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

Artificial Intelligence

Lecturer 9 – Propositional Logic

School of Information and Communication Technology - HUST

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, …

Propositional Logic

- Propositional Logic is about facts in the world that are either true or false, nothing else
- Propositional variables stand for basic facts
- Sentences are made of
  - propositional variables (A,B,…),
  - logical constants (TRUE, FALSE), and
  - logical connectives (not,and,or,..)
- The meaning of sentences ranges over the Boolean values {True, False}
  - Examples: It's sunny, John is married

Language of Propositional Logic

- Symbols
  - Propositional variables: A,B,…,P,Q,…
  - Logical constants: TRUE, FALSE
  - Logical connectives:

$$\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$$

- Sentences
  - Each propositional variable is a sentence
  - Each logical constant is a sentence
  - If α and β are sentences then the following are sentences

$$(\alpha), \neg\alpha, \alpha \wedge \beta, \alpha \vee \beta, \alpha \Rightarrow \beta, \alpha \Leftrightarrow \beta$$

Formal Language of Propositional Logic

- Formal Grammar
    - Sentence -> Asentence | Csentence
    - Asentence -> TRUE | FALSE | A | B|…
    - Csentence -> (Sentence) | Sentence | Sentence Connective Sentence
    - Connective -> $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

Semantic of Propositional Logic

- The meaning of TRUE is always True, the meaning of FALSE is always False
- The meaning of a propositional variable is either True or False
  - depends on the interpretation
    - assignment of Boolean values to propositional variables
- The meaning of a sentence is either True or False
  - depends on the interpretation

Semantic of Propositional Logic

- True table

P	Q	Not P	P and Q	P or Q	P implies Q	P equiv Q
False	False	True	False	False	True	True
False	True	True	False	True	True	False
True	False	False	False	True	False	False
True	True	False	True	True	True	True

$$a \Rightarrow b \Leftrightarrow \neg a \vee b \Leftrightarrow \neg b \Rightarrow \neg a$$

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

Γ ψ

Inference in Propositional Logic

- Forward Chaining
- Backward Chaining

Forward Chaining

- Given a set of rules, i.e. formulae of the form

$$p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q$$

and a set of known facts, i.e., formulae of the form

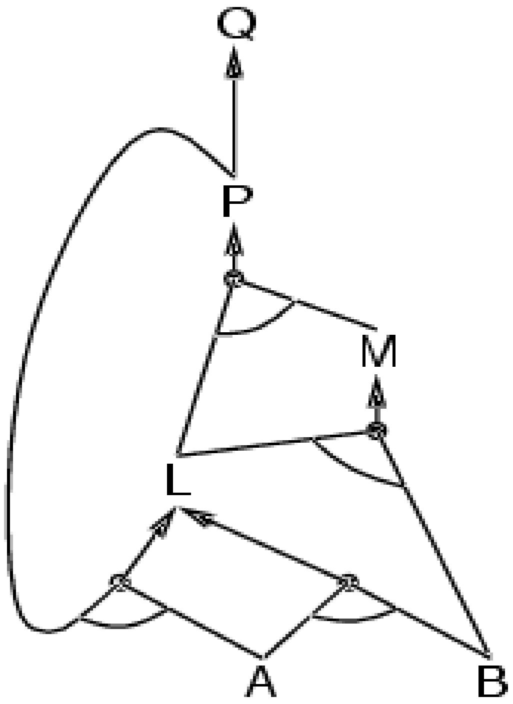
$$q, r, \ldots$$

- A new fact *p* is added
- Find all rules that have *p* as a premise
- If the other premises are already known to hold then
  - add the consequent to the set of know facts, and
  - trigger further inferences

