CS 1571 Introduction to AI
Lecture 12

Propositional logic

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Simulated annealing competition

Top 3 simulated annealing entries for HW-3 Problem 2:

- **Timothy Sweetser,**
- **Brian Taylor**
- **Rishi Sadhir**

Logical inference problem

Logical inference problem:

- **Given:**
  - a knowledge base KB (a set of sentences) and
  - a sentence α (called **a theorem)**,
- **Does a KB semantically entail** α ? $KB \models \alpha$?

In other words:

- In all interpretations in which sentences in the KB are true, is α also true?

Logical inference problem

Logical inference problem:

- **Given:**
  - a knowledge base KB (a set of sentences) and
  - a sentence α (called **a theorem)**,
- **Does a KB semantically entail** α ? $KB \models \alpha$

Approaches to solve the logical inference problem:

- **Truth-table approach**
- **Inference rules**
- **Conversion to SAT**
  - **Resolution refutation**

Properties of inference solutions

- **Truth-table approach**
  - **Blind**
  - **Exponential in the number of variables**
- **Inference rules**
  - **More efficient**
  - **Many inference rules to cover logic**
- **Conversion to SAT - Resolution refutation**
  - **More efficient**
  - **Sentences must be converted into CNF**
  - **One rule – the resolution rule - is sufficient to perform all inferences**

KB in restricted forms

If the sentences in the KB are restricted to some special forms some of the sound inference rules may become complete

Example:

- **Horn form (Horn normal form)**
  - a clause with **at most one positive literal**

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

Can be written also as:

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

- **Two inference rules that are sound and complete for KBs in the Horn normal form:**
  - **Resolution**
  - **Modus ponens**

KB in Horn form

- **Horn form:** a clause with **at most one positive literal**

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

- **Not all sentences in propositional logic can be converted into the Horn form**
- **KB in Horn normal form**:
  - Two types of propositional statements:
    - **Rules** $(\neg B_1 \vee \neg B_2 \vee \ldots \neg B_k \vee A)$

      $$\equiv$$

      $$(\neg(B_1 \wedge B_2 \wedge \ldots B_k) \vee A)$$

      $$\equiv$$

      $$(B_1 \wedge B_2 \wedge \ldots B_k \Rightarrow A)$$

    - Propositional symbols: **facts** B

KB in Horn form

- **Application of the resolution rule:**
  - Infers new facts from previous facts

$$\frac{(A \vee \neg B), B}{A} \qquad \frac{(A \vee \neg B), \quad (B \vee \neg C)}{(A \vee C)}$$

  - Resolution is **sound and complete** for inferences on propositional symbols for KB in the Horn normal form (clausal form)

- Similarly, **modus ponens is sound and complete** when the HNF is written in the implicative form

Complexity of inferences for KBs in HNF

Question:

How efficient the inferences in the HNF can be?

Answer:

Inference on propositional symbols →

Procedures linear in the size of the KB in the Horn form exist.

- Size of a clause: the number of literals it contains.
- Size of the KB in the HNF: the sum of the sizes of its elements.

Example:

$$A, B, (A \wedge B \Rightarrow C), (C \Rightarrow D), (C \Rightarrow E), (E \wedge F \Rightarrow G)$$

or

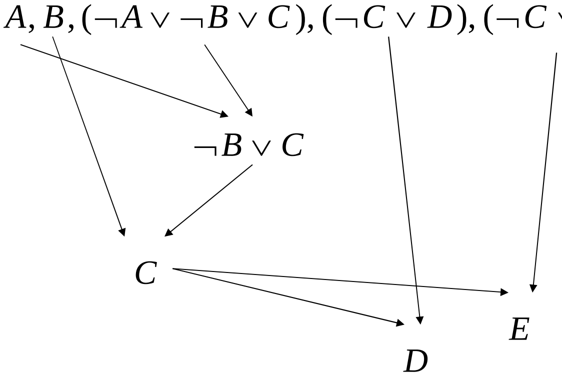
$$A, B, (\neg A \vee \neg B \vee C), (\neg C \vee D), (\neg C \vee E), (\neg E \vee \neg F \vee G)$$

The size is: 12

Complexity of inferences for KBs in HNF

How to do the inference? If the HNF (is in the clausal form) we can apply resolution.

