Artificial Intelligence

Lec 21: First Order Logic (contd.)

Pratik Mazumder

Content taken from various sources

Generalized Modus Ponens

- For atomic sentences pi, pi′, and q, where there is a substitution θ, such that SUBST(θ, p_i′) = SUBST(θ, p_i), for all i,

$$\frac{p1', p2', \ldots, pn', (p1 \land p2 \land \ldots \land pn \Rightarrow q)}{\mathrm{SUBST}(\theta, q)}$$

- There are n + 1 premises to this rule: the n atomic sentences p_i′ and the one implication.

- The conclusion is the result of applying the substitution θ to the consequent q. (Generalized Modus Ponens)

KB
∀ x King(x) ∧ Greedy (x) ⇒ Evil(x)
King(John)
Greedy (John)
Brother (Richard , John)
----
p1′ is King(John)   p1 is King(x)
p2′ is Greedy (y)   p2 is Greedy (x)
θ is {x/John, y/John}   q is Evil(x)
SUBST(θ, q) is Evil(John)

Unification

- Lifted inference rules require finding substitutions that make different logical expressions look identical.

- This process is called unification and is a key component of all First-Order inference algorithms.

- The UNIFY algorithm takes two sentences and returns a unifier for them if one exists.
  UNIFY(p, q) = θ where SUBST(θ, p) = SUBST(θ, q)

- UNIFY(Knows(John, x), Knows(John, Jane)) = {x/Jane}
  UNIFY(Knows(John, x), Knows(y, Bill)) = {x/Bill , y/John}
  UNIFY(Knows(John, x), Knows(y, Mother (y))) = {y/John, x/Mother (John)}
- UNIFY(Knows(John, x), Knows(x, Elizabeth)) = fail

- This problem can be avoided by **standardizing apart** one of the two sentences being unified, which means renaming its variables to avoid name clashes.

- For example, we can rename x in Knows(x, Elizabeth) to x17 (a new variable name) without changing its meaning. Now the unification will work:
  UNIFY(Knows(John, x), Knows(x17, Elizabeth)) = {x/Elizabeth, x17/John}

Forward Chaining

- In FOL, forward chaining is employed to first-order definite clauses
  - **Definite clauses** are **disjunctions of literals** of which **exactly one is positive**, i.e., Indian(x) $\vee$ $\neg$Voter (x)
    - Can also be written using implication: Voter(x)->Indian(x)
  - A **definite clause** either is **atomic or** is an **implication** whose antecedent (**LHS**) is a **conjunction of positive literals** and whose consequent (**RHS**) is a **single positive literal**.

  King(x) $\wedge$ Greedy (x) $\Rightarrow$ Evil(x)
  King(John)
  Greedy(y)

- Unlike propositional literals, first-order literals can include variables, in which case those variables are assumed to be universally quantified.

- **Not every knowledge base can be converted into a set of definite clauses because of the single-positive-literal restriction, but many can.**

Forward Chaining

- The law says that it is a crime for an American to sell weapons to hostile nations.
- The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.
- **Prove:** West is a criminal.

- $\forall$ x,y,z American(x) $\wedge$ Weapon(y) $\wedge$ Sells(x, y, z) $\wedge$ Hostile(z) $\Rightarrow$ Criminal (x)
  $\exists$ x Missile(x) $\wedge$ Owns(Nono, x)
  $\forall$ x Missile(x) $\wedge$ Owns(Nono, x) $\Rightarrow$ Sells(West, x, Nono)
  $\forall$ x Missile(x) $\Rightarrow$ Weapon(x)
  $\forall$ x Enemy(x, America) $\Rightarrow$ Hostile(x)
  American(West)
  Enemy(Nono, America)

  Prove: Criminal(West)

- The sentence $\exists$ x Missile(x) $\wedge$ Owns(Nono, x) is transformed into two definite clauses by Existential Instantiation, introducing a new constant M1
  Missile(M1) $\wedge$ Owns(Nono, M1) –> two definite clauses Missile(M1), Owns(Nono, M1)

Forward Chaining

- KB
  1. $\forall$ x,y,z American(x) $\land$ Weapon(y) $\land$ Sells(x, y, z) $\land$ Hostile(z) $\Rightarrow$ Criminal (x)
  2. $\forall$ x Missile(x) $\land$ Owns(Nono, x) $\Rightarrow$ Sells(West, x, Nono)
  3. $\forall$ x Missile(x) $\Rightarrow$ Weapon(x)
  4. $\forall$ x Enemy(x, America) $\Rightarrow$ Hostile(x)
  5. American(West)
  6. Enemy(Nono, America)
  7. Missile(M1)
  8. Owns(Nono, M1)

