Methods of Proof

Chapter 7, second half.

Proof methods

- Proof methods divide into (roughly) two kinds:

  Application of inference rules:

  Legitimate (sound) generation of new sentences from old.

  - Resolution
  - Forward & Backward chaining

  Model checking

  Searching through truth assignments.

    - Improved backtracking: Davis--Putnam-Logemann-Loveland (DPLL)
    - Heuristic search in model space: Walksat.

Normal Form

We like to prove:

$$\boxed{\begin{array}{l} KB \models \alpha \\ \textit{equivalent to}: KB \land \neg\alpha \textit{ unsatisfiable} \end{array}}$$

We first rewrite $KB \land \neg\alpha$ into conjunctive normal form (CNF).

A "conjunction of disjunctions"

literals

$$\underbrace{(A \lor \neg B)}_{\text{Clause}} \land \underbrace{(B \lor \neg C \lor \neg D)}_{\text{Clause}}$$

- Any KB can be converted into CNF.
- In fact, any KB can be converted into CNF-3 using clauses with at most 3 literals.

Example: Conversion to CNF

$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
   $(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$

2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$

3. Move $\neg$ inwards using de Morgan's rules and double-negation: $\neg(\alpha \vee \beta) = \neg\alpha \wedge \neg\beta$
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge ((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1})$

4. Apply distributive law ($\wedge$ over $\vee$) and flatten:
   $(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$

Resolution

- Resolution: inference rule for CNF: sound and complete!

$(A \vee B \vee C)$

$(\neg A)$

"If A or B or C is true, but not A, then B or C must be true."

$- - - - - - - - - - - -$

$\therefore (B \vee C)$

$(A \vee B \vee C)$

$(\neg A \vee D \vee E)$

"If A is false then B or C must be true, or if A is true then D or E must be true, hence since A is either true or false, B or C or D or E must be true."

$- - - - - - - - - - -$

$\therefore (B \vee C \vee D \vee E)$

$(A \vee B)$

$(\neg A \vee B)$

$- - - - - - - -$

Simplification

$\therefore (B \vee B) \equiv B$

Resolution Algorithm

- The resolution algorithm tries to prove: $KB \models \alpha$ *equivalent to* $KB \wedge \neg\alpha$ *unsatisfiable*

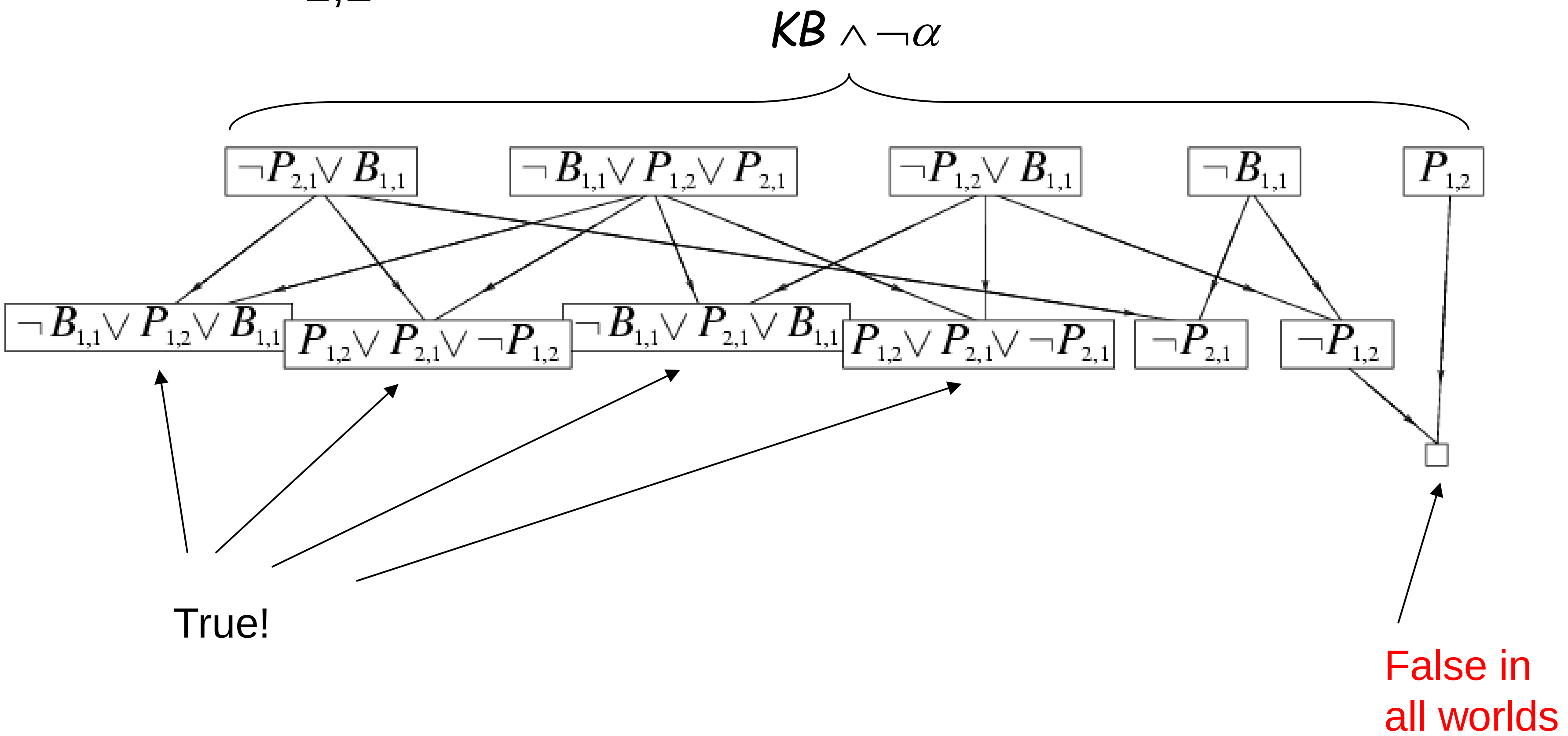
- Generate all new sentences from KB and the query.
- One of two things can happen:

1. We find $P \wedge \neg P$ which is unsatisfiable. I.e. we can entail the query.

2. We find no contradiction: there is a model that satisfies the sentence $KB \wedge \neg\alpha$ (non-trivial) and hence we cannot entail the query.

Resolution example

- $KB = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$
- $\alpha = \neg P_{1,2}$



Horn Clauses

- Resolution can be exponential in space and time.

- If we can reduce all clauses to "Horn clauses" resolution is linear in space and time

A clause with at most 1 positive literal.
e.g. $A \vee \neg B \vee \neg C$

- Every Horn clause can be rewritten as an implication with a conjunction of positive literals in the premises and a single positive literal as a conclusion.

e.g. $B \wedge C \Rightarrow A$

- 1 positive literal: definite clause
- 0 positive literals: Fact or integrity constraint:
  e.g. $(\neg A \vee \neg B) \equiv (A \wedge B \Rightarrow False)$
- Forward Chaining and Backward chaining are sound and complete with Horn clauses and run linear in space and time.