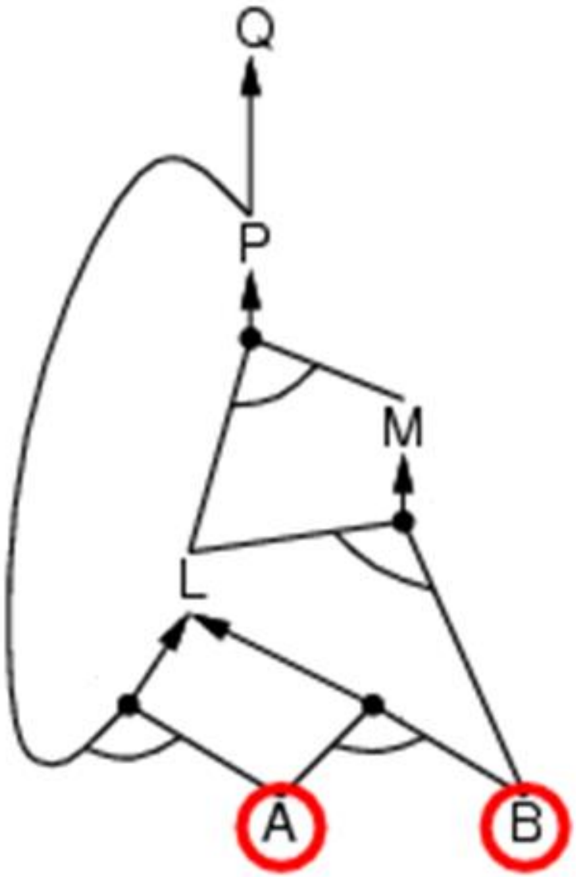
Forward Chaining

- Example

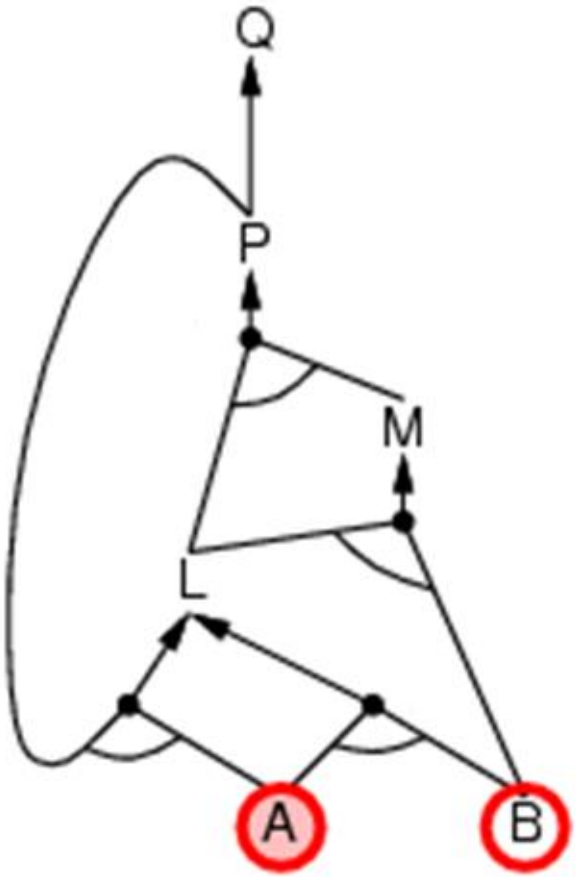
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