  Prove: Criminal(West)

- Can remove $\forall$ since there are only $\forall$ quantifiers

Forward Chaining

- <u>KB</u>
  1. American(x) ∧ Weapon(y) ∧ Sells(x, y, z) ∧ Hostile(z) ⇒ Criminal (x)
  2. Missile(x) ∧ Owns(Nono, x) ⇒ Sells(West, x, Nono)
  3. Missile(x) ⇒ Weapon(x)
  4. Enemy(x, America) ⇒ Hostile(x)
  5. American(West)
  6. Enemy(Nono, America)
  7. Missile(M1)
  8. Owns(Nono, M1)

  Prove: Criminal(West)

- Forward Chaining:
  - Ensure all sentences are definite clauses.
  - Iteratively use GMP on the definite clauses to infer new sentences and add them to KB, till no new sentences can be inferred.
  - On each iteration, add to KB all the atomic sentences that can be inferred in one step from the implication sentences and the atomic sentences already in KB using GMP.
  - In case of substitution conflicts, use STANDARDIZE-APART to replace the variables with new ones that have not been used before.

Forward Chaining

- KB
  1. American(x) ∧ Weapon(y) ∧ Sells(x, y, z) ∧ Hostile(z) ⇒ Criminal (x)
  2. Missile(x) ∧ Owns(Nono, x) ⇒ Sells(West, x, Nono)
  3. Missile(x) ⇒ Weapon(x)
  4. Enemy(x, America) ⇒ Hostile(x)
  5. American(West)
  6. Enemy(Nono, America)
  7. Missile(M1)
  8. Owns(Nono, M1)

  Prove: Criminal(West)

- Consider 2 which is in the format p1∧p2⇒q, and check if there are atomic sentences p1′, p2′, for which there is a substitution θ, such that SUBST(θ, p1′) = SUBST(θ, p1) and SUBST(θ, p2′) = SUBST(θ, p2)
- We can see that if we use θ = {x/M1}, then Missile(x) = Missile(M1) and Owns(Nono, x) = Owns(Nono, M1)
- Apply GMP to 2,7,8, and get SUBST(θ, q) = SUBST({x/M1}, Sells(West, x, Nono)) = Sells(West, M1, Nono)
  9. Sells(West, M1, Nono)
- Similarly, using θ = {x/M1} and GMP on 3,7, we get
  10. Weapon(M1)

Forward Chaining

- <u>KB</u>
  1. American(x) ∧ Weapon(y) ∧ Sells(x, y, z) ∧ Hostile(z) ⇒ Criminal (x)
  2. Missile(x) ∧ Owns(Nono, x) ⇒ Sells(West, x, Nono)
  3. Missile(x) ⇒ Weapon(x)
  4. Enemy(x, America) ⇒ Hostile(x)
  5. American(West)
  6. Enemy(Nono, America)
  7. Missile(M1)
  8. Owns(Nono, M1)

  Prove: Criminal(West)

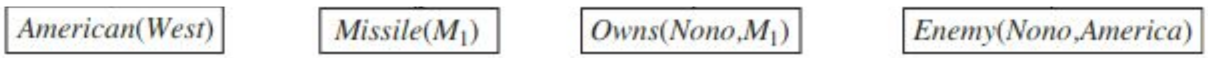
  9. Sells(West, M1, Nono)
  10. Weapon(M1)
- Using θ = {x/Nono} and GMP on 4,6, we get

  11. Hostile(Nono)

- Using θ = {x/West,y/M1,z/Nono} and GMP on 1,5,10,9,11 we get
  SUBST({x/West,y/M1,z/Nono},Criminal (x)) = Criminal(West)