Try it Yourselves

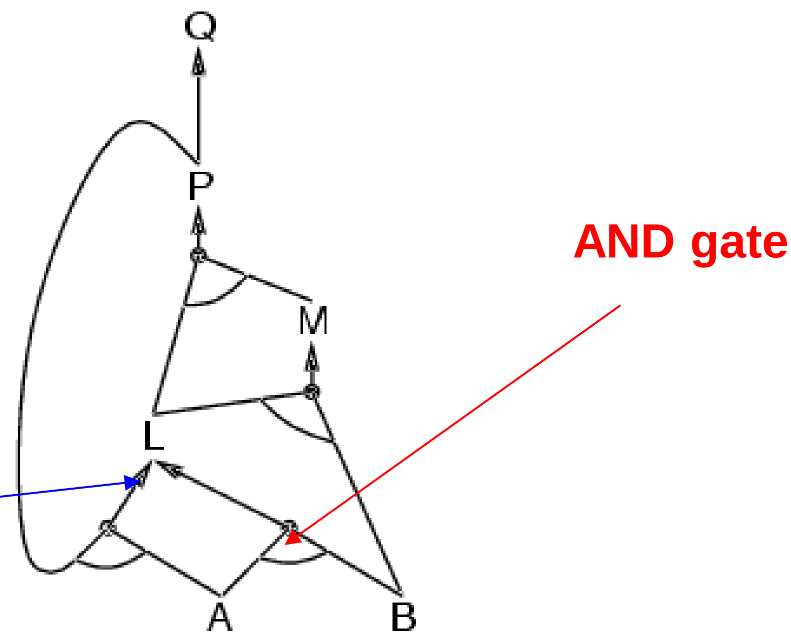
- 7.9 page 238: (Adapted from Barwise and Etchemendy (1993).) If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

- *Derive the KB in normal form.*
- *Prove: Horned, Prove: Magical.*

Forward chaining

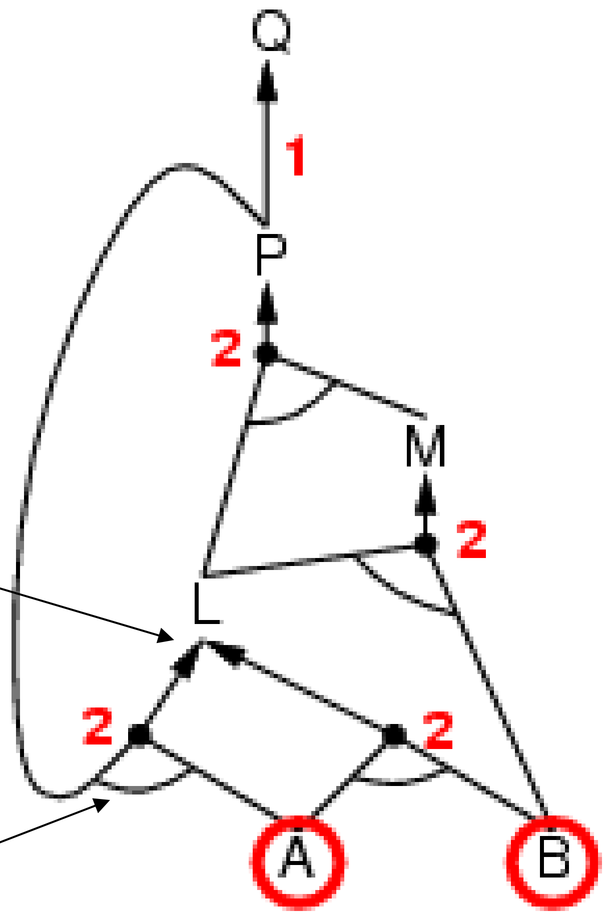
- Idea: fire any rule whose premises are satisfied in the *KB*,
  - add its conclusion to the *KB*, until query is found

