Inference Rules & Resolution

Prof. Dr. A. K. M. Akhtar Hossain

Dept. of CSE, University of Rajshahi

Inference Rules

- Like PL, a key inference rule in FOPL is **modus ponens**.
- From the assertion "**Leo is a lion**" and the implication "**all lions are ferocious**" .
- We can conclude that **Leo is ferocious**.
- Written in symbolic form we have :
  - **Assertion: LION (leo)**
  - **Implication : $\forall$x LION (x) $\rightarrow$ FEROCIOUS (x)**
  - **Conclusion : FEROCIOUS (leo)**

Resolution

- Its an inference rule used in both propositional as well as First-Order Predicate Logic (FOPL/FOL) in different ways.
- It is also called **proof by Refutation**.
- It is basically used for proving the satisfiability of a sentence.
- Proof by call refutation technique is used to prove the given statement. It’s a **iterative process**:
- **Step 1**: Select two clauses that contain conflicting terms.
- **Step 2**: Combine those two clauses and cancel out the conflicting terms, resulting a new clause that has been inferred from them.

Resolution

- The key idea is to use the knowledge base and **negated goal** to obtain null clause (which indicates contradiction).
- Since the knowledge base itself is consistent, **the contradiction must be introduced by a negated goal.**
- As a result we have to conclude that the original goal is to true.

Steps in Resolution (Algorithm)

- 1. Convert facts into First Order Predicate Logic (FOPL/FOL).
- 2. Convert into FOPL to CNF (Conjunctive Normal Form).
- 3. Negate the statement to be proved and add the result to the knowledge base.
- 4. Draw Resolution graph.
- 5. If empty clause (Nil) is produced, stop and report that original theorem is true.

Table: Lists some of the important laws of PL (Some Equivalence Laws)

Name of Laws	Statements
Idempotency	P V P = P P & P = P
Associativity	(PVQ)VR = P V (Q V R) (P & Q) & R = P & (Q & R)
Commutativity	P V Q = Q V P P & Q = Q & P P ↔Q = Q↔P
Distributivity	P & (Q V R) = (P & Q)V (P &R) P V (Q & R) = (P V Q) & (P V R)
De Morgan's Laws	~(P V Q) = ~P & ~Q ~(P & Q) = ~P V ~Q
Conditional Elimination	P → Q = ~P V Q
Bi-conditional Elimination	P ↔Q = (p → Q) & (Q →P)

Table: Equivalent Logical Expressions

Name of the Rules	Expressions
Double negation	~(~F) = F
Commutativity	F & G = G & F, F V G = G V F
Associativity	(F & G) & H = F & (G & H) (F V G) V H = F V (G V H)
Distributivity	F V (G & H) = (F V G) & (F V H) F & (G V H) = (F & G) V (F & H)
De Morgan's Laws	~(F & G) = ~F V ~G ~(F V G) = ~F & ~G
Conditional Elimination	F → G = ~F V G

Table: Equivalent Logical Expressions Cont..

Bi-conditional Elimination	$F \leftrightarrow G = (\sim F \lor G) \,\&\, (\sim G \lor F)$
Quantifiers	$\forall x\, F[x] \lor G = \forall x\, (F[x] \lor G)$ $\exists x\, F[x] \lor G = \exists x\, (F[x] \lor G)$ $\forall x\, F[x] \,\&\, G = \forall x\, (F[x] \,\&\, G)$ $\exists x\, F[x] \,\&\, G = \exists x\, (F[x] \,\&\, G)$ $\sim(\forall x)\, F[x] = \exists x\, (\sim F[x])$ $\sim(\exists x)\, F[x] = \forall x\, (\sim F[x])$ $\forall x\, F[x] \,\&\, \forall x\, G[x] = \forall x\, (F[x] \,\&\, G[x])$ $\exists x\, F[x] \lor \exists x\, G[x] = \exists x\, (F[x] \lor G[x])$

Example #1

- Consider the following Knowledge Base:

1. The humidity is high or the sky is cloudy.
2. If the sky is cloudy, then it will rain.
3. If the humidity is high, then it is hot.
4. It is not hot.

Goal: It will rain.

Step1: Converted into FOPL/FOL

1. The humidity is high or the sky is cloudy.
2. If the sky is cloudy, then it will rain.
3. If the humidity is high, then it is hot.
4. It is not hot.

- Let P : The humidity is high
- Let Q: sky is cloudy

The humidity is high or the sky is cloudy

$P \vee Q$

Step1: Converted into FOPL/FOL

1. The humidity is high or the sky is cloudy.
2. If the sky is cloudy, then it will rain.
3. If the humidity is high, then it is hot.
4. It is not hot.

Goal: It will rain.

- Let Q: sky is cloudy
- Let R: it will rain

If the sky is cloudy, then it will rain

$Q \rightarrow R$

Step1: Converted into FOPL/FOL

1. The humidity is high or the sky is cloudy.
2. If the sky is cloudy, then it will rain.
3. If the humidity is high, then it is hot.
4. It is not hot.

- **Goal:** It will rain.

- Let P : The humidity is high
- Let S: it is hot

If the humidity is high, then it is hot

$P \rightarrow S$

Step1: Converted into FOPL/FOL

1. The humidity is high or the sky is cloudy.
2. If the sky is cloudy, then it will rain.
3. If the humidity is high, then it is hot.
4. It is not hot.

- **Goal:** It will rain.

- Let S: it is hot

It is not hot

$\neg S$

Step1: Converted into FOPL/FOL

1. The humidity is high or the sky is cloudy.
2. If the sky is cloudy, then it will rain.
3. If the humidity is high, then it is hot.
4. It is not hot.