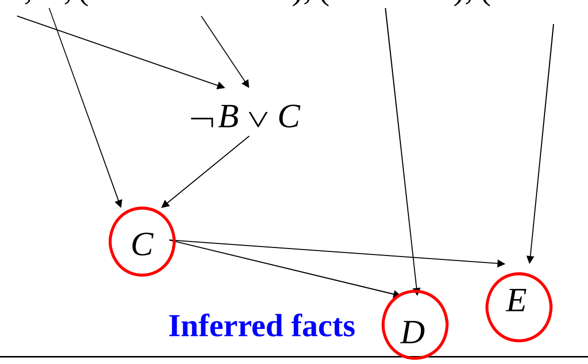
$$A, B, (\neg A \vee \neg B \vee C), (\neg C \vee D), (\neg C \vee E), (\neg E \vee \neg F \vee G)$$



Complexity of inferences for KBs in HNF

How to do the inference? If the HNF (is in the clausal form) we can apply resolution.

$$A, B, (\neg A \vee \neg B \vee C), (\neg C \vee D), (\neg C \vee E), (\neg E \vee \neg F \vee G)$$

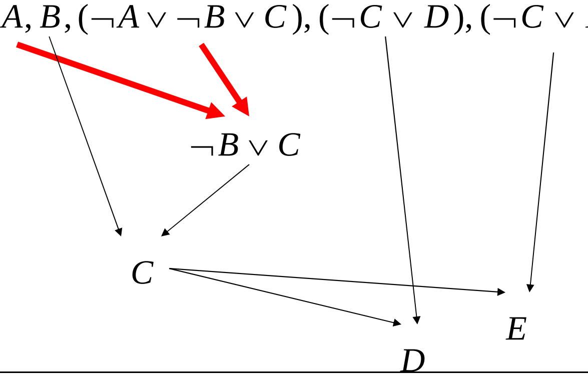


Complexity of inferences for KBs in HNF

Features:

- Every resolution is a **positive unit resolution**; that is, a resolution in which **one clause is a positive unit clause** (i.e., a proposition symbol).

$$A, B, (\neg A \vee \neg B \vee C), (\neg C \vee D), (\neg C \vee E), (\neg E \vee \neg F \vee G)$$

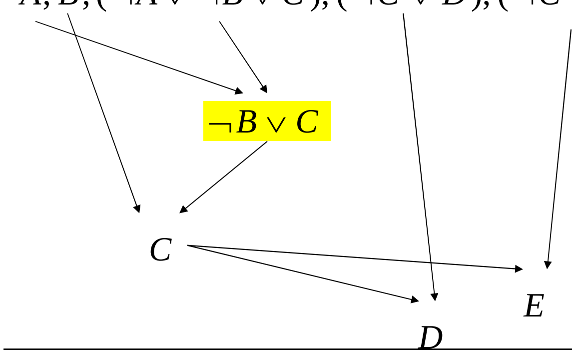


Complexity of inferences for KBs in HNF

Features:

- At each resolution, the input clause which is not a unit clause is a logical consequence of the result of the resolution. (Thus, the input clause may be deleted upon completion of the resolution operation.)

$$A, B, (\neg A \vee \neg B \vee C), (\neg C \vee D), (\neg C \vee E), (\neg E \vee \neg F \vee G)$$

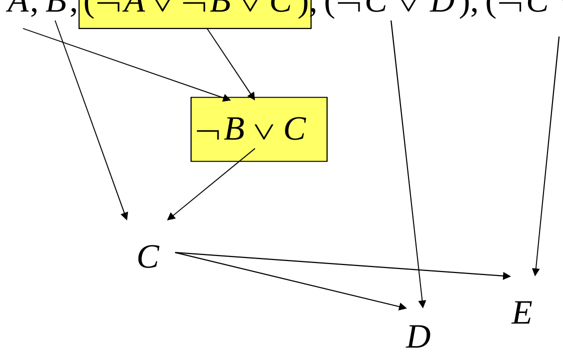


Complexity of inferences for KBs in HNF

Features:

- At each resolution, the input clause which is not a unit clause is a logical consequence of the result of the resolution. (Thus, the input clause may be deleted upon completion of the resolution operation.)

$$A, B, (\neg A \vee \neg B \vee C), (\neg C \vee D), (\neg C \vee E), (\neg E \vee \neg F \vee G)$$

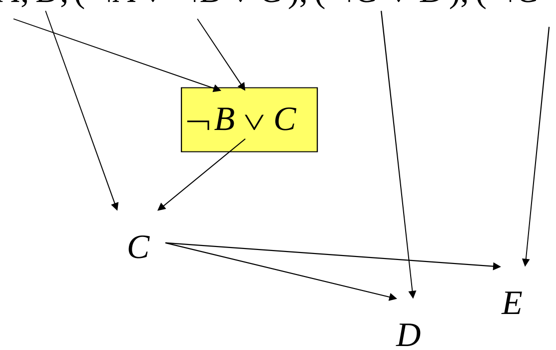


Complexity of inferences for KBs in HNF

Features:

- Following this deletion, the size of the KB (the sum of the lengths of the remaining clauses) is one less than it was before the operation.)