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



- Forward chaining is sound and complete for Horn KB

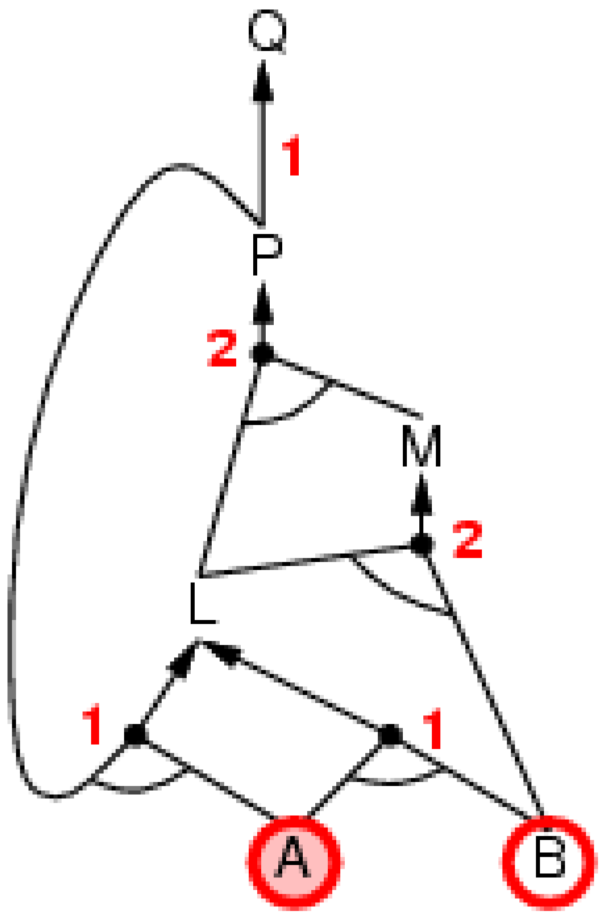
Forward chaining example



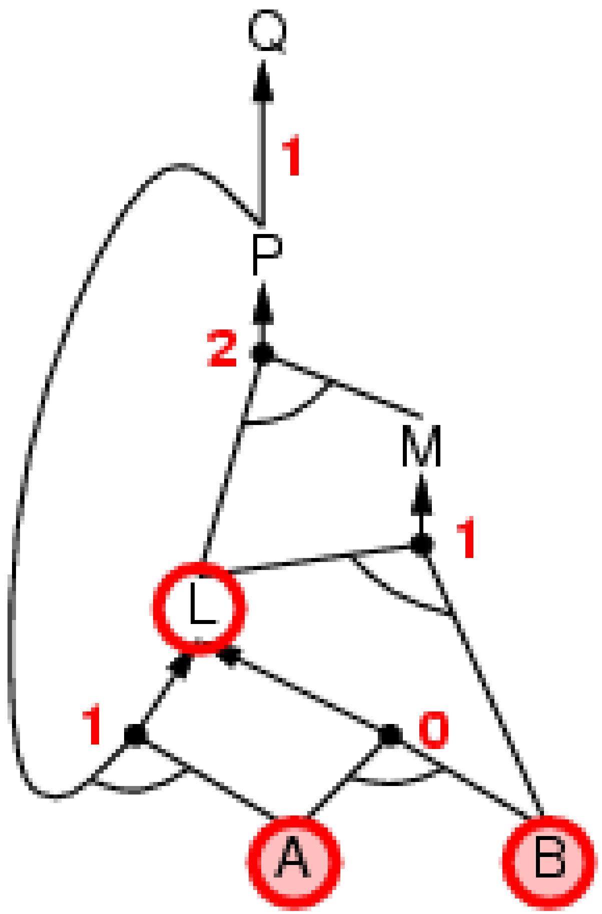
“OR” Gate

“AND” gate

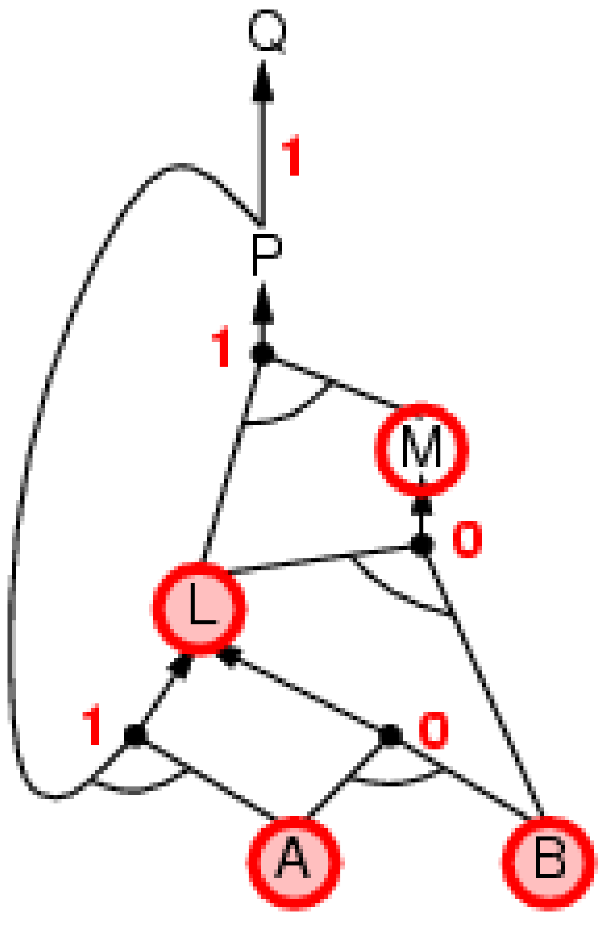
Forward chaining example



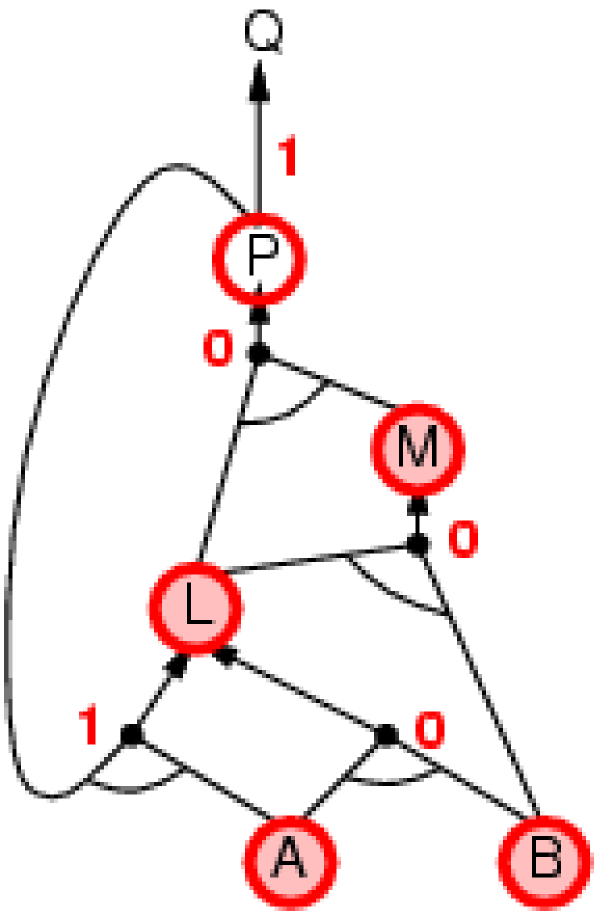
Forward chaining example



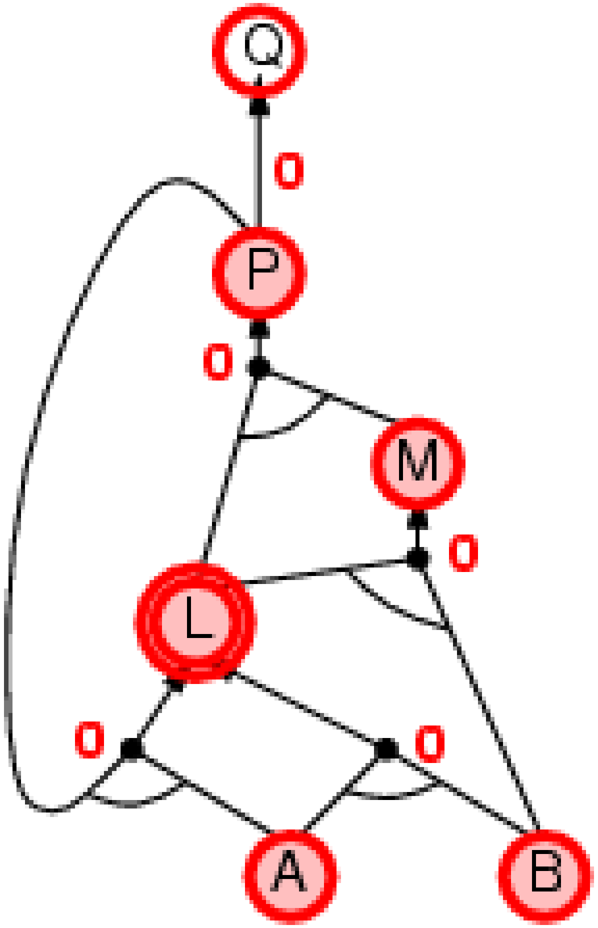
Forward chaining example



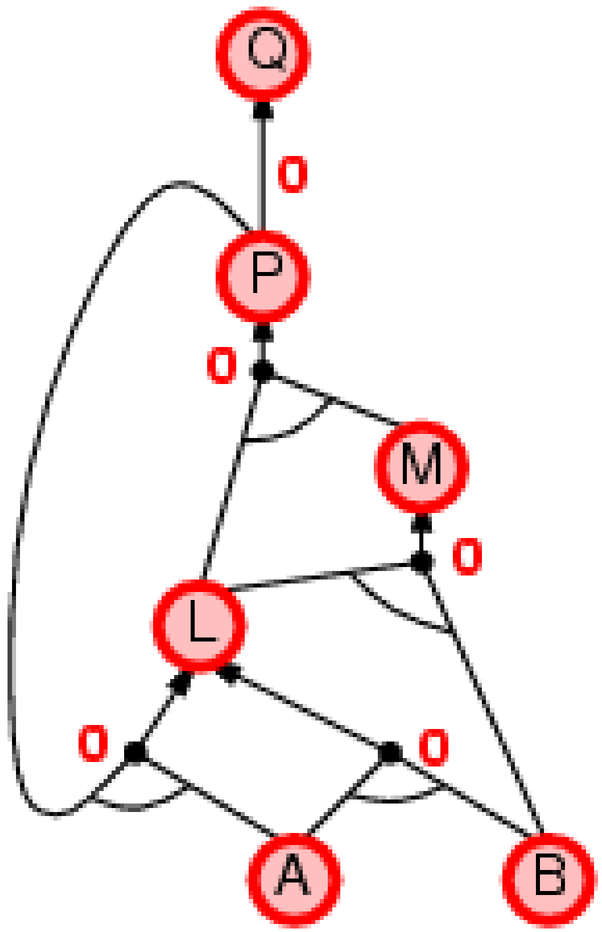
Forward chaining example



Forward chaining example



Forward chaining example



Backward chaining

Idea: work backwards from the query *q*

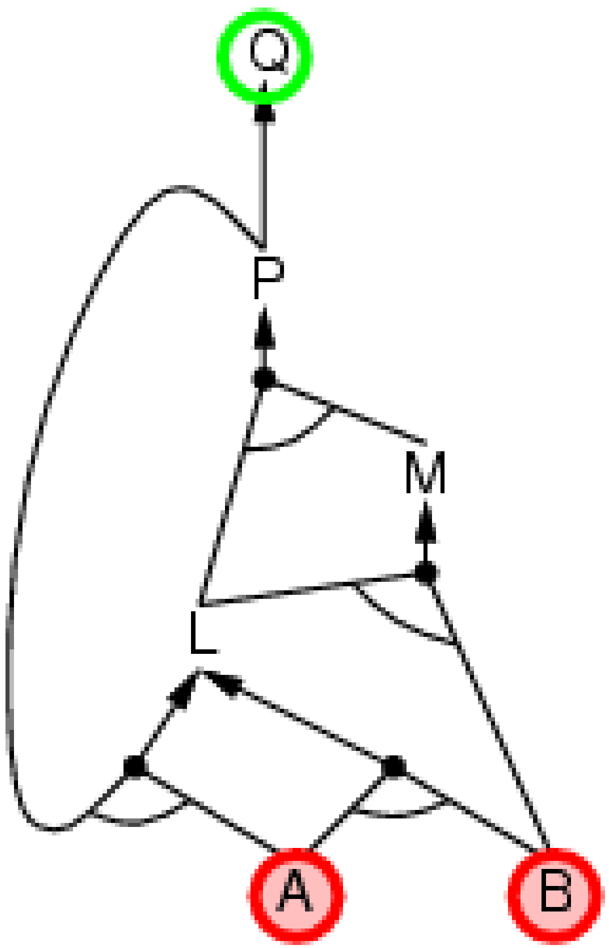
- check if *q* is known already, or
- prove by BC all premises of some rule concluding *q*
- Hence BC maintains a stack of sub-goals that need to be proved to get to q.