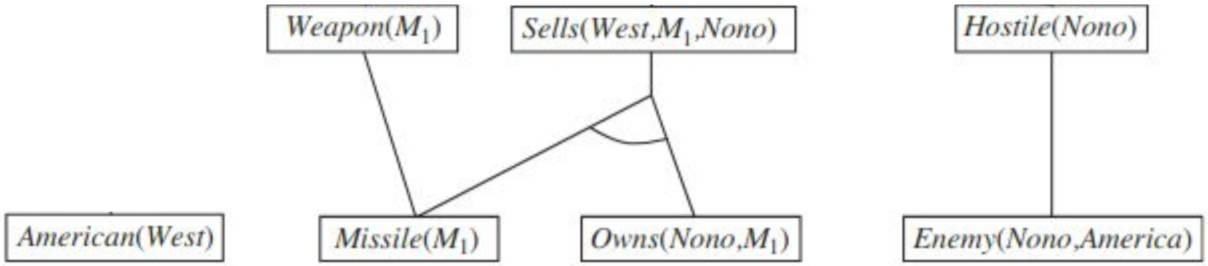
Forward Chaining

1. American(x) ∧ Weapon(y) ∧ Sells(x, y, z) ∧ Hostile(z) ⇒ Criminal (x)
2. Missile(x) ∧ Owns(Nono, x) ⇒ Sells(West, x, Nono)
3. Missile(x) ⇒ Weapon(x)
4. Enemy(x, America) ⇒ Hostile(x)
5. American(West)
6. Enemy(Nono, America)
7. Missile(M1)
8. Owns(Nono, M1)



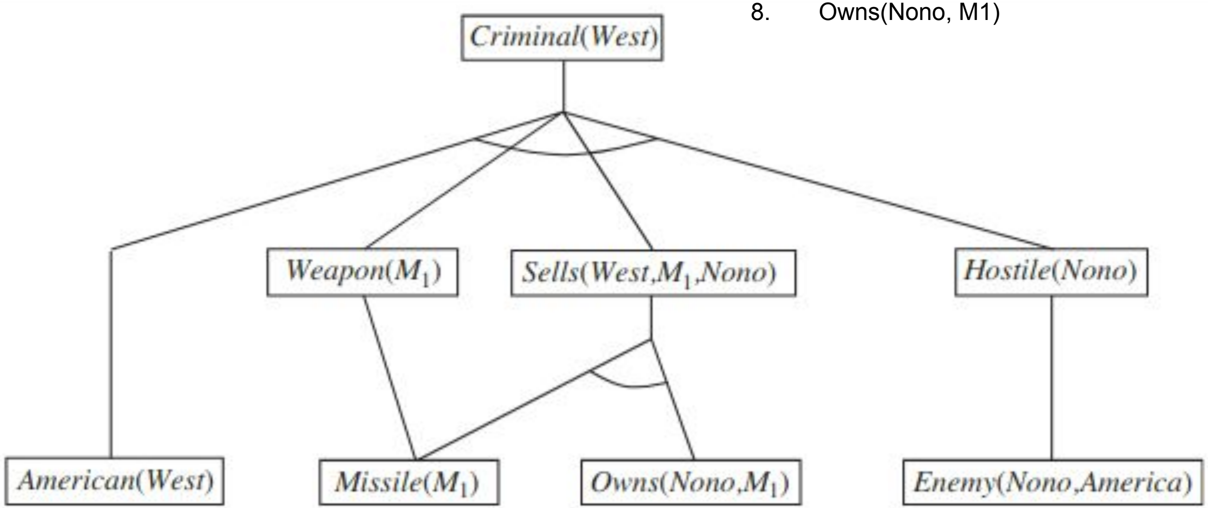
Forward Chaining

1. American(x) ∧ Weapon(y) ∧ Sells(x, y, z) ∧ Hostile(z) ⇒ Criminal (x)
2. Missile(x) ∧ Owns(Nono, x) ⇒ Sells(West, x, Nono)
3. Missile(x) ⇒ Weapon(x)
4. Enemy(x, America) ⇒ Hostile(x)
5. American(West)
6. Enemy(Nono, America)
7. Missile(M1)
8. Owns(Nono, M1)



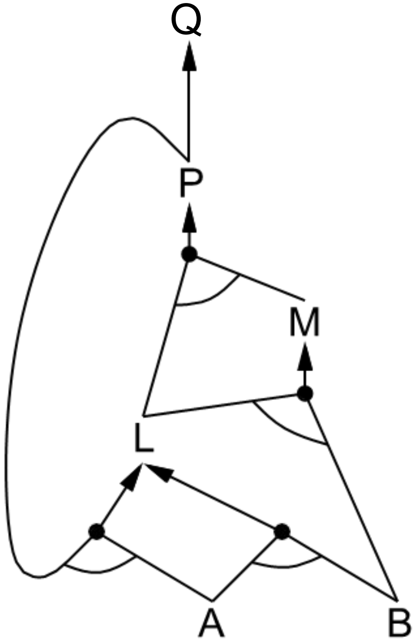
Forward Chaining

1. American(x) ∧ Weapon(y) ∧ Sells(x, y, z) ∧ Hostile(z) ⇒ Criminal (x)
2. Missile(x) ∧ Owns(Nono, x) ⇒ Sells(West, x, Nono)
3. Missile(x) ⇒ Weapon(x)
4. Enemy(x, America) ⇒ Hostile(x)
5. American(West)
6. Enemy(Nono, America)
7. Missile(M1)
8. Owns(Nono, M1)