$A, B,$ ~~$(\neg A \vee \neg B \vee C)$~~$, (\neg C \vee D), (\neg C \vee E), (\neg E \vee \neg F \vee G)$

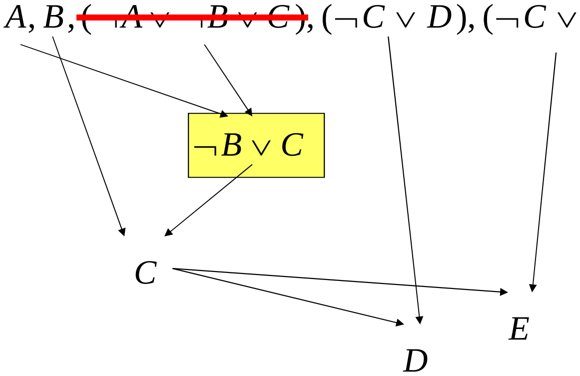


Complexity of inferences for KBs in HNF

Features:

- If n is the size of the KB, then at most n positive unit resolutions may be performed on it.

$A, B,$ ~~$(\neg A \vee \neg B \vee C)$~~$, (\neg C \vee D), (\neg C \vee E), (\neg E \vee \neg F \vee G)$



Complexity of inferences for KBs in HNF

A linear time resolution algorithm:

- **The number of positive unit resolutions is limited to the size of the formula (n)**

- **But to assure overall linear time we need to access each proposition in a constant time:**
- Data structures indexed by proposition names may be accessed in constant time. (This is possible if the proposition names are number in a range (e.g., 1..n), so that array lookup is the access operation.
- If propositions are accessed by name, then a symbol table is necessary, and the algorithm will run in time O(n · log(n)).

Forward and backward chaining

Two inference procedures based on **modus ponens** for **Horn KBs**:

- **Forward chaining**

  Idea: Whenever the premises of a rule are satisfied, infer the conclusion. Continue with rules that became satisfied.

- **Backward chaining (goal reduction)**

  Idea: To prove the fact that appears in the conclusion of a rule prove the premises of the rule. Continue recursively.

Both procedures are **complete for KBs in the Horn form** !!!

Forward chaining example

- **Forward chaining**
  Idea: Whenever the premises of a rule are satisfied, infer the conclusion. Continue with rules that became satisfied.

Assume the KB with the following rules and facts:

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 ?

Forward chaining example

Theorem: E

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

Forward chaining example

Theorem: E

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

Rule R1 is satisfied.

F4: C

Forward chaining example

Theorem: E

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

Rule R1 is satisfied.

F4: C

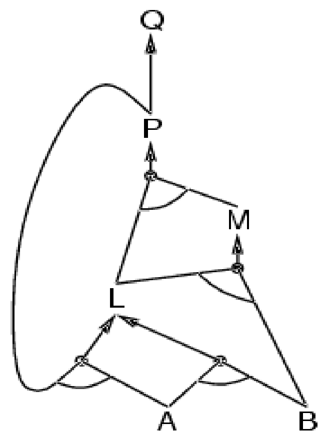
Rule R2 is satisfied.

F5: E ✓

Forward chaining

- Efficient implementation: linear in the size of the KB
- **Example:**

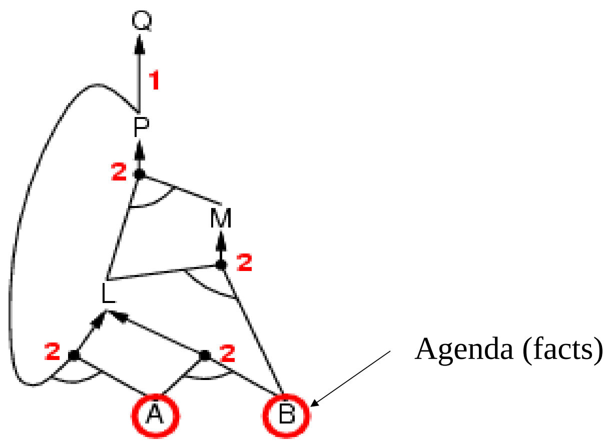
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