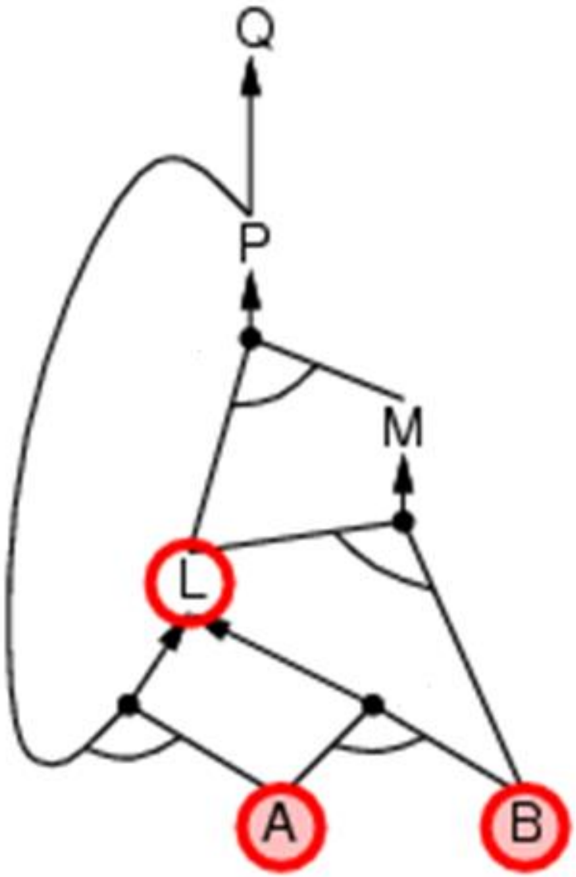
Forward Chaining



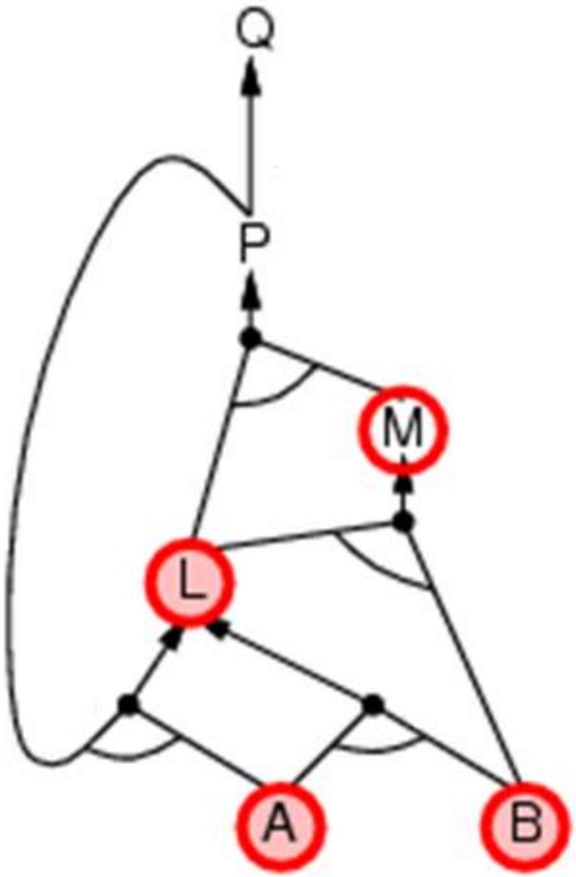
Forward Chaining



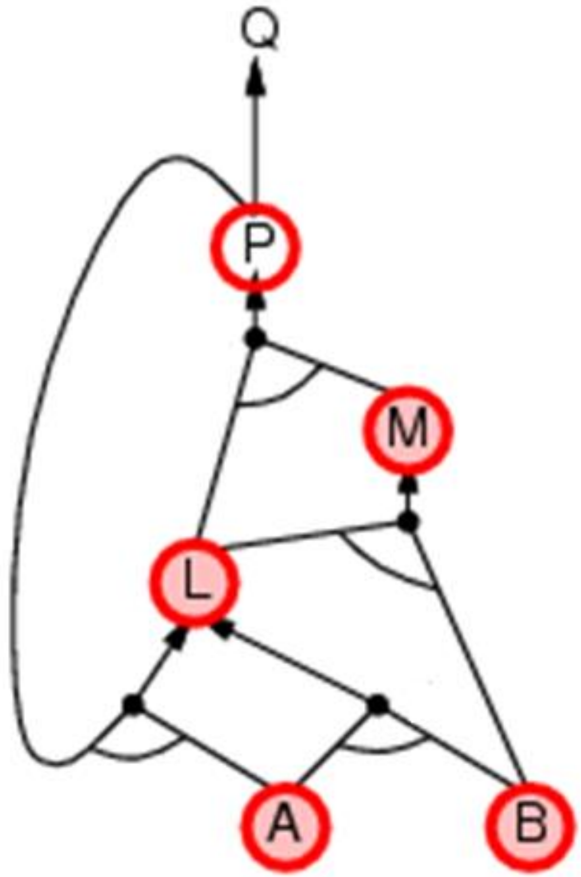
Forward Chaining



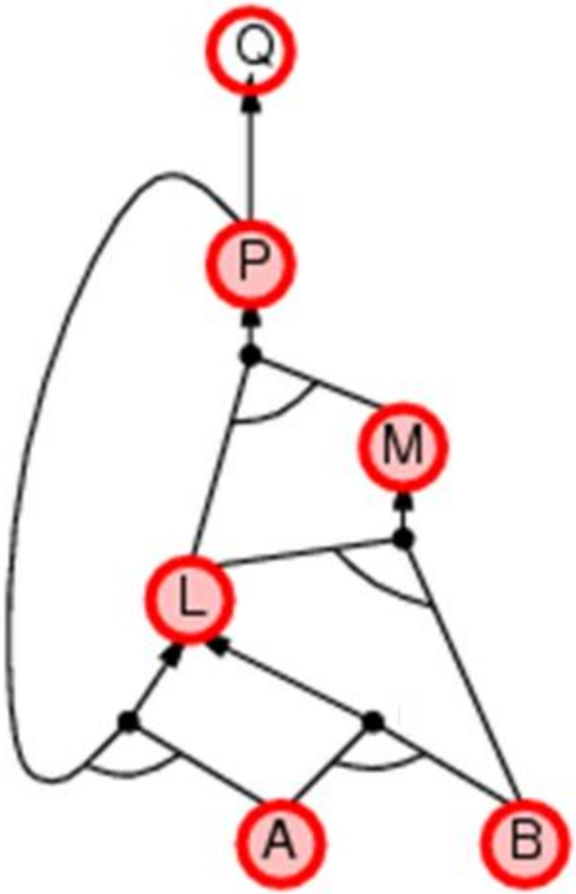
Forward Chaining



Forward Chaining



Forward Chaining



Example

E1. Given Fact = $\{a,b,m_a\}$. Prove h_c

1. $a,b,m_a \rightarrow c$
2. $a,b,c \rightarrow A$
3. $b,A \rightarrow h_c$
4. $a,b,c \rightarrow B$
5. $a,b,c \rightarrow C$
6. $a,B \rightarrow h_c$
7. $A,B \rightarrow C$
8. $B,C \rightarrow A$
9. $A,C \rightarrow B$

Forward Chaining

Input:

- Sentences/clauses in Horn format (Fact)
- A rule set R $p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q$
- Goal

Output:

- "Success" if Goal can be inferred from Fact

Method: Use

- Temp - a set of propositional variables which are true at the current time
- Sat - a set of satisfied rules

