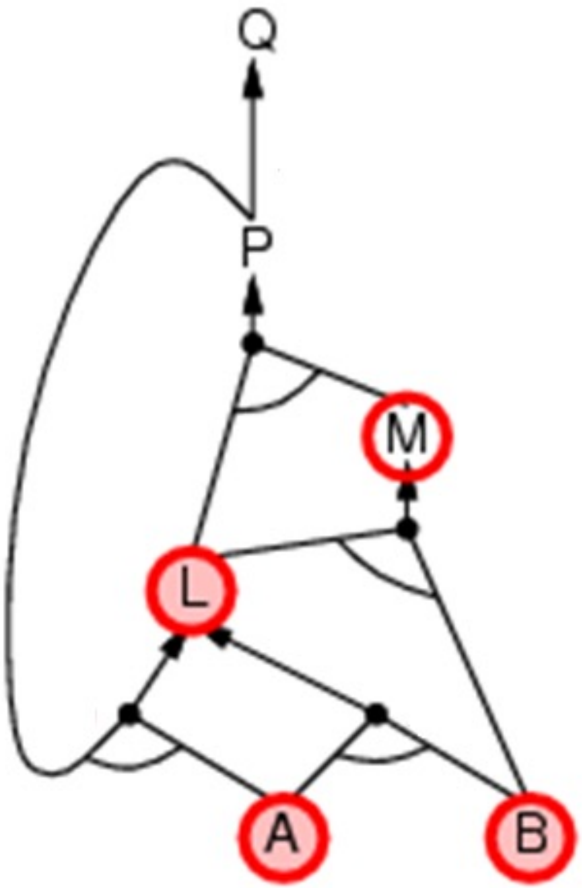
A

B

Inferred facts

L

M



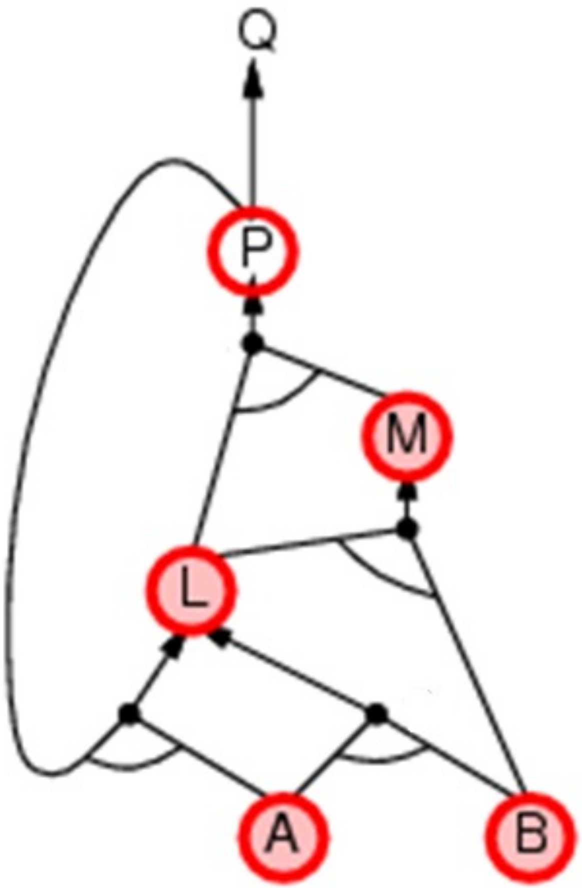
Forward Chaining

Modus ponens: $\dfrac{L \wedge M \qquad \neg(L \wedge M) \vee P}{P}$

rules
$$P \to Q$$
$$L \wedge M \to P$$
$$B \wedge L \to M$$
$$A \wedge P \to L$$
$$A \wedge B \to L$$