Forward chaining

- Count the number of facts in the antecedent of the rule

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



Forward chaining

- Inferred facts decrease the count

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

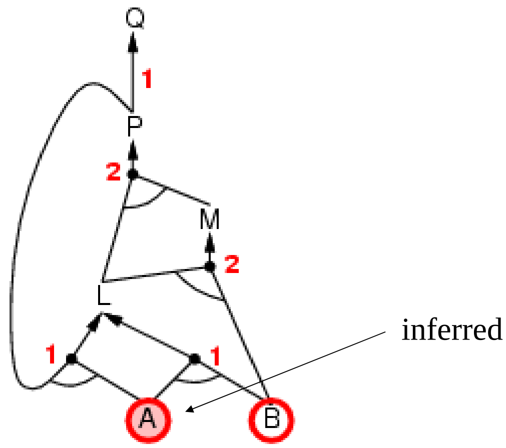
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Forward chaining

- New facts can be inferred when the count associated with a rule becomes 0

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

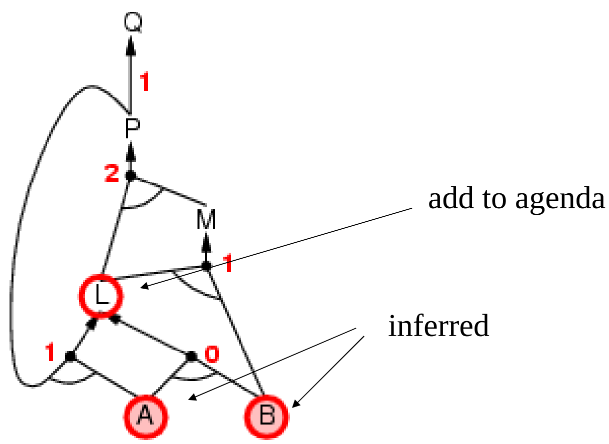
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Forward chaining

•

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

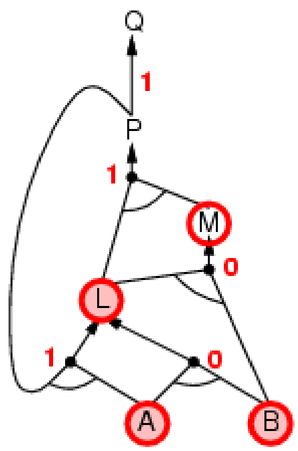
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Forward chaining

•

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

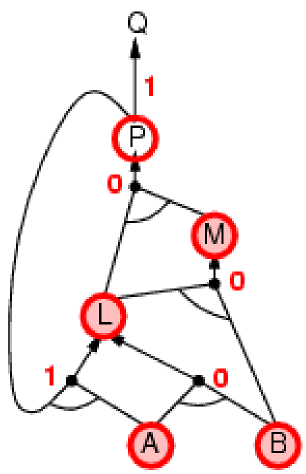
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Forward chaining

•

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

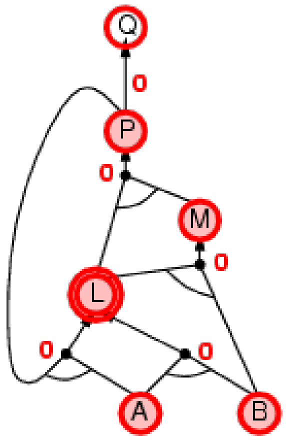
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Forward chaining

•

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

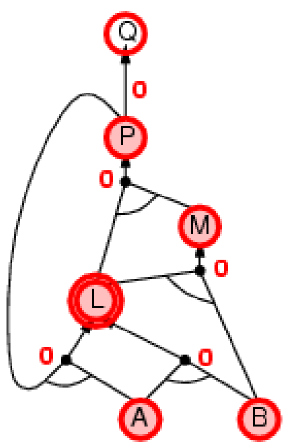
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