- $P \vee Q$
- $Q \rightarrow R$
- $P \rightarrow S$
- $\neg S$

- **Goal:** It will rain. (R)

Step 2: Convert into FOPL to CNF &

- P V Q
- Q→R
- P→S
- ¬S

- P V Q
- ¬Q V R
- ¬P V S
- ¬S

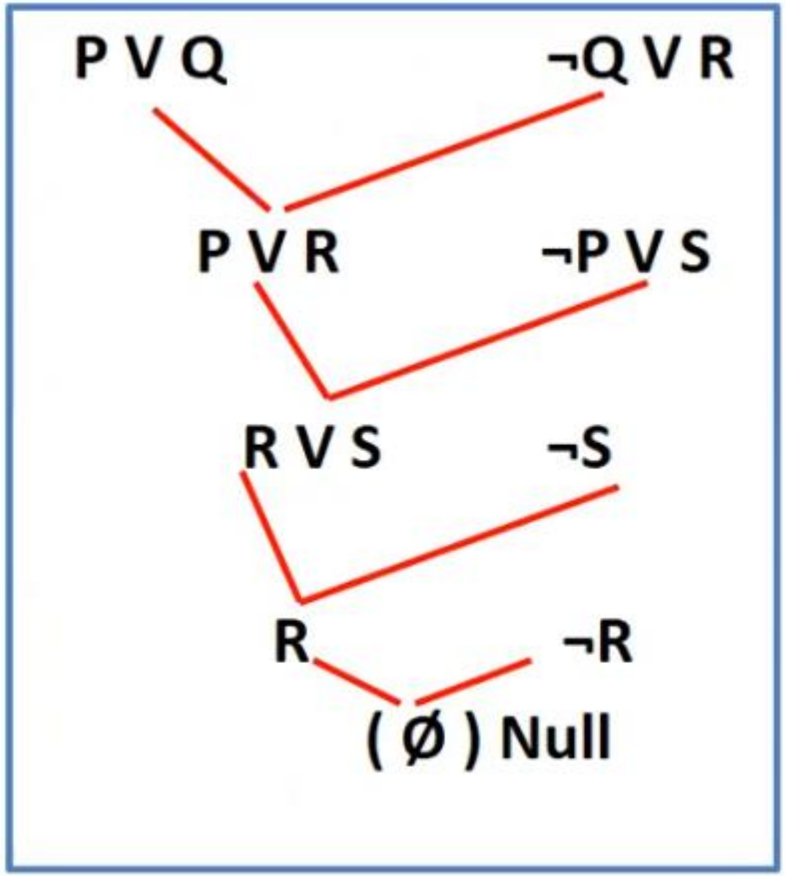
Step 3: Negation the Goal

- GOAL: It will rain . (R)

Negation of Goal (¬R): It will not rain.

Step 4: Resolution of Graph

- P V Q
- ¬Q V R
- ¬P V S
- ¬S
- ¬R (Goal)



Assignment #4

- **Consider following the knowledge based:**
- 1. Marcus was a man.
- 2. Marcus was a Pompeian.
- 3. All Pompeian's were Romans.
- 4. Caesar was a ruler.
- 5. All Romans were either loyal to Caesar or hated him.
- 6. Everyone is loyal to someone.
- 7. People only try to assassinate rulers they are not loyal to.
- 8. Marcus tried to assassinate Caesar.
- ***GOAL:***

  Is Marcus hate Caesar?
- Or, **hate(Marcus, Caesar)**

Step 1: Convert into FOPL/FOL

- 1. Marcus was a man.
- PL: man(Marcus)
- 2. Marcus was a Pompeian.
- PL: Pompeian(Marcus)
- 3. All Pompeian's were Romans.
- PL: $\forall x$:Pompeian(x) $\rightarrow$ Roman(x)
- 4. Caesar was a ruler.
- PL: rular(Caesar)
- 5. All Romans were either loyal to Caesar or hated him.
- PL: $\forall x$: Roman(x) $\rightarrow$ loyalto(x, Ceaser) V hate (x, Caesar)

Step 1: Convert into FOL

- 6. Everyone is loyal to someone.
- PL: $\forall x :\rightarrow y : loyalto\ (x, y)$
- PL: $\exists y: \forall x : loyalto\ (x, y)$
- 7. People only try to assassinate rulers they are not loyal to.
- PL: $\exists y: \forall x : person(x) \land ruler(y) \land tryassassinate(x, y) \rightarrow \neg loyalto\ (x, y)$
- 8. Marcus tried to assassinate Caesar.
- PL: tryassassinate (Marcus, Caesar)

Step2: Convert FOL into CNF

- 1. man(Marcus)
- 2. Pompeian(Marcus)
- 3. ¬Pompeian(x_1) V Roman(x_1)
- 4. rular(Caesar)
- 5. ¬Roman(x_2) V loyalto(x_2,Caesar) V hate(x_2, Caesar)
- 6. loyalto(x_3, fl(x_3))
- 7. ¬man(x_4) V ¬ruler(y_1) V ¬tryassassinate(x_4,y_1) V loyalto(x_4, y_1)
- 8. tryassassinate(Marcus, Caesar)

GOAL

- You have to write the remaining steps in order to prove the goal.

Assignment #4

- 1. If it is sunny and warm day you will enjoy.
- 2. If it is raining then you will get wet.
- 3. It is warm day.
- 4. It is raining.
- 5. It is sunny.

- Goal: You will enjoy.
- Prove: enjoy.

- END TODAY
- THANKS