facts
$$A$$
$$B$$

Inferred facts
L
M
P



Forward Chaining

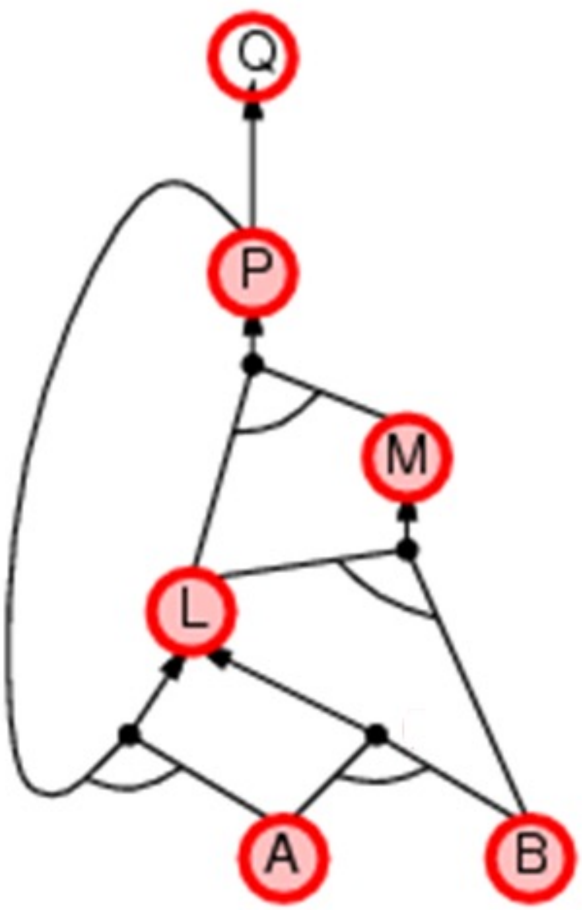
Modus ponens: $\dfrac{P \qquad \neg(P) \vee Q}{Q}$

Remark:
We could prove that Q is True without using all the clauses in the KB

rules
$$P \rightarrow Q$$
$$L \wedge M \rightarrow P$$
$$B \wedge L \rightarrow M$$
$$A \wedge P \rightarrow L$$
$$A \wedge B \rightarrow L$$

facts
$$A$$
$$B$$

Inferred facts
- L
- M
- P
- Q



Forward Chaining algorithm

```
function PL-FC-ENTAILS?(KB, q) returns true or false
  local variables: count, a table, indexed by clause, initially the number of premises
                   inferred, a table, indexed by symbol, each entry initially false
                   agenda, a list of symbols, initially the symbols known to be true

  while agenda is not empty do
      p ← POP(agenda)
      unless inferred[p] do
          inferred[p] ← true
          for each Horn clause c in whose premise p appears do
              decrement count[c]
              if count[c] = 0 then do
                  if HEAD[c] = q then return true
                  PUSH(HEAD[c], agenda)
  return false
```

Forward Chaining properties

- Computational complexity:
  - Runs in linear time in the number of literals in the Horn formulae
    - *I.e.* linear complexity in the size of the KB
- Completeness
  - Forward chaining is complete for KBs in the Horn form

Exercise

- Use forward chaining to prove theorem *E*, using the following KB

$$
\begin{array}{ll}
\textbf{KB:} & \text{R1: } A \wedge B \rightarrow C \\
 & \text{R2: } C \wedge D \rightarrow E \\
 & \text{R3: } C \wedge F \rightarrow G \\
\hline
 & \text{F1: } A \\
 & \text{F2: } B \\
 & \text{F3: } D
\end{array}
$$

$$\textbf{Theorem: } E \quad ?$$

Propositional logic

Inference in Propositional Logic:

Backward chaining

Backward Chaining

- **Idea** (goal reduction):
  - To prove the fact (theorem) that appears in the conclusion of a rule, …
  - … prove the premises of the rule…
    - If the premise is a conjunction, then process the conjunction conjunct by conjunct
  - … and continue, recursively.

