Inference in First-order Logic

Yifeng Zeng

yfzeng@cs.aau.dk

Outline

- Propositionalize FOL
- Generalized Modus Ponens
- Generalized Resolution
- Resolution Strategies

Universal Instantiation(UI)

- Every instantiation of a universally quantified sentence is entailed for any variable v and ground term θ

$$\frac{\forall v \quad \alpha}{SUBST(v/\theta, \alpha)}$$

- The substitution has to be done by a ground term
  - Ground term contains no variable and can not be further instantiated

E.g., $\forall x \quad Study(x, AAU)$ with the substitution $\{x/Thomas\}$ gives us $Study(Thomas, AAU)$

Existential Instantiation(EI)-*Skolemization*

- For any sentence α, variable v, and constant symbol(Skole constant) H that does not appear elsewhere in the knowledge base
  - Those contained in the knowledge base are not known whether they are **True** or **False**

$$\frac{\exists v \quad \alpha}{SUBST(v/H, \alpha)}$$

E.g., $\exists x \ Study(x, AAU)$ with the substitution $\{x/Student\}$ we may infer $Study(Student, AAU)$ as long as Student does not appear elsewhere in the knowledge base

UI and EI

- UI can be applied several times to add new sentences
  - The new KB is logically equivalent to the old
- EI can be applied once to replace the existential sentence
  - The new KB is not equivalent to the old, but is satisfiable iff the old was satisfiable

Example: Reasoning in FOL

- The law says that it is a crime for an American to sell weapon 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. Colonel West is American.
  - Prove that Colonel West is criminal
- Reasoning
  - Formulate FOL
  - Use UI and EI to inference

Formulation

- $\cdots$ it is a crime for an American to sell weapon to hostile nations $\cdots$
  - P_1: $\forall x, y, z \quad American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$
- $\cdots$ Nono has some missiles $\cdots$
  - P_2:$\exists x \quad Owns(Nono, x) \wedge Missile(x)$
- $\cdots$ all of its missiles were sold to it by Colonel West $\cdots$
  - P_3:$\forall x \quad Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$
- $\cdots$ Missiles are weapons $\cdots$
  - P_4:$\forall x \quad Missile(x) \Rightarrow Weapon(x)$
- $\cdots$ an enemy of America counts as "hostile" $\cdots$
  - P_5:$\forall x \quad Enemy(x, America) \Rightarrow Hostile(x)$
- $\cdots$ the country Nono, an enemy of America $\cdots$
  - P_6:$Enemy(Nono, America)$
- $\cdots$ West, who is America $\cdots$
  - P_7:$American(West)$

Forward Chaining

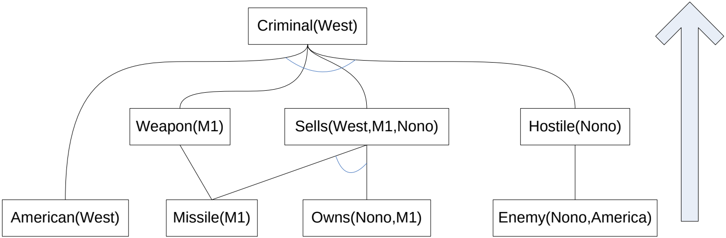
When a new fact P is added into the KB

For each rule such that P unifies with a premise

If the other premises are known

Add the conclusion into the KB

Forward Chaining Proof



Backward Chaining

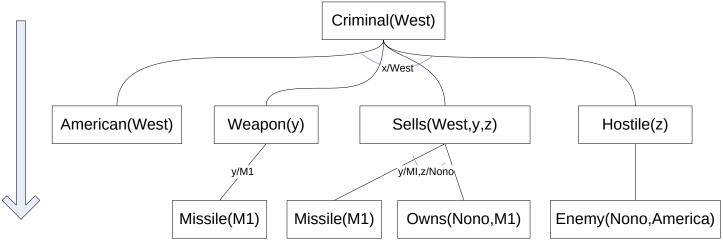
When a query Q is asked

If a matching fact Q' is known, return the unifier

For each rule whose consequent Q' matches Q

Attempt to prove each premise of the rule

Backward Chaining Proof



Propositionalize KB

- Repeat the use of UI and EI to reduce the FOL inference to propositional inference
- Problem 1: Generate lots of irrelevant sentences
- Problem 2: With function symbols, there are infinitely many ground terms
    - $Father(Father(Father(Father(John))))$

Generalized Modus Ponens(GMP)

- For atomic sentences p_i, $p_i^{'}$, and q where there is a substitution θ such that $SUBST(\theta, p_i^{'}) = SUBST(\theta, p_i)$ for all i:
  - p_i and $p_i^{'}$ are unified with the substitution θ

$$\frac{p_1^{'}, p_2^{'}, \cdots, p_n^{'} \quad (p_1 \wedge p_2 \wedge \cdots p_n \Rightarrow q)}{SUBST(\theta, q)}$$