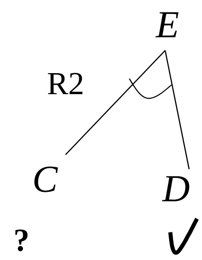
$A \wedge B \Rightarrow L$

A

B



Backward chaining example



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

R2: $C \wedge D \Rightarrow E$

R3: $C \wedge F \Rightarrow G$

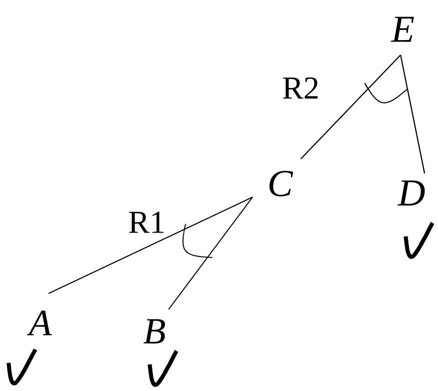
F1: A

F2: B

F3: D

- Backward chaining is more focused:
  - tries to prove the theorem only

Backward chaining example



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

- Backward chaining is more focused:
  - tries to prove the theorem only

Backward chaining

•

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

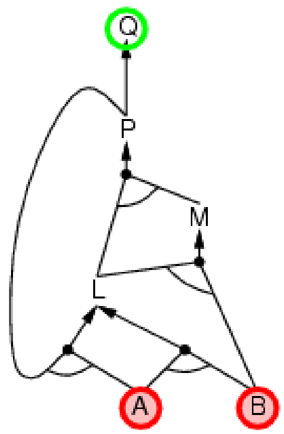
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



CS 2740 Knowledge Representation — M. Hauskrecht

Backward chaining

•

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

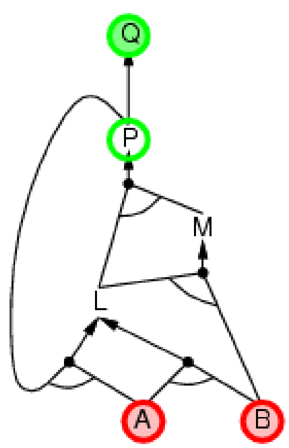
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



CS 2740 Knowledge Representation — M. Hauskrecht

Backward chaining

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

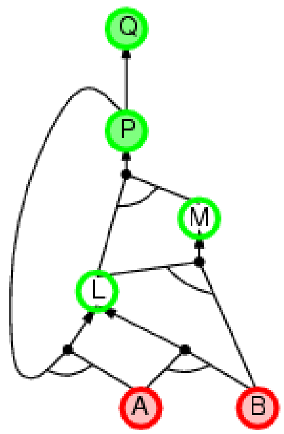
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Backward chaining

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

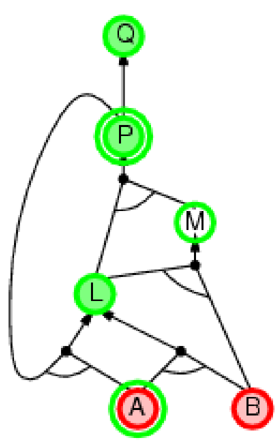
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Backward chaining

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

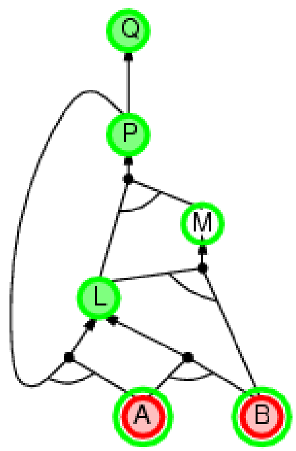
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Backward chaining

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

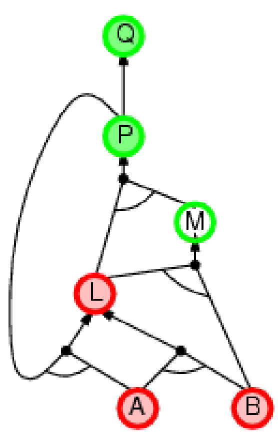
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Backward chaining

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

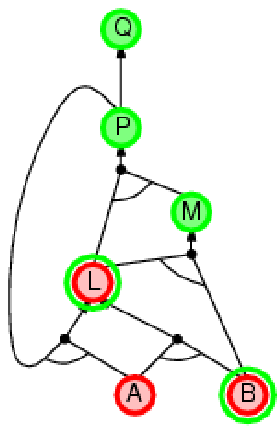
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Backward chaining