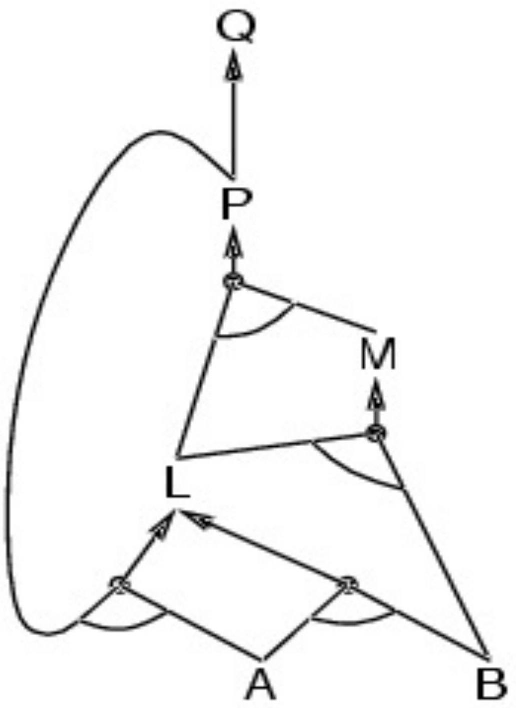
Backward Chaining

- Example:
  - Goal: Proving that Q is True, given the following KB:

rules
$$P \rightarrow Q$$
$$L \wedge M \rightarrow P$$
$$B \wedge L \rightarrow M$$
$$A \wedge P \rightarrow L$$
$$A \wedge B \rightarrow L$$