Forward Chaining

```
{1 Temp = Fact;
   Sat= FindRules(Temp,R);
   while Sat<>0 and Goal∉Temp do
   {2   r ← get(Sat); /* r: left → q  */
     R = R \ {r}; Trace = Trace ∪ {r};
     Temp = Temp ∪ {q};
     Sat = FindRules(Temp,R)
   }2
   if Goal ⊆ Temp then exit("Success")
   else   exit("Not success")
}1
```

Example

E1. Given Fact = $\{a,b,m_a\}$. Prove h_c

1. $a,b,m_a \rightarrow c$
2. $a,b,c \rightarrow A$
3. $b,A \rightarrow h_c$
4. $a,b,c \rightarrow B$
5. $a,b,c \rightarrow C$
6. $a,B \rightarrow h_c$
7. $A,B \rightarrow C$
8. $B,C \rightarrow A$
9. $A,C \rightarrow B$

Exercises

Compare stack and queue

1- Given Fact={a}, Goal={u}

1. $a \rightarrow b$
2. $b \rightarrow c$
3. $c \rightarrow d$
4. $a \rightarrow u$

2 - Given Fact={a}, Goal={u}

1. $a \rightarrow b$
2. $d \rightarrow c$
3. $c \rightarrow u$
4. $a \rightarrow m$
5. $b \rightarrow n$
6. $m \rightarrow p$
7. $p \rightarrow q$
8. $q \rightarrow u$

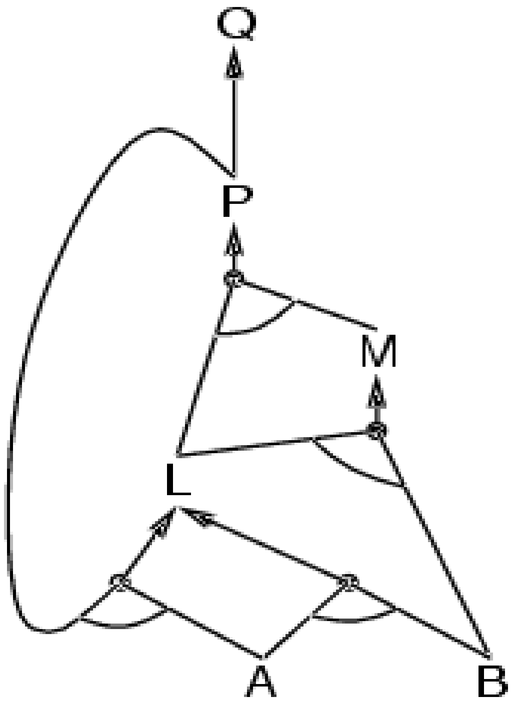
Backward Chaining

- Given a set of rules, and a set of known facts
- We ask whether a fact P is a consequence of the set of rules and the set of known facts
- The procedure check whether P is in the set of known facts
- Otherwise find all rules that have P as a consequent
  - If the premise is a conjunction, then process the conjunction conjunct by conjunct