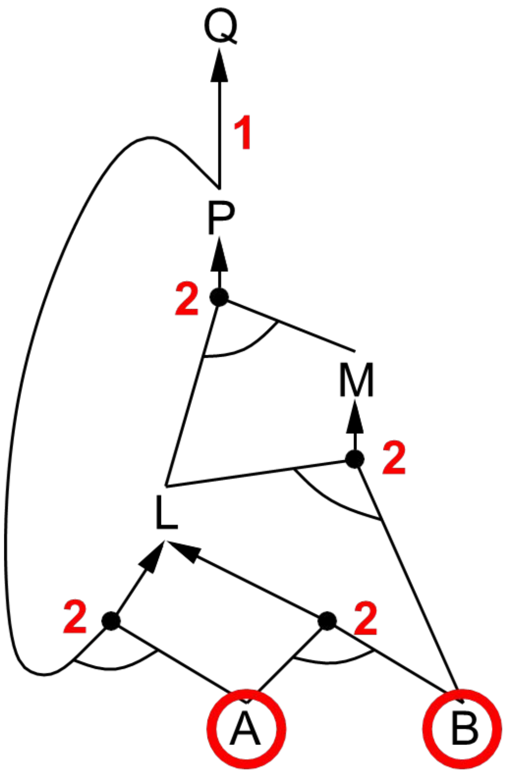
Forward Chaining in PL - Graphical Solution

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



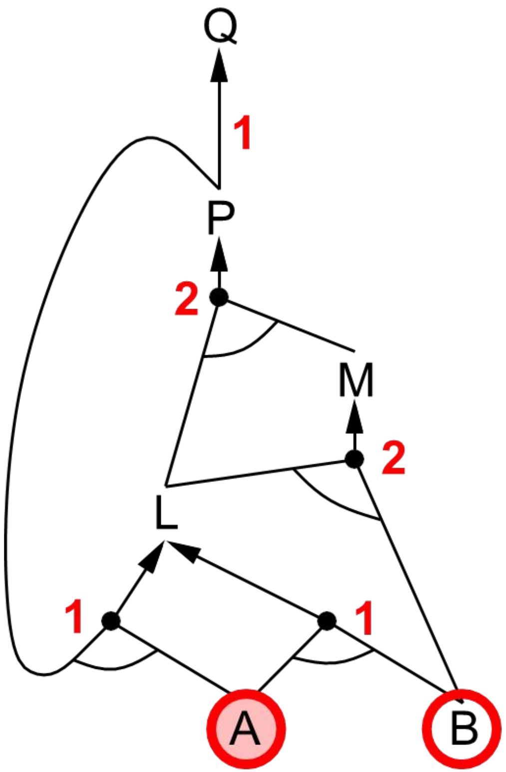
Forward Chaining in PL - Graphical Solution

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



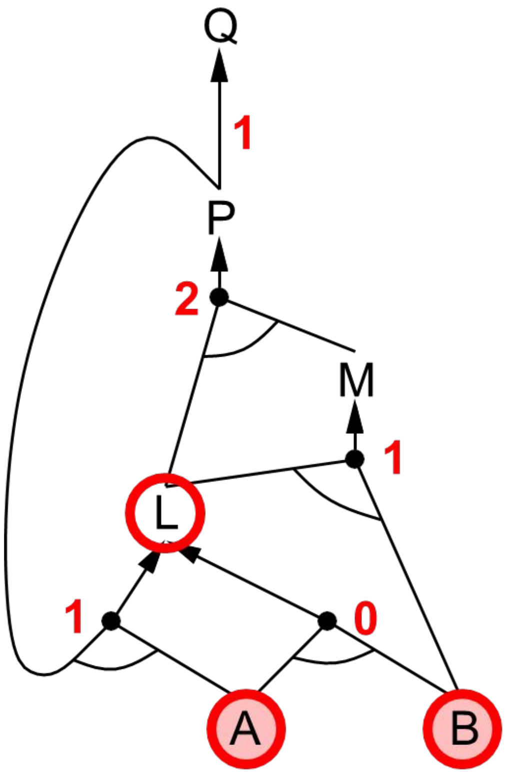
Forward Chaining in PL - Graphical Solution