facts
$$A$$
$$B$$

Inferred facts



Backward Chaining

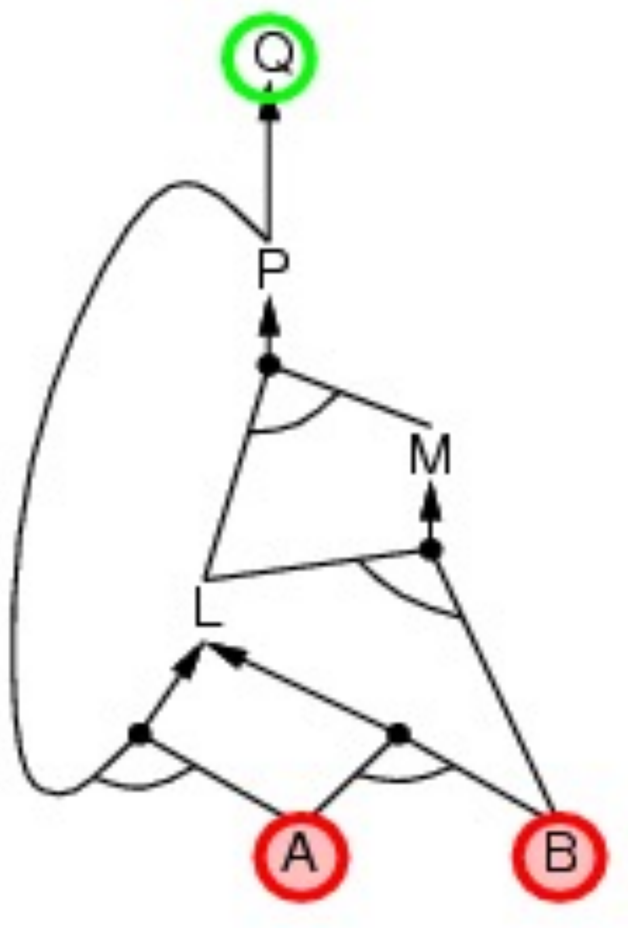
rules

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

facts

$$A$$
$$B$$

Inferred facts



Backward Chaining

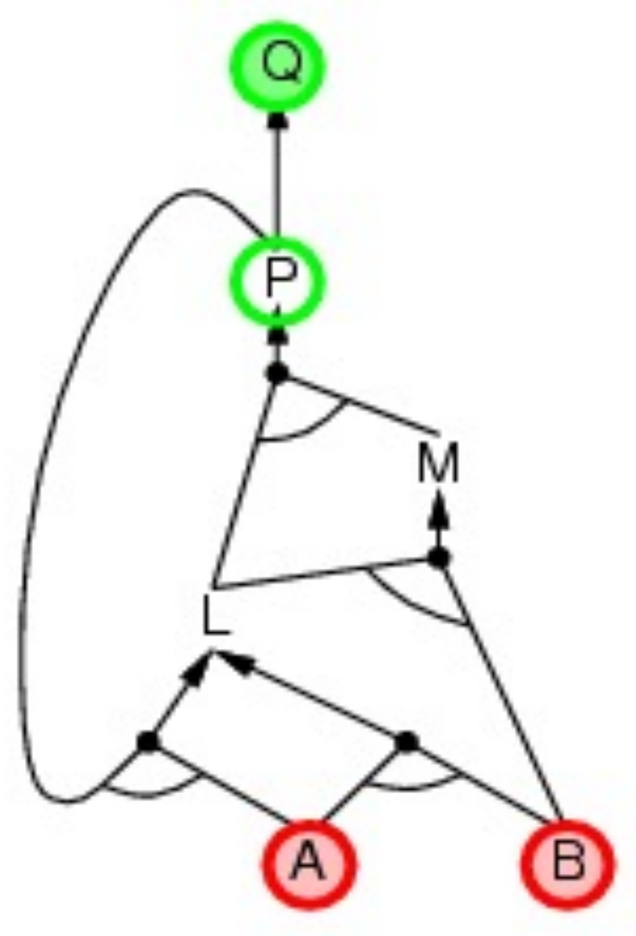
rules

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

facts

$$A$$
$$B$$

Inferred facts



Backward Chaining

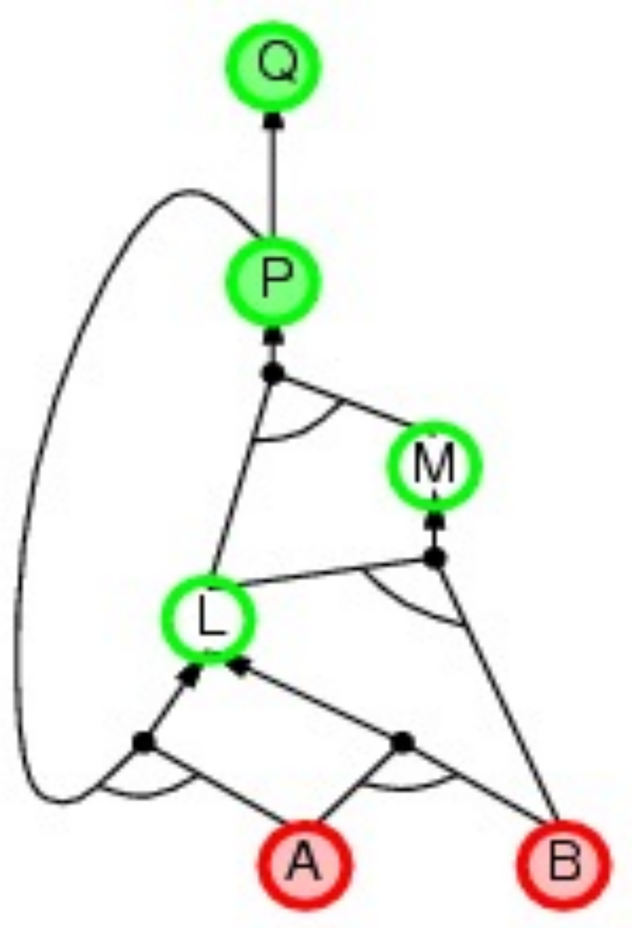
rules

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