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

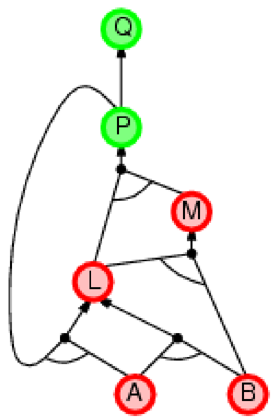
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Backward chaining

•

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

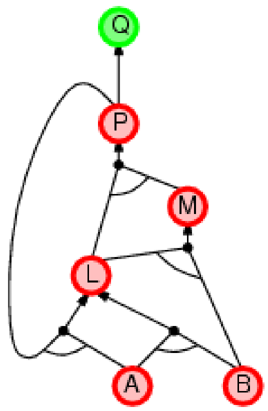
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



CS 2740 Knowledge Representation M. Hauskrecht

Backward chaining

•

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

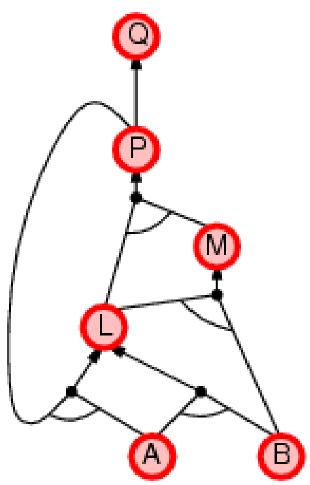
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



CS 2740 Knowledge Representation M. Hauskrecht

Forward vs Backward chaining

- **FC is data-driven**, automatic, unconscious processing,
  - e.g., object recognition, routine decisions
- May do lots of work that is irrelevant to the goal
- **BC is goal-driven**, appropriate for problem-solving,
  - e.g., Where are my keys? How do I get into a PhD program?
- Complexity of BC can be **much less** than **linear in size of KB**

KB agents based on propositional logic

- Propositional logic allows us to build **knowledge-based agents** capable of answering queries about the world by inferring new facts from the known ones
- **Example:** an agent for diagnosis of a bacterial disease

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

First-order logic

CS 1571 Intro to AI M. Hauskrecht

Limitations of propositional logic

World we want to represent and reason about consists of a number of **objects** with variety of **properties** and **relations** among them

Propositional logic:

- Represents statements about the world without reflecting this structure and without modeling these entities explicitly

Consequence:

- some knowledge is hard or impossible to encode in the propositional logic.
- Two cases that are hard to represent:
  - **Statements about similar objects, or relations**
  - **Statements referring to groups of objects**.

CS 1571 Intro to AI M. Hauskrecht

Limitations of propositional logic

- **Statements about similar objects and relations needs to be enumerated**
- **Example:** Seniority of people domain

  For inferences we need:

  John is older than Mary $\wedge$ *Mary is older than Paul*
  $\Rightarrow$ *John is older than Paul*

  Jane is older than Mary $\wedge$ *Mary is older than Paul*
  $\Rightarrow$ *Jane is older than Paul*

- **Problem:** if we have many people and their age relations we need to represent many rules to support the inferences
- **Possible solution: ??**

Limitations of propositional logic

- **Statements about similar objects and relations needs to be enumerated**
- **Example:** Seniority of people domain

  For inferences we need:

  John is older than Mary $\wedge$ *Mary is older than Paul*
  $\Rightarrow$ *John is older than Paul*

  Jane is older than Mary $\wedge$ *Mary is older than Paul*
  $\Rightarrow$ *Jane is older than Paul*

- **Problem:** if we have many people and their age relations we need to represent many rules to support the inferences
- **Possible solution: introduce variables**

  PersA *is older than* ***PersB*** $\wedge$ ***PersB*** *is older than* ***PersC***
  $\Rightarrow$ ***PersA*** *is older than* ***PersC***

Limitations of propositional logic

- **Statements referring to groups of objects require exhaustive enumeration of objects**
- **Example:**

Assume we want to express *Every student likes vacation*

Doing this in propositional logic would require to include statements about every student

John likes vacation $\wedge$
Mary likes vacation $\wedge$
Ann likes vacation $\wedge$
…

- **Solution:** Allow quantification in statements

First-order logic (FOL)

- More expressive than **propositional logic**
- **Eliminates deficiencies of PL by:**
  - Representing objects, their properties, relations and statements about them;
  - Introducing variables that refer to an arbitrary objects and can be substituted by a specific object
  - Introducing quantifiers allowing us to make statements over groups objects without the need to represent each of them separately