Avoid loops: check if new sub-goal is already on the goal stack

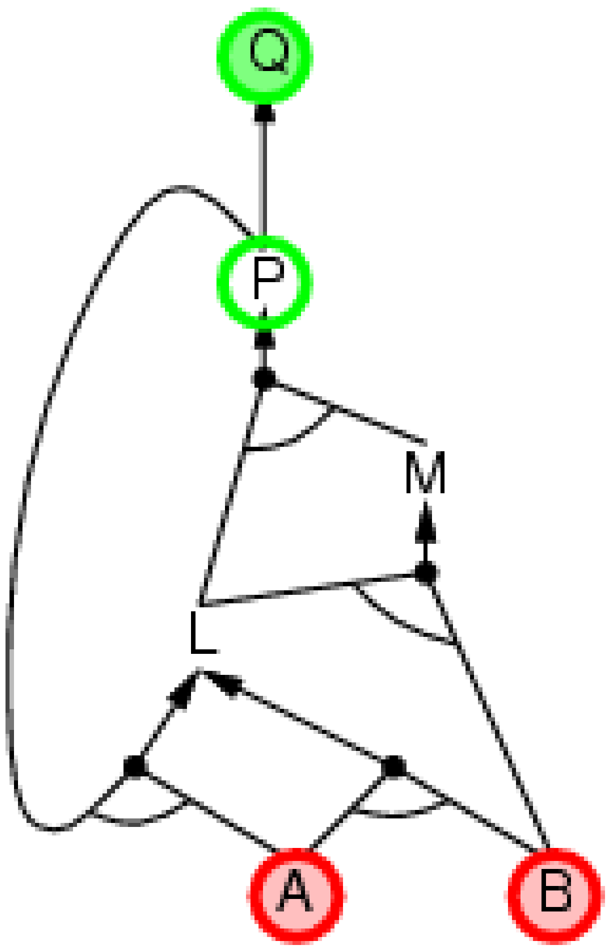
Avoid repeated work: check if new sub-goal