facts

$$A$$
$$B$$

Inferred facts



Backward Chaining

rules

$P \rightarrow Q$

$L \wedge M \rightarrow P$

$B \wedge L \rightarrow M$

$A \wedge P \rightarrow L$

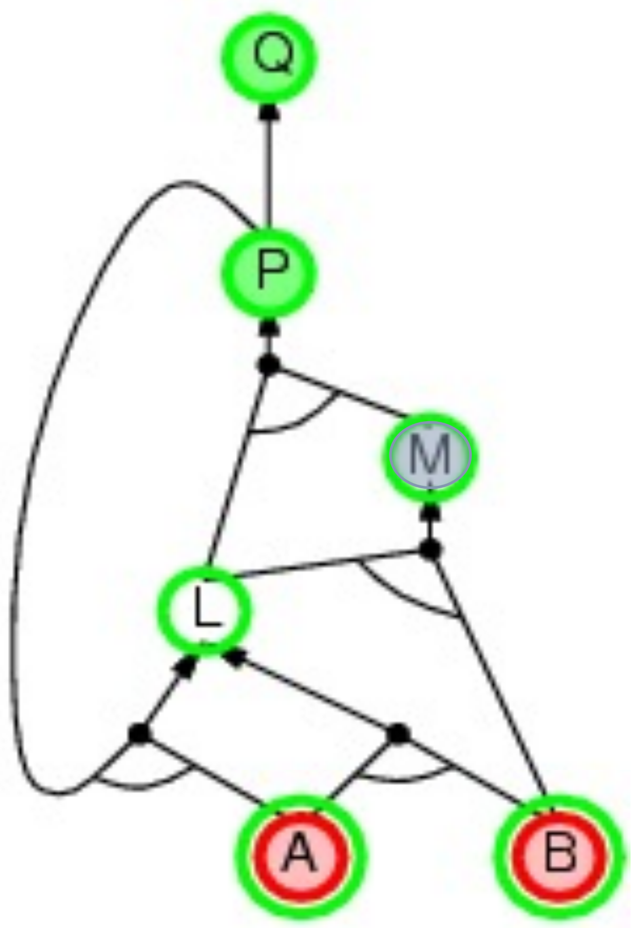
$A \wedge B \rightarrow L$

facts

A

B

Inferred facts



Backward Chaining

rules

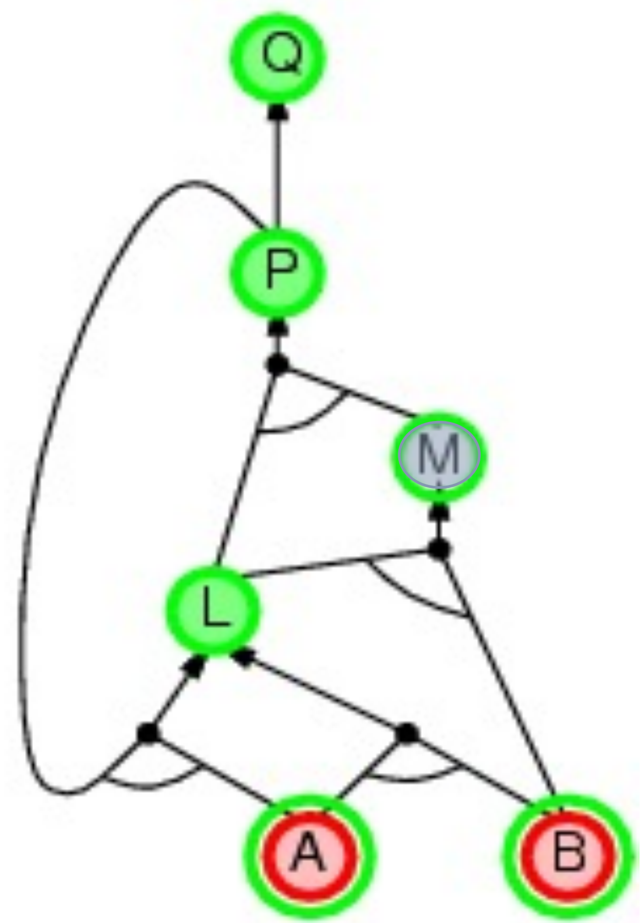
$$P \rightarrow Q$$
$$L \wedge M \rightarrow P$$
$$B \wedge L \rightarrow M$$
$$A \wedge P \rightarrow L$$
$$A \wedge B \rightarrow L$$