Unification

- **UNIFY(p,q)=**θ where $SUBST(\theta, p) = SUBST(\theta, q)$
- E.g., $UNIFY(Knows(Thomas, x), Knows(Thomas, Bill)) = ?$
  - $\{x/Bill\}$
- E.g., $UNIFY(Knows(Thomas, x), Knows(y, Bill)) = ?$
  - $\{x/Bill, y/Thomas\}$
- E.g.,
  $UNIFY(Knows(Thomas, x), Knows(y, Friends(y))) = ?$
  - $\{y/Bill, x/Friends(y)\}$
- E.g., $UNIFY(Knows(Thomas, x), Knows(x, Bill)) = ?$
  - *fail*

Unification - Note

- E.g., $UNIFY(Knows(Thomas, x), Knows(y, z)) = ?$
  - $\{y/Thomas, x/Bill, z/Bill\}$
  - $\{y/Thomas, x/z\}$

We require the most general unifier

Reasoning with Horn Logic

- We can convert Horn sentences to a canonical form and then use generalized Modus Ponens with unification.
  - We skolemize existential formulas and remove the universal quantifiers
  - This gives us a conjunction of clauses, that are inserted into the KB
  - Modus Ponens help us in inferring new clauses
- Forward and backward chaining

Completeness Issues

- Reasoning with Modus Ponens is incomplete
- Example
  - $\forall x \;\; PhD(x) \Rightarrow HighlyQualified(x)$
  - $\forall x \;\; \neg PhD(x) \Rightarrow EarlyEarnings(x)$
  - $\forall x \;\; HighlyQualified(x) \Rightarrow Rich(x)$
  - $\forall x \;\; EarlyEarnings(x) \Rightarrow Rich(x)$
- We should be able to conclude $Rich(Me)$, but forward and backward chaining will not achieve it
- The problem is that $\forall x \;\; \neg PhD(x) \Rightarrow EarlyEarnings(x)$ cannot be converted to Horn form, and thus cannot be used by Modus Ponens

Resolution - Generalized Resolution Rule

- For atoms p_j,q_k, where $Unify(p_j, -q_k) = \theta$, we have

$$\frac{p_1 \vee \cdots \vee p_j \vee \cdots \vee p_{n1} \qquad q_1 \vee \cdots \vee q_k \vee \cdots \vee q_{n2}}{SUBST(\theta, p_1 \vee \cdots \vee p_{j-1} \vee p_{j+1} \vee \cdots \vee p_{n1} \vee q_1 \vee \cdots \vee q_{k-1} \vee q_{k+1} \vee \cdots \vee q_{n2})}$$

Resolution - Generalized Resolution Rule(different form!)

- For atoms p_i,q_i,r_i,s_i, where $Unify(p_j, q_k) = \theta$, we have:

$$\frac{p_1 \wedge \cdots p_j \wedge \cdots \wedge p_{n1} \Rightarrow r_1 \vee \cdots \vee r_{n2} \qquad s_1 \wedge \cdots \wedge s_{n3} \Rightarrow q_1 \vee \cdots \vee q_k \vee \cdots \vee q_{n4}}{SUBST(\theta, p_1 \wedge \cdots p_{j-1} \wedge p_{j+1} \wedge \cdots \wedge p_{n1} \wedge s_1 \wedge \cdots \wedge s_{n3} \Rightarrow r_1 \vee \cdots \vee r_{n2} \vee q_1 \vee \cdots \vee q_{k-1} \vee q_{k+1} \vee \cdots \vee q_{n4}}$$

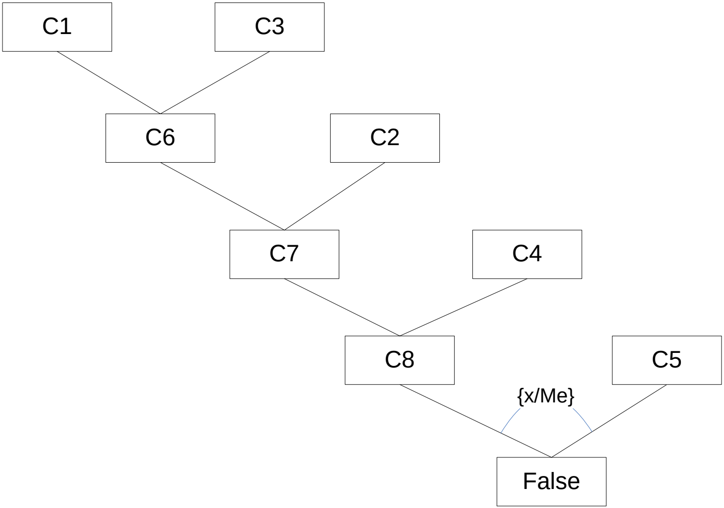
Example

- Knowledge base
  - $\forall x \;\; PhD(x) \Rightarrow HighlyQualified(x)$
    - C1:$\neg PhD(x) \vee HighlyQualified(x)$
  - $\forall x \;\; \neg PhD(x) \Rightarrow EarlyEarnings(x)$
    - C2:$PhD(x) \vee EarlyEarnings(x)$
  - $\forall x \;\; HighlyQualified(x) \Rightarrow Rich(x)$
    - C3:$\neg HighltQualified(x) \vee Rich(x)$
  - $\forall x \;\; EarlyEarnings(x) \Rightarrow Rich(x)$
    - C4:$\neg EarlyEarnings(x) \vee Rich(x)$
- Theorem
  - $Rich(Me)$
- Add C5:$\neg Rich(Me)$ into the knowledge base and use resolution rules to deduce contradiction