1. has already been proved true, or
2. has already failed

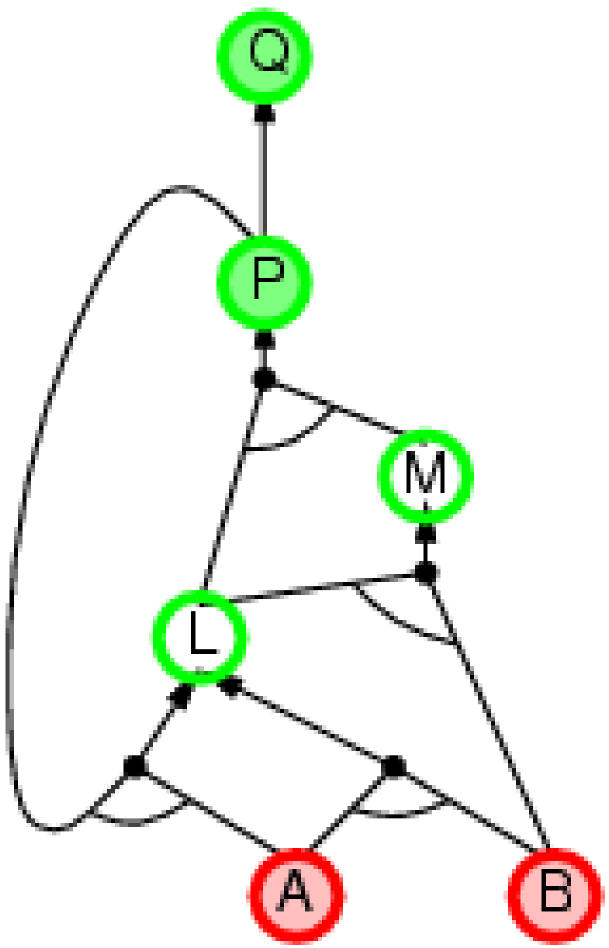
Backward chaining example



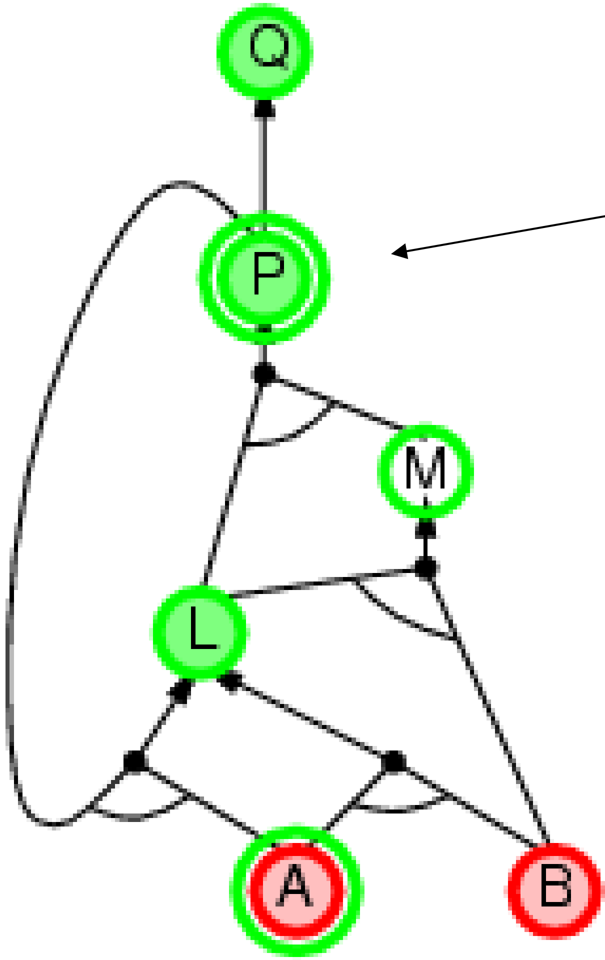
Backward chaining example



Backward chaining example

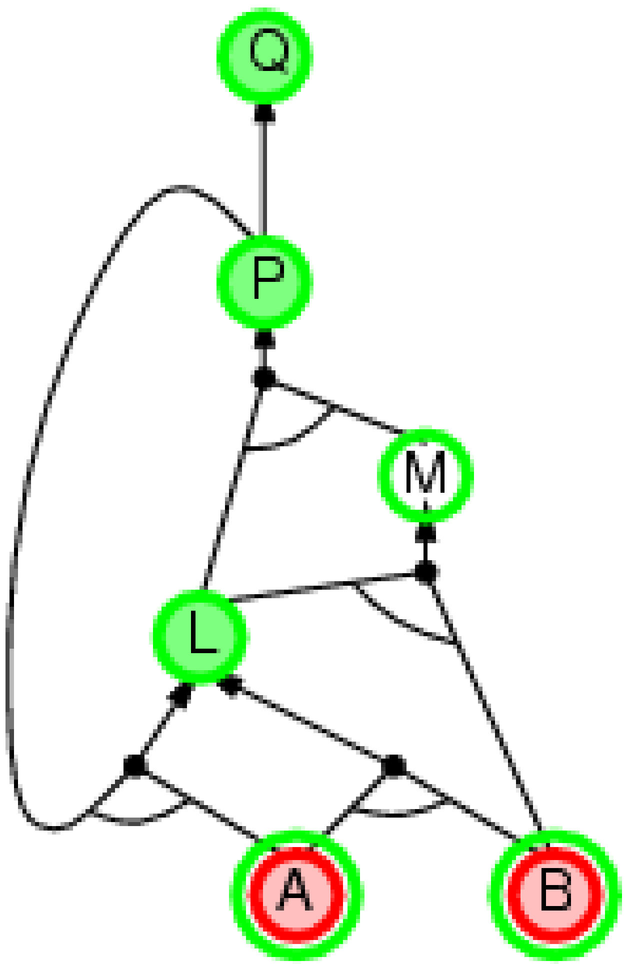


Backward chaining example

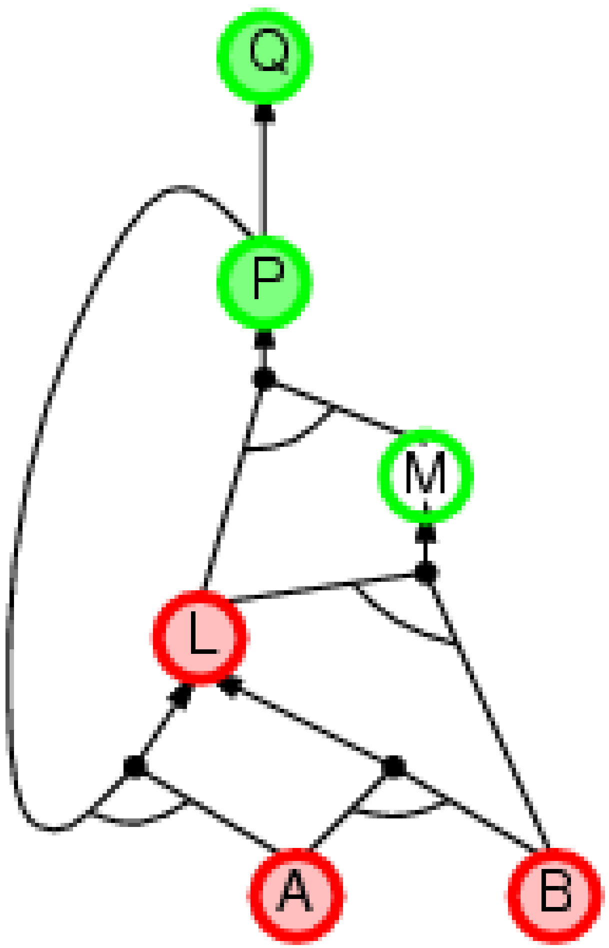


we need P to prove L and L to prove P.