facts

$$A$$
$$B$$

Inferred facts

L
M
P
Q



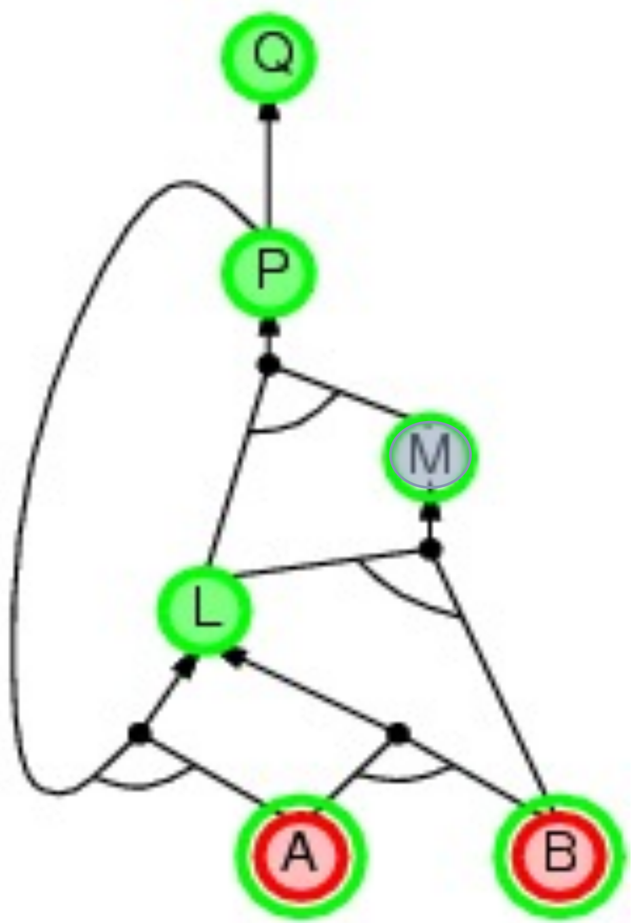
Backward Chaining

Remark:
We could prove that Q is True without using all the clauses in the KB

rules
$$P \rightarrow Q$$
$$L \wedge M \rightarrow P$$
$$B \wedge L \rightarrow M$$
$$A \wedge P \rightarrow L$$
$$A \wedge B \rightarrow L$$

facts
$$A$$
$$B$$

Inferred facts
L
M
P
Q



Backward Chaining properties

- Computational complexity:
  - Runs in (maximum) linear time in the number of literals in the Horn formulae
    - *I.e.* linear complexity in the size of the KB
      - **But, in practice, it is much less**
- Completeness
  - Backward chaining is complete for KBs in the Horn form

Exercise

- Use backward chaining to prove theorem *E*, using the following KB

$$
\begin{array}{lll}
\textbf{KB:} & \text{R1:} & A \land B \rightarrow C \\
 & \text{R2:} & C \land D \rightarrow E \\
 & \text{R3:} & C \land F \rightarrow G \\
\hline
 & \text{F1:} & A \\
 & \text{F2:} & B \\
 & \text{F3:} & D
\end{array}
$$

Theorem: E ?

Solution

KB: R1: $A \wedge B \rightarrow C$

R2: $C \wedge D \rightarrow E$

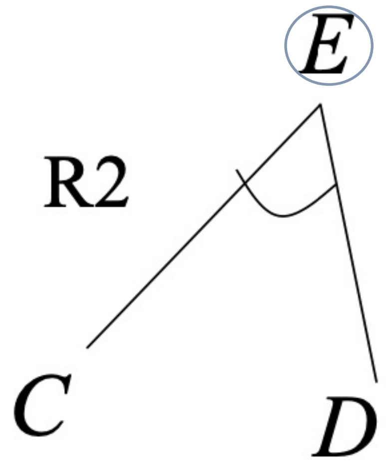
R3: $C \wedge F \rightarrow G$

F1: A

F2: B

F3: D

Theorem: E ?