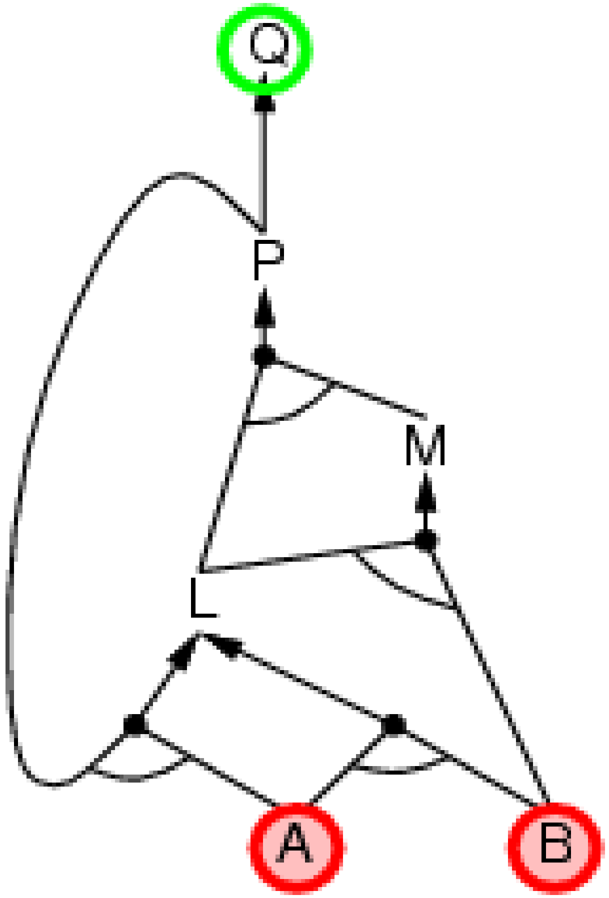
Backward Chaining

- Example

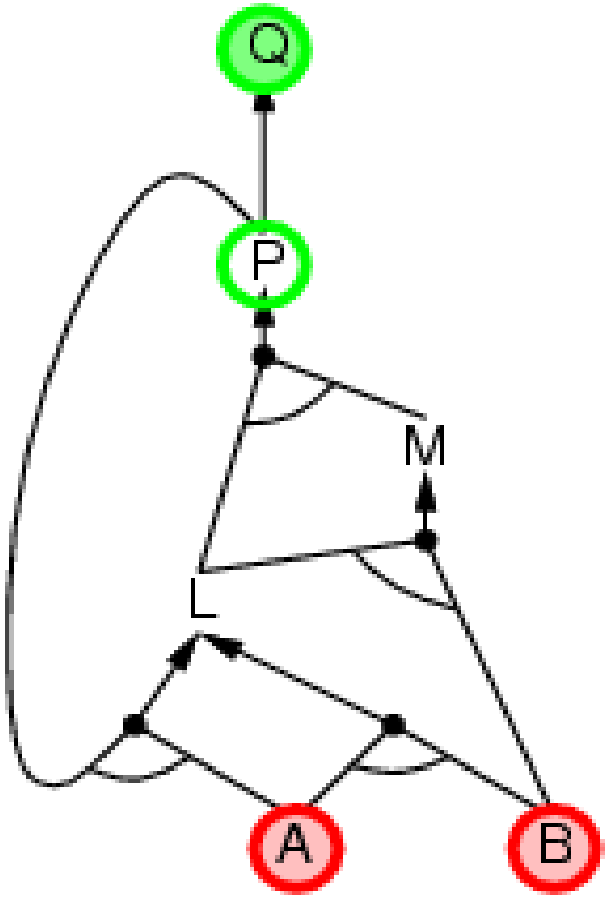
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