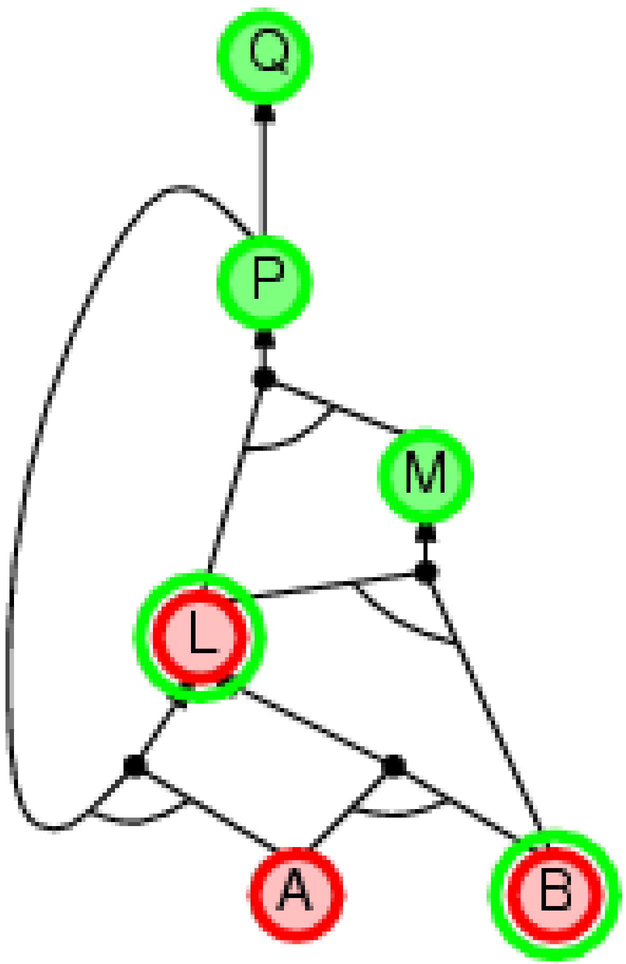
Backward chaining example



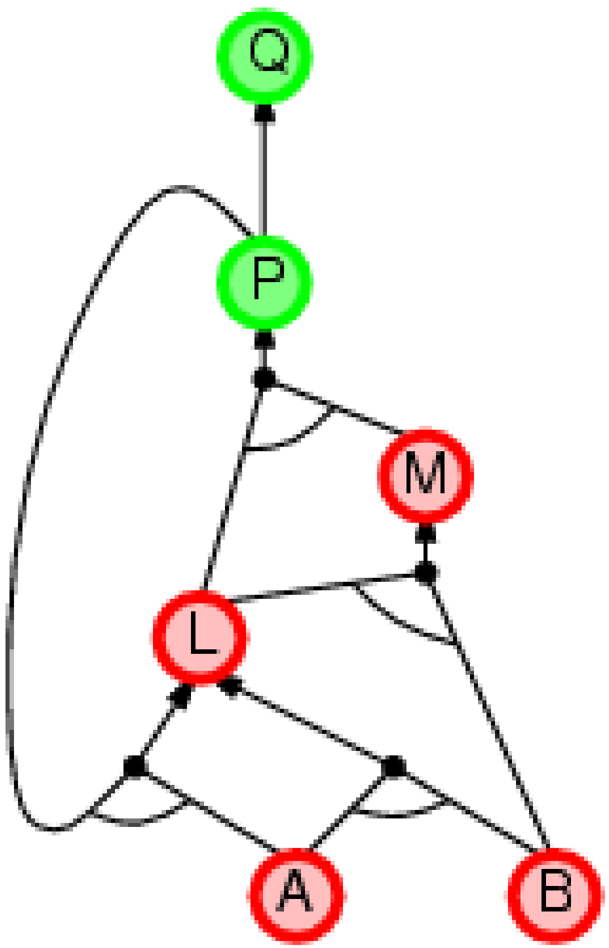
Backward chaining example



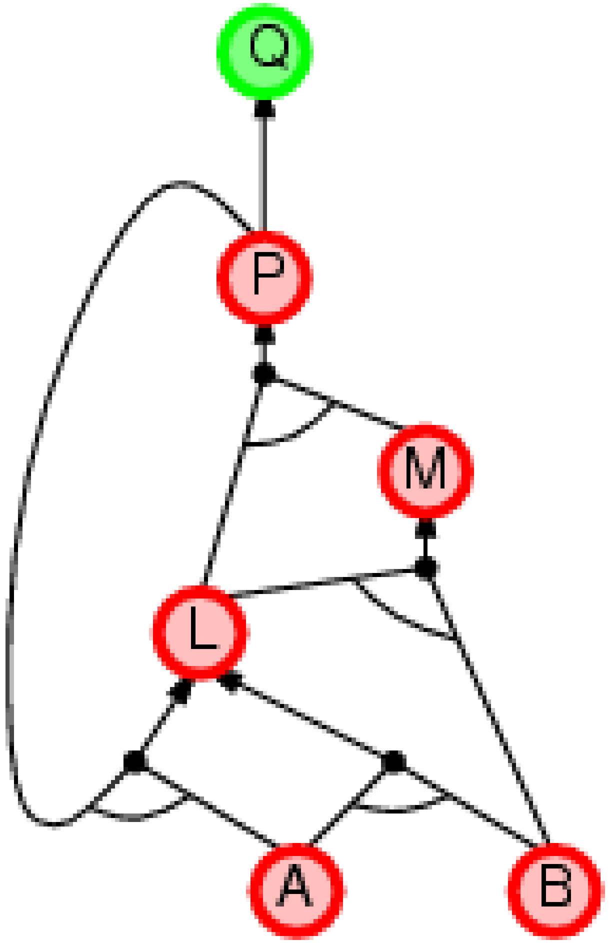
Backward chaining example



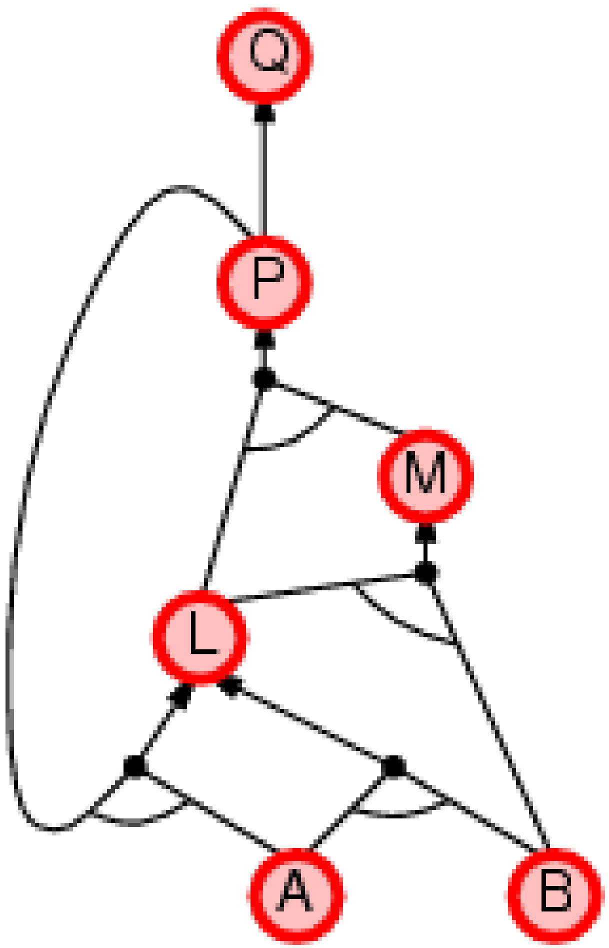
Backward chaining example



Backward chaining example



Backward chaining example



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

Model Checking

Two families of efficient algorithms:

- Complete backtracking search algorithms: DPLL algorithm
- Incomplete local search algorithms
  - `WalkSAT` algorithm

The DPLL algorithm

Determine if an input propositional logic sentence (in CNF) is satisfiable. This is just backtracking search for a CSP.

Improvements:

1. Early termination

   A clause is true if any literal is true.

   A sentence is false if any clause is false.

2. Pure symbol heuristic

   Pure symbol: always appears with the same "sign" in all clauses.

   e.g., In the three clauses $(A \lor \neg B)$, $(\neg B \lor \neg C)$, $(C \lor A)$, A and B are pure, C is impure.

   Make a pure symbol literal true. (if there is a model for S, then making a pure symbol true is also a model).

3. Unit clause heuristic

   Unit clause: only one literal in the clause

   The only literal in a unit clause must be true.

   Note: literals can become a pure symbol or a unit clause when other literals obtain truth values. e.g.

$$(A \lor \cancel{True}) \land (\neg A \lor B)$$

$$A = pure$$

The `WalkSAT` algorithm

- Incomplete, local search algorithm
- Evaluation function: The min-conflict heuristic of minimizing the number of unsatisfied clauses
- Balance between greediness and randomness

Hard satisfiability problems

- Consider *random* 3-CNF sentences. e.g.,

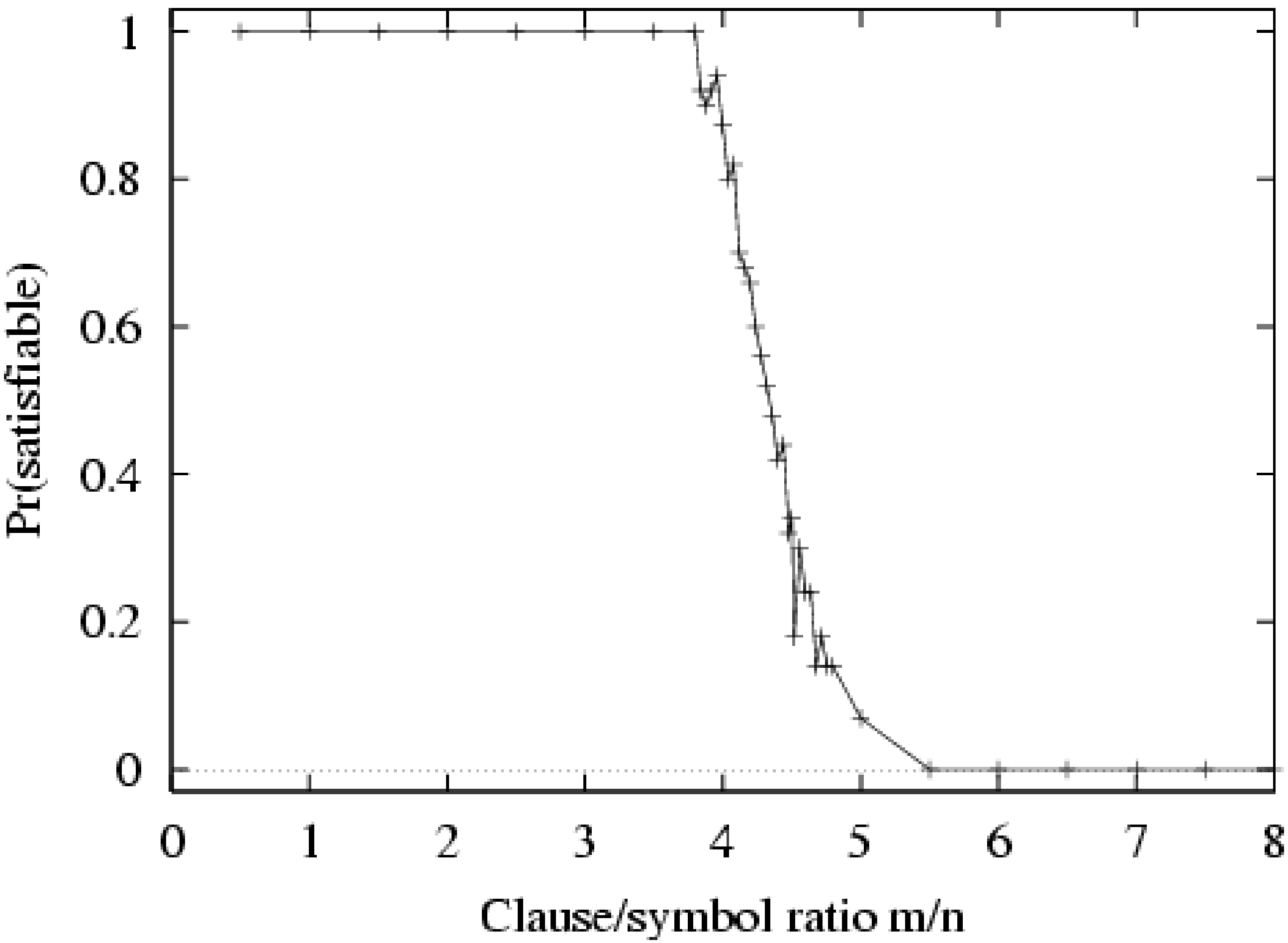
  $(\neg D \vee \neg B \vee C) \wedge (B \vee \neg A \vee \neg C) \wedge (\neg C \vee \neg B \vee E) \wedge (E \vee \neg D \vee B) \wedge (B \vee E \vee \neg C)$

  m = number of clauses (5)