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



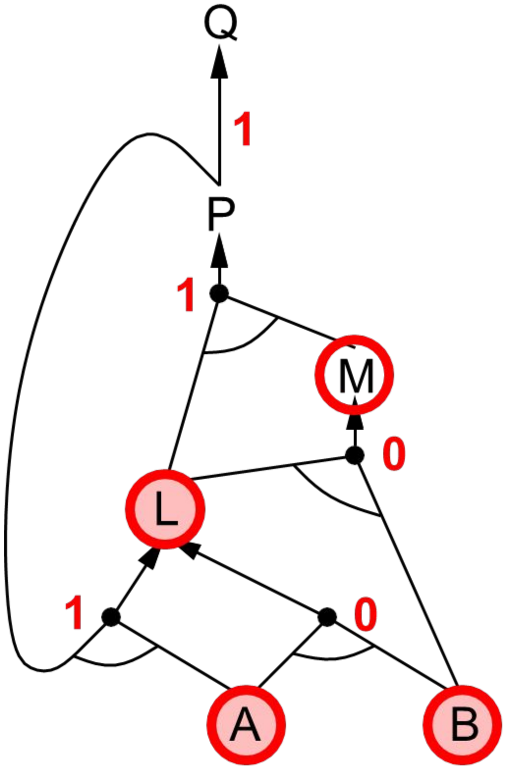
Forward Chaining in PL - Graphical Solution

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



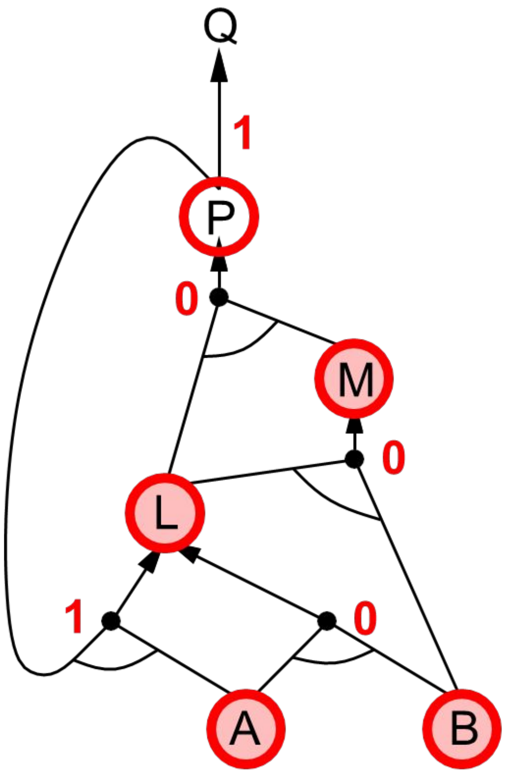
Forward Chaining in PL - Graphical Solution

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



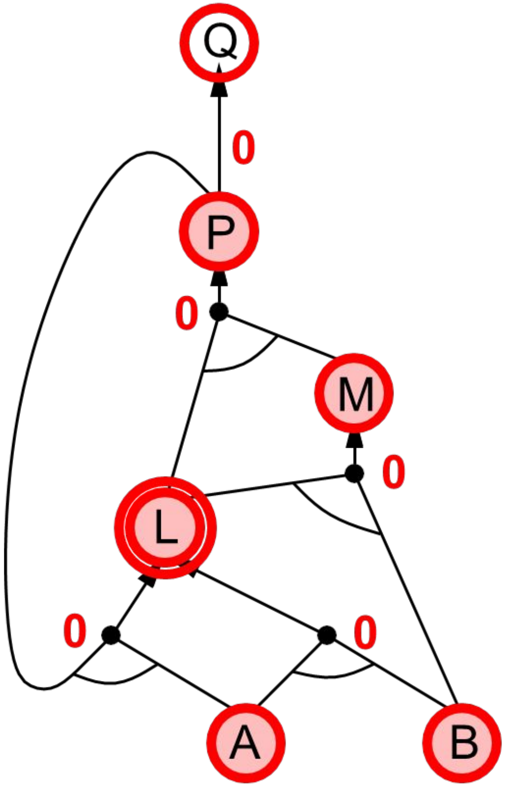
Forward Chaining in PL - Graphical Solution

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



Forward Chaining in PL - Graphical Solution

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



Forward Chaining in PL - Graphical Solution

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