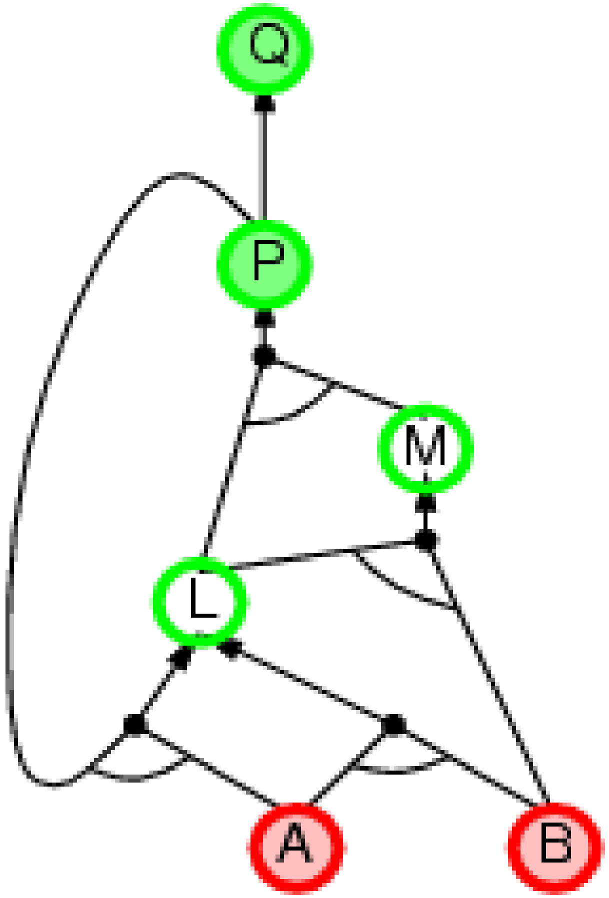
Backward Chaining



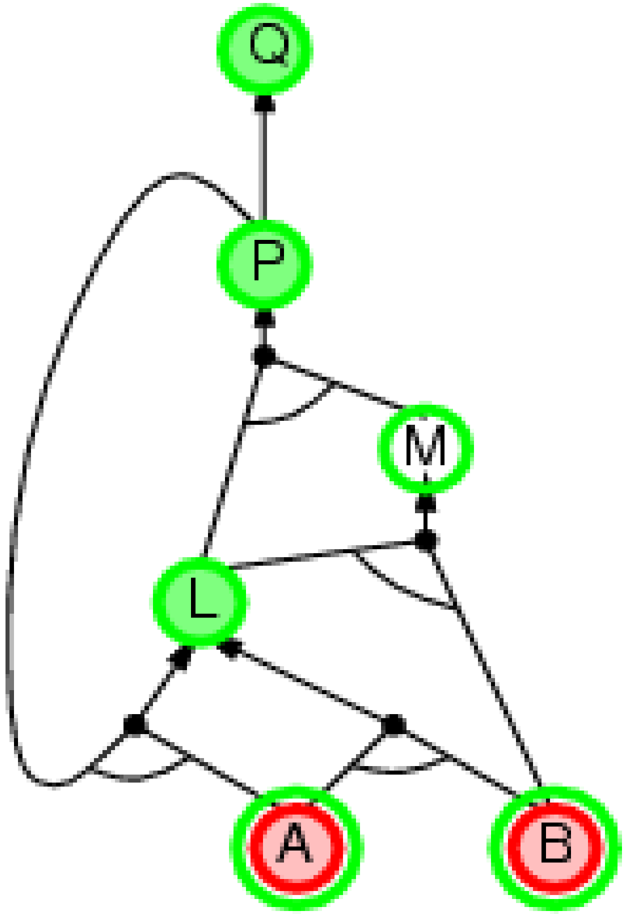
Backward Chaining



Backward Chaining



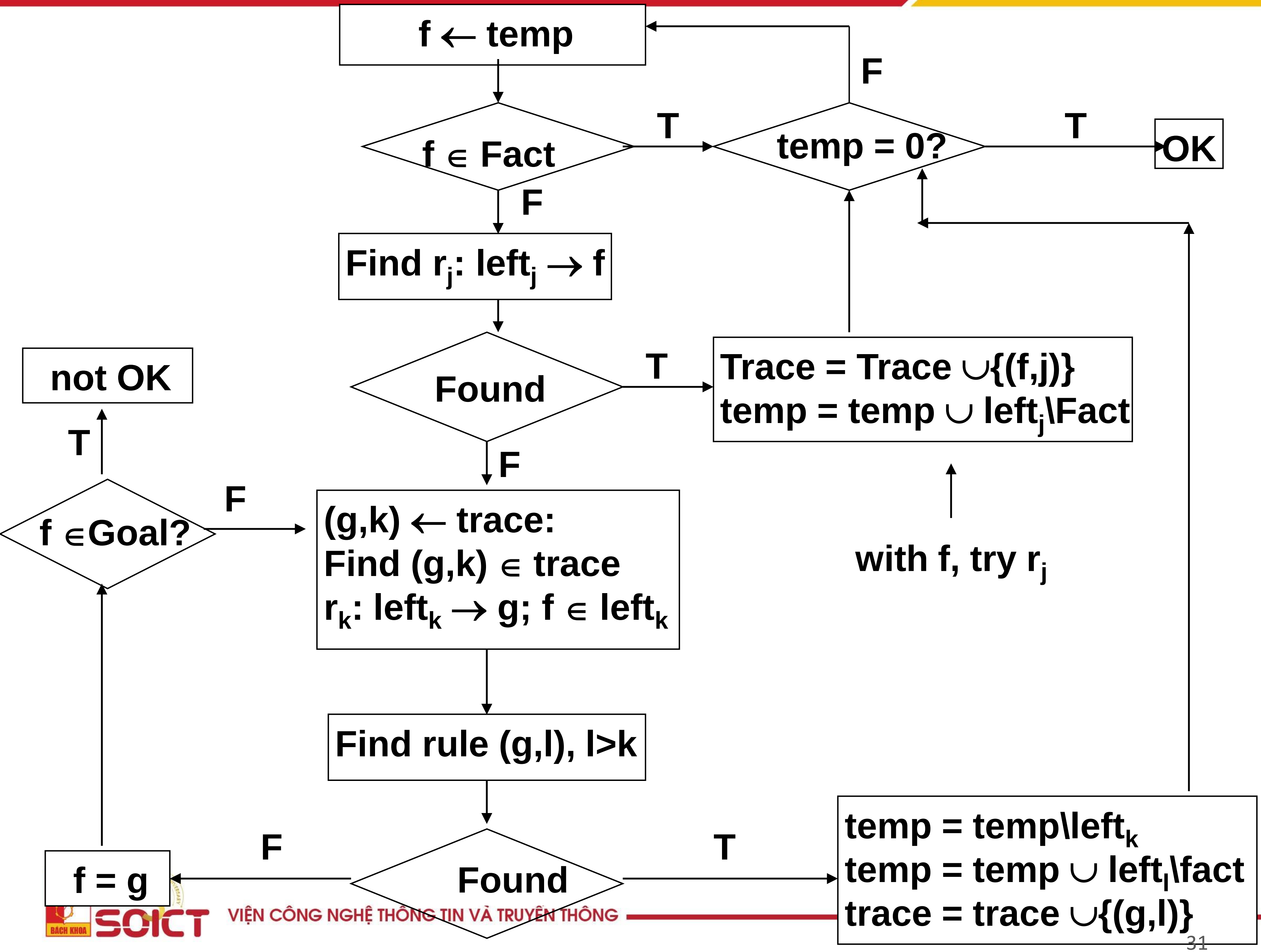
Backward Chaining



Backward Chaining

Variables:

- Goal: set of variables needed to be proved
- temp = {f| f is needed to be proved until now}
- trace ={(f,j)| to prove f, use rule j: $left_j \rightarrow f$}
- Flag Back = true when backtrack
  false otherwise



f ← temp
f ∈ Fact
T
F
temp = 0?
F
T
OK
Find r_j: $left_j \rightarrow f$
not OK
Found
T
F
Trace = Trace ∪{(f,j)}
temp = temp ∪ $left_j$\Fact
T
f ∈Goal?
F
(g,k) ← trace:
Find (g,k) ∈ trace
r_k: $left_k \rightarrow g$; f ∈ $left_k$
with f, try r_j
Find rule (g,l), l>k
F
T
f = g
Found
temp = temp\$left_k$
temp = temp ∪ $left_l$\fact
trace = trace ∪{(g,l)}

Backward Chaining

Example:

1. $A,C \rightarrow B$
2. $a,b,m_a \rightarrow c$
3. $a,b,c \rightarrow A$
4. $a,b,c \rightarrow B$
5. $a,b,c \rightarrow C$
6. $a,B \rightarrow h_c$
7. $b,A \rightarrow h_c$
8. $c,S \rightarrow h_c$
9. $a,b,c \rightarrow S$

1'. $h_a,c \rightarrow B$

Fact=$\{a,b,m_a\}$; Goal=$\{h_c\}$

Exercises

E1. Given Fact={a,b,m_a}, Goal={h_c}

1. a,b,$m_a \rightarrow$ c
2. a,b,C $\rightarrow$ s
3. a,s $\rightarrow h_a$
4. b,s $\rightarrow h_b$
5. c,s $\rightarrow h_c$
6. a,B $\rightarrow h_c$
7. a,b,c $\rightarrow$ B

E2. Given Fact={a}, Goal={u}

1. a $\rightarrow$ b
2. d $\rightarrow$ c
3. c $\rightarrow$ u
4. a $\rightarrow$ m
5. b $\rightarrow$ n
6. m $\rightarrow$ p
7. p $\rightarrow$ q
8. q $\rightarrow$ u

Transformation rules

$$\left.\begin{aligned}(\alpha \land \beta) &\equiv (\beta \land \alpha)\\ (\alpha \lor \beta) &\equiv (\beta \lor \alpha)\end{aligned}\right\} \text{giao hoán}$$

$$\left.\begin{aligned}((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma))\\ ((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma))\end{aligned}\right\} \text{kết hợp}$$

$$\neg(\neg\alpha) \equiv \alpha \quad \text{phủ định kép}$$

$$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{tương phản}$$

$$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \lor \beta)$$

$$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha))$$

$$\left.\begin{aligned}\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta)\\ \neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta)\end{aligned}\right\} \text{de Morgan}$$

$$\left.\begin{aligned}(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma))\\ (\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma))\end{aligned}\right\} \text{phân phối}$$

Transformation rules (con't)

Luật hấp thu:

- $(A \vee (A \wedge B) \equiv A$
- $(A \wedge (A \vee B)) \equiv A$

Các luật về 0, 1:

- $A \wedge 0 \Leftrightarrow 0$
- $A \vee 0 \Leftrightarrow A$
- $A \vee 1 \Leftrightarrow 1$
- $A \wedge 1 \Leftrightarrow A$
- $\neg 1 \Leftrightarrow 0$
- $\neg 0 \Leftrightarrow 1$

Luật bài trung:

- $\neg A \vee A \Leftrightarrow 1$

Luật mâu thuẫn:

- $\neg A \wedge A \Leftrightarrow 0$

Transform into CNF

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

1. Remove $\Leftrightarrow$, replace $\alpha \Leftrightarrow \beta$ by $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.

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

2. Remove $\Rightarrow$, replace $\alpha \Rightarrow \beta$ by $\neg\alpha \vee \beta$.

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

3. Move negation inward using the de Morgan's rule :

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

4. Applying the "and" distribution rule :

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

Example

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

1. Remove $\Rightarrow$

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

2. Move negation inward

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

3. Distribution

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

Exercises

Transform the following expression into CNF.

1. $P \lor (\neg P \land Q \land R)$
2. $(\neg P \land Q) \lor (P \land \neg Q)$
3. $\neg(P \Rightarrow Q) \lor (P \lor Q)$
4. $(P \Rightarrow Q) \Rightarrow R$
5. $(P \Rightarrow (Q \Rightarrow R)) \Rightarrow ((P \land S) \Rightarrow R)$
6. $(P \land (Q \Rightarrow R)) \Rightarrow S$
7. $P \land Q \Rightarrow R \land S$
8. $((a \lor b) \land c) \rightarrow (c \land d)$

Priority: $\neg \land \lor \rightarrow \leftrightarrow$

$$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$$
$$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$$
$$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$$
$$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$$
$$\neg(\neg\alpha) \equiv \alpha$$
$$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$$
$$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$$
$$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$$
$$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$$
$$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$$
$$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$$
$$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$$