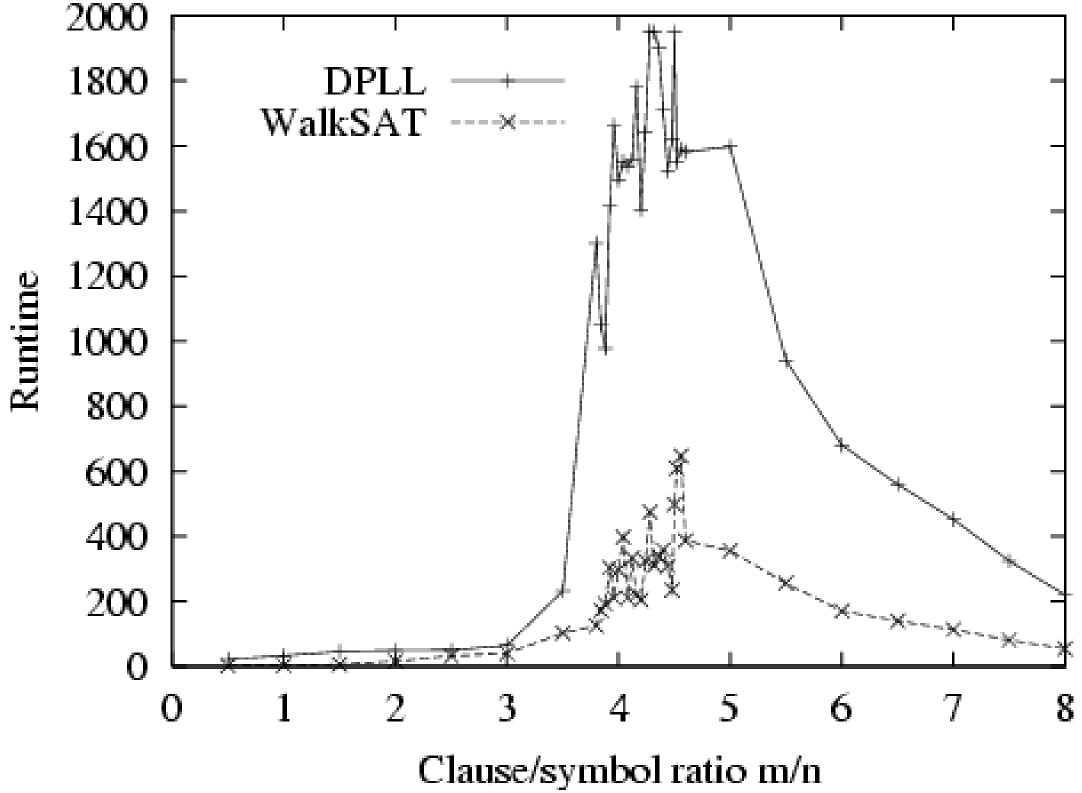
  n = number of symbols (5)

  – Hard problems seem to cluster near *m/n* = 4.3 (critical point)

Hard satisfiability problems



Hard satisfiability problems



- Median runtime for 100 satisfiable random 3-CNF sentences, $n = 50$

Inference-based agents in the wumpus world

A wumpus-world agent using propositional logic:

$\neg P_{1,1}$ (no pit in square [1,1])
$\neg W_{1,1}$ (no Wumpus in square [1,1])
$B_{x,y} \Leftrightarrow (P_{x,y+1} \vee P_{x,y-1} \vee P_{x+1,y} \vee P_{x-1,y})$ (Breeze next to Pit)
$S_{x,y} \Leftrightarrow (W_{x,y+1} \vee W_{x,y-1} \vee W_{x+1,y} \vee W_{x-1,y})$ (stench next to Wumpus)
$W_{1,1} \vee W_{1,2} \vee \ldots \vee W_{4,4}$ (at least 1 Wumpus)

$$\left.\begin{array}{l}\neg W_{1,1} \vee \neg W_{1,2} \\ \neg W_{1,1} \vee \neg W_{8,9}\end{array}\right\}$$ (at most 1 Wumpus)

…

$\Rightarrow$ 64 distinct proposition symbols, 155 sentences

Expressiveness limitation of propositional logic

- KB contains "physics" sentences for every single square

- For every time *t* and every location [*x,y*],
  $L^{t}_{x,y} \wedge FacingRight^{t} \wedge Forward^{t} \Rightarrow L^{t+1}_{x+1,y}$

  position (x,y) at time t of the agent.

- Rapid proliferation of clauses.
  First order logic is designed to deal with this through the introduction of variables.

2.(12 pts) **Inference**

Consider the following logical sentence in CNF form:

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

a.(2 pts) Is this sentence in Horn form?
Are all clauses definite clauses?

b.(4 pts) Use either resolution or forward chaining to show that this sentence is unsatisfiable. In other words, there is no assignment of False/True values for the variables A,B,C that will render the sentence True.

c.(2 pts) Next consider the following sentence:
$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee B) \wedge (\neg A \vee \neg B \vee \neg C) \wedge (A \vee B) \wedge (A \vee \neg C)$.
Is this sentence in Horn form?

d.(4 pts) We will solve this problem using backtracking search. We use the following heuristics: 1) order the variables according to the number of clauses they are involved in (the variable involved in the largest number is first), 2) choose a value for a variable that satisfies the largest number of clauses.
Provide a solution (i.e. an assignment of values True/False to the variables such that all clauses are satisfied).

2.(12 pts) **Inference**

Consider the following logical sentence in CNF form:
$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee B) \wedge (A) \wedge (\neg A \vee \neg B \vee \neg C)$.

a.(2 pts) Is this sentence in Horn form?
Are all clauses definite clauses?

a) answer: Yes, No (last clause).

b.(4 pts) Use either resolution or forward chaining to show that this sentence is unsatisfiable. In other words, there is no assignment of False/True values for the variables A,B,C that will render the sentence True.

c.(2 pts) Next consider the following sentence:
$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee B) \wedge (\neg A \vee \neg B \vee \neg C) \wedge (A \vee B) \wedge (A \vee \neg C)$.
Is this sentence in Horn form?

d.(4 pts) We will solve this problem using backtracking search. We use the following heuristics: 1) order the variables according to the number of clauses they are involved in (the variable involved in the largest number is first), 2) choose a value for a variable that satisfies the largest number of clauses.
Provide a solution (i.e. an assignment of values True/False to the variables such that all clauses are satisfied).

2.(12 pts) **Inference**

Consider the following logical sentence in CNF form:
$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee B) \wedge (A) \wedge (\neg A \vee \neg B \vee \neg C)$.

a.(2 pts) Is this sentence in Horn form?
Are all clauses definite clauses?

a) answer: Yes, No (last clause).

b.(4 pts) Use either resolution or forward chaining to show that this sentence is unsatisfiable. In other words, there is no assignment of False/True values for the variables A,B,C that will render the sentence True.

b) answer: FC: First rewrite al clauses into implications, and then derive A,B,C,False in that order. With resolution you use derive A,B,C in the same order and then show that you get an empty clause.

c.(2 pts) Next consider the following sentence:
$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee B) \wedge (\neg A \vee \neg B \vee \neg C) \wedge (A \vee B) \wedge (A \vee \neg C)$.
Is this sentence in Horn form?

d.(4 pts) We will solve this problem using backtracking search. We use the following heuristics: 1) order the variables according to the number of clauses they are involved in (the variable involved in the largest number is first), 2) choose a value for a variable that satisfies the largest number of clauses.
Provide a solution (i.e. an assignment of values True/False to the variables such that all clauses are satisfied).