Propositional logic

Discussion

Forward *v.s.* backward chaining

- **Forward** Chaining is **data-driven**, automatic, does not take into account the goal
  - *E.g.*, routine decisions
  - Might do a lot of work that is not relevant for the goal
    - Complexity is linear in the size of the KB

- **Backward** Chaining is **goal-driven**
  - More appropriate for problem-solving
  - *E.g.*, where are my keys?
  - Complexity can be much less than linear in the size of the KB

KB agents based on propositional logic

- Propositional Logic allows us to build KB agents that can answer queries about the worl by inferring new facts from the known ones
- Example from the last lecture: MYCIN

Facts: The stain of the organism is gram-positive
The growth conformation of the organism is chains

Rules: **(If)** The stain of the organism is gram-positive $\wedge$
The morphology of the organism is coccus $\wedge$
The growth conformation of the organism is chains
(Then) $\Rightarrow$ The identity of the organism is streptococcus

Limitations linked to the Horn forms

- The Horn form works with propositional symbols:
  - Only non-negated propositional symbols may occur in the rules' premises and in the conclusions
  - Only non-negated propositions can be used as facts
- Question:
  - How to express sentences such as:
    - If it is not windy, then we will play tennis ??
  - Solution 1: create an explicit proposition for Not_Windy
  - Solution 2: the negation of the propositional symbol will become true if we fail to prove that the propositional symbol is true (cannot be used in every context)

Propositional logic

Homework

Exercise 1

- Use forward chaining to reach the goal C with the following KB, and represent it using a table showing (in lines):
  - Iteration number (starting at 0)
  - The current content of the KB
  - The rule that has just been triggered at this iteration

Rules:

R1	$A \rightarrow E$
R2	$B \rightarrow D$
R3	$H \rightarrow A \land F$
R4	$E \land G \rightarrow C$
R5	$E \land K \rightarrow B$
R6	$D \land E \land K \rightarrow C$
R7	$G \land K \land F \rightarrow A$

Facts:

H

K

Exercise 2

- The following is the rule set of a simple weather forecast expert system:

1	IF	*cyclone*	THEN	*clouds*
2	IF	*anticyclone*	THEN	*clear sky*
3	IF	*pressure is low*	THEN	*cyclone*
4	IF	*pressure is high*	THEN	*anticyclone*
5	IF	*arrow is down*	THEN	*pressure is low*
6	IF	*arrow is up*	THEN	*pressure is high*

Exercise 2

- **Question 1**:
  - Use **forward chaining** to reason about the weather if the KB contains:
  - Rules:

1	IF	*cyclone*	THEN	*clouds*
2	IF	*anticyclone*	THEN	*clear sky*
3	IF	*pressure is low*	THEN	*cyclone*
4	IF	*pressure is high*	THEN	*anticyclone*
5	IF	*arrow is down*	THEN	*pressure is low*
6	IF	*arrow is up*	THEN	*pressure is high*

  - Fact: *arrow is down*

Exercise 2

- **Question 2**:
  - Use **backward chaining** to show that *clear sky* is True, knowing the following KB
  - Rules:

1	IF	*cyclone*	THEN	*clouds*
2	IF	*anticyclone*	THEN	*clear sky*
3	IF	*pressure is low*	THEN	*cyclone*
4	IF	*pressure is high*	THEN	*anticyclone*
5	IF	*arrow is down*	THEN	*pressure is low*
6	IF	*arrow is up*	THEN	*pressure is high*

  - Fact: *arrow is up*

Chapter 4 – part 2

Questions

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you
for your
attention!