1. $P \vee (\neg P \wedge Q \wedge R)$
2. $(\neg P \wedge Q) \vee (P \wedge \neg Q)$
3. $\neg(P \Rightarrow Q) \vee (P \vee Q)$
4. $(P \Rightarrow Q) \Rightarrow R$
5. $(P \Rightarrow (Q \Rightarrow R)) \Rightarrow ((P \wedge S) \Rightarrow R)$
6. $(P \wedge (Q \Rightarrow R)) \Rightarrow S$
7. $P \wedge Q \Rightarrow R \wedge S$

Hợp giải

Dạng kết nối chuẩn (Conjunctive Normal Form - CNF)

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

- Luật hợp giải cho CNF:

$$\frac{l_1 \vee \ldots \vee l_k, \qquad m_1 \vee \ldots \vee m_n}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n}$$

trong đó l_i và m_j bù nhau

E.g., $\dfrac{P_{1,3} \vee P_{2,2}, \quad \neg P_{2,2}}{P_{1,3}}$

Resolution

- Herbrand’s Theorem (~1930)
  - A set of sentences S is unsatisfiable if and only there exists a finite subset S_g of the set of all ground instances Gr(S), which is unsatisfiabe
- Herbrand showed that there is a procedure to demonstrate the unsatisfiability of a unsatisfiable set of sentences
- Robinson propose the Resolution procedure (~1950)

Idea of Resolution

- Refutation-based procedure
  - $S \models A$ if and only if $S \cup \{\neg A\}$ is unsatisfible
- Resolution procedure
  - Transform $S \cup \{\neg A\}$ into a set of clauses
  - Apply Resolution rule to find a the empty clause (contradiction)
    - If the empty clause is found
      - Conclude $S \models A$
    - Otherwise
      - No conclusion

Idea of Resolution

- A clause is a disjunction of literals, i.e., has the form

$$P_1 \vee P_2 \vee \ldots \vee P_n \qquad\qquad P_i \equiv [\neg] R_i$$

- The empty clause corresponds to a contradiction

- Resolution rule

$$\frac{A \vee B \qquad \neg B \vee C}{A \vee C}$$

Robinson’s Resolution

function PL-RESOLUTION(KB, α) **returns** *true* or *false*

clauses ← the set of clauses in the CNF representation of $KB \wedge \neg\alpha$
new ← { }
loop do
  for each C_i, C_j **in** *clauses* **do**
    resolvents ← PL-RESOLVE(C_i, C_j)
    if *resolvents* contains the empty clause **then return** *true*
    new ← *new* ∪ *resolvents*
  if *new* ⊆ *clauses* **then return** *false*
  clauses ← *clauses* ∪ *new*

Robinson's Resolution

Given KB = {P1, P2, ..., Pn}. Prove Q.

Add $\neg$Q to KB: KB = KB $\wedge$ $\neg$Q. Prove unsatisfied.

1. Write each Pi, $\neg$Q in one line.
2. Transfer to CNF representation
   $(a_1 \vee \ldots \vee a_n) \wedge (b_1 \vee \ldots \vee b_n)$ (*)
3. Rewrite each line (*) into smaller lines:
   $a_1 \vee \ldots \vee a_n$
   $b_1 \vee \ldots \vee b_n$

Robinson's Resolution

Consider 2 lines

u) $\neg p \vee q$

v) $p \vee r$

Resolution:

w) $q \vee r$

Contrast appears when KB contains 2 lines:

i) $\neg t$

ii) t

$\Rightarrow$ done

Examples

E1)

1. a
2. $a \rightarrow b$
3. $b \rightarrow (c \rightarrow d)$
4. c

Prove d

E2)

1. $a \wedge b \rightarrow c$
2. $b \wedge c \rightarrow d$
3. a
4. b

Prove d

Examples

E3)

1. p
2. $p \rightarrow q$
3. $q \wedge r \wedge s \rightarrow t$
4. $p \rightarrow u$
5. $v \rightarrow w$
6. $u \rightarrow v$
7. $v \rightarrow t$

Given r,s are true. Prove t

E4)

1. $((a \vee b) \wedge c) \rightarrow (c \wedge d)$
2. $a \wedge m \wedge d \rightarrow f$
3. $m \rightarrow b \wedge c$
4. $a \rightarrow c$
5. $(a \wedge f) \rightarrow (\neg e \vee g)$
6. $(m \wedge f) \rightarrow g$

Given a,m are true. Prove g

Exercise 5

1. $a1 \vee a2 \Rightarrow a3 \vee a4$
2. $a1 \Rightarrow a5$
3. $a2 \wedge a3 \Rightarrow a5$
4. $a2 \wedge a4 \Rightarrow a6 \wedge a7$
5. $a5 \Rightarrow a7$
6. $a1 \wedge a3 \Rightarrow a6 \vee a7$

- Given a1, a2 are true .
- Transfer the above sentences to the CNF representation
- Apply the Robinson's resolution, prove a7 is true.