2.(12 pts) **Inference**

Consider the following logical sentence in CNF form:
$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee B) \wedge (A) \wedge (\neg A \vee \neg B \vee \neg C)$.

a.(2 pts) Is this sentence in Horn form?
Are all clauses definite clauses?

a) answer: Yes, No (last clause).

b.(4 pts) Use either resolution or forward chaining to show that this sentence is unsatisfiable. In other words, there is no assignment of False/True values for the variables A,B,C that will render the sentence True.

b) answer: FC: First rewrite al clauses into implications, and then derive A,B,C,False in that order. With resolution you use derive A,B,C in the same order and then show that you get an empty clause.

c.(2 pts) Next consider the following sentence:
$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee B) \wedge (\neg A \vee \neg B \vee \neg C) \wedge (A \vee B) \wedge (A \vee \neg C)$.
Is this sentence in Horn form?

c) answer: No (fourth clause).

d.(4 pts) We will solve this problem using backtracking search. We use the following heuristics: 1) order the variables according to the number of clauses they are involved in (the variable involved in the largest number is first), 2) choose a value for a variable that satisfies the largest number of clauses.
Provide a solution (i.e. an assignment of values True/False to the variables such that all clauses are satisfied).

2.(12 pts) **Inference**

Consider the following logical sentence in CNF form:
$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee B) \wedge (A) \wedge (\neg A \vee \neg B \vee \neg C)$.

a.(2 pts) Is this sentence in Horn form?
Are all clauses definite clauses?

a) answer: Yes, No (last clause).

b.(4 pts) Use either resolution or forward chaining to show that this sentence is unsatisfiable. In other words, there is no assignment of False/True values for the variables A,B,C that will render the sentence True.

b) answer: FC: First rewrite al clauses into implications, and then derive A,B,C,False in that order. With resolution you use derive A,B,C in the same order and then show that you get an empty clause.

c.(2 pts) Next consider the following sentence:
$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee B) \wedge (\neg A \vee \neg B \vee \neg C) \wedge (A \vee B) \wedge (A \vee \neg C)$.
Is this sentence in Horn form?

c) answer: No (fourth clause).

d.(4 pts) We will solve this problem using backtracking search. We use the following heuristics: 1) order the variables according to the number of clauses they are involved in (the variable involved in the largest number is first), 2) choose a value for a variable that satisfies the largest number of clauses.
Provide a solution (i.e. an assignment of values True/False to the variables such that all clauses are satisfied).

d) answer: A=F, B=T, C=F (no backtracking needed).

2.(10 pts) **Propositional Logic**
Consider the following knowledge base: $KB_0 = \neg A \vee \neg B \vee C$ and the sentence $\alpha = \neg A \vee \neg B$

a.(2 pts) Use De Morgan's law to rewrite $\neg\alpha = \neg(\neg A \vee \neg B)$

b.(2 pts) Consider the updated KB: $KB_1 = KB_0 \wedge \neg\alpha$. Is the KB_1 in Horn form? Are all clauses in KB_1 definite clauses?

c.(4 pts) Use resolution to prove $C = true$.

d.(2 pts) Is α entailed by KB_0, i.e. $KB_0 \models \alpha$?

2. (10 pts) **Propositional Logic**
Consider the following knowledge base: $KB_0 = \neg A \vee \neg B \vee C$ and the sentence $\alpha = \neg A \vee \neg B$

a. (2 pts) Use De Morgan's law to rewrite $\neg\alpha = \neg(\neg A \vee \neg B)$

a) answer: $\neg(\neg A \vee \neg B) = A \wedge B$.

b. (2 pts) Consider the updated KB: $KB_1 = KB_0 \wedge \neg\alpha$. Is the KB_1 in Horn form? Are all clauses in KB_1 definite clauses?

c. (4 pts) Use resolution to prove $C = true$.

d. (2 pts) Is α entailed by KB_0, i.e. $KB_0 \models \alpha$?

2.(10 pts) **Propositional Logic**

Consider the following knowledge base: $KB_0 = \neg A \vee \neg B \vee C$ and the sentence $\alpha = \neg A \vee \neg B$

a.(2 pts) Use De Morgan's law to rewrite $\neg\alpha = \neg(\neg A \vee \neg B)$

a) answer: $\neg(\neg A \vee \neg B) = A \wedge B$.

b.(2 pts) Consider the updated KB: $KB_1 = KB_0 \wedge \neg\alpha$. Is the KB_1 in Horn form? Are all clauses in KB_1 definite clauses?

b) answer: Yes and yes. There are 3 clauses, each clause contains exactly 1 positive literal.

c.(4 pts) Use resolution to prove $C = true$.

d.(2 pts) Is α entailed by KB_0, i.e. $KB_0 \models \alpha$?

2.(10 pts) **Propositional Logic**

Consider the following knowledge base: $KB_0 = \neg A \vee \neg B \vee C$ and the sentence $\alpha = \neg A \vee \neg B$

a.(2 pts) Use De Morgan's law to rewrite $\neg\alpha = \neg(\neg A \vee \neg B)$

a) answer: $\neg(\neg A \vee \neg B) = A \wedge B$.

b.(2 pts) Consider the updated KB: $KB_1 = KB_0 \wedge \neg\alpha$. Is the KB_1 in Horn form? Are all clauses in KB_1 definite clauses?

b) answer: Yes and yes. There are 3 clauses, each clause contains exactly 1 positive literal.

c.(4 pts) Use resolution to prove $C = true$.

c) answer: First Use A and $\neg A \vee \neg B \vee C$ to conclude $\neg B \vee C$. Then use that and B to conclude C.

d.(2 pts) Is α entailed by KB_0, i.e. $KB_0 \models \alpha$?

2.(10 pts) **Propositional Logic**
Consider the following knowledge base: $KB_0 = \neg A \vee \neg B \vee C$ and the sentence $\alpha = \neg A \vee \neg B$

a.(2 pts) Use De Morgan's law to rewrite $\neg\alpha = \neg(\neg A \vee \neg B)$

a) answer: $\neg(\neg A \vee \neg B) = A \wedge B$.

b.(2 pts) Consider the updated KB: $KB_1 = KB_0 \wedge \neg\alpha$. Is the KB_1 in Horn form? Are all clauses in KB_1 definite clauses?

b) answer: Yes and yes. There are 3 clauses, each clause contains exactly 1 positive literal.

c.(4 pts) Use resolution to prove $C = true$.

c) answer: First Use A and $\neg A \vee \neg B \vee C$ to conclude $\neg B \vee C$. Then use that and B to conclude C.

d.(2 pts) Is α entailed by KB_0, i.e. $KB_0 \models \alpha$?

d) answer: No: we have just shown that $KB_1 = KB_0 \wedge \neg\alpha$ has a solution and is therefore not unsatisfiable.

Summary

- Logical agents apply inference to a knowledge base to derive new information and make decisions
- Basic concepts of logic:
  - syntax: formal structure of sentences
  - semantics: truth of sentences wrt models
  - entailment: necessary truth of one sentence given another
  - inference: deriving sentences from other sentences
  - soundness: derivations produce only entailed sentences
  - completeness: derivations can produce all entailed sentences
- Wumpus world requires the ability to represent partial and negated information, reason by cases, etc.
- Resolution is complete for propositional logic
  Forward, backward chaining are linear-time, complete for Horn clauses
- Propositional logic lacks expressive power