Inference - 1

- From C1 and C3, we get C6: $\neg PhD(x) \vee Rich(x)$
- From C6 and C2, we get C7: $EarlyEarnings(x) \vee Rich(x)$
- From C7 and C4, we get C8: $Rich(x)$
- From C8 and C5, we get **False**

Inference - 2



Conversion to Normal Form

- A formula is said to be in clause form if it is of the form
  - $\forall x_1, x_1, \cdots, x_n \;\; [C_1 \wedge \cdots \wedge C_k]$
- All first-order logic formulas can be converted to clause form

$$\forall x \quad \{p(x) \Rightarrow \exists z \;\; \{\neg \forall y \;\; [q(x, y) \Rightarrow p(f(x_1))] \wedge \forall y \;\; [q(x, y) \Rightarrow p(x)]\}\}$$

Step 1

- Take the existential clause and eliminate redundant quantifiers.
  - Introduce $\exists x_1$ and eliminate $\exists z$

$$\forall x \quad \{p(x) \Rightarrow \exists z \ \{\neg\forall y \ [q(x,y) \Rightarrow p(f(x_1))] \wedge \forall y \ [q(x,y) \Rightarrow p(x)]\}\}$$

$$\downarrow$$

$$\boxed{\exists x_1 \ \forall x \quad \{p(x) \Rightarrow \{\neg\forall y \ [q(x,y) \Rightarrow p(f(x_1))] \wedge \forall y \ [q(x,y) \Rightarrow p(x)]\}}$$

Step 2

- Rename any variable that is quantified more than once
  - y has been quantified twice

$$\exists x_1 \ \ \forall x \ \ \{p(x) \Rightarrow \{\neg\forall y \ \ [q(x,y) \Rightarrow p(f(x_1))] \wedge \forall y \ \ [q(x,y) \Rightarrow p(x)]\}\}$$

$$\downarrow$$

$$\boxed{\exists x_1 \ \ \forall x \ \ \{p(x) \Rightarrow \{\neg\forall y \ \ [q(x,y) \Rightarrow p(f(x_1))] \wedge \forall z \ \ [q(x,z) \Rightarrow p(x)]\}\}}$$

Step 3

- Eliminate implication

$$\exists x_1 \ \forall x \ \ \{p(x) \Rightarrow \{\neg \forall y \ \ [q(x,y) \Rightarrow p(f(x_1))] \wedge \forall z \ \ [q(x,z) \Rightarrow p(x)]\}\}$$

$$\downarrow$$

$$\boxed{\exists x_1 \ \forall x \ \ \{\neg p(x) \vee \{\neg \forall y \ \ [\neg q(x,y) \vee p(f(x_1))] \wedge \forall z \ \ [\neg q(x,z) \vee p(x)]\}\}}$$

Step 4

- Move $\neg$ all the way inwards

$$\exists x_1 \;\; \forall x \;\; \{\neg p(x) \vee \{\neg \forall y \;\; [\neg q(x,y) \vee p(f(x_1))] \wedge \forall z \;\; [\neg q(x,z) \vee p(x)]\}\}$$

$$\downarrow$$

$$\boxed{\exists x_1 \;\; \forall x \;\; \{\neg p(x) \vee \{\exists y \;\; [q(x,y) \wedge \neg p(f(x_1))] \wedge \forall z \;\; [\neg q(x,z) \vee p(x)]\}}$$

Step 5

- Push the quantifier to the right

$$\exists x_1 \ \forall x \ \ \{\neg p(x) \vee \{\exists y \ \ [q(x,y) \wedge \neg p(f(x_1))] \wedge \forall z \ \ [\neg q(x,z) \vee p(x)]\}\}$$

$$\downarrow$$

$$\boxed{\exists x_1 \ \forall x \ \ \{\neg p(x) \vee \{[\exists y \ \ q(x,y) \wedge \neg p(f(x_1))] \wedge [\forall z \ \ \neg q(x,z) \vee p(x)]\}\}}$$

Step 6

- Eliminate existential quantifiers(Skolemization)
  - Pick out the leftmost $\exists y \;\; B(y)$ and replace it by $B(f(x_1, x_2, \cdots, x_n))$ where
    - $x_1, x_2, \cdots, x_n$ are all the distinct free variables of $\exists y \;\; B(y)$ that are universally quantified to the left of $\exists y \;\; B(y)$
    - $f(\cdot)$ is any n-ary function constant which does not occur already.

$$\exists x_1 \;\; \forall x \quad \{\neg p(x) \vee \{[\exists y \;\; q(x, y) \wedge \neg p(f(x_1))] \wedge [\forall z \;\; \neg q(x, z) \vee p(x)]\}\}$$

$$\downarrow$$

$$\boxed{\forall x \quad \{\neg p(x) \vee \{[q(x, g(x)) \wedge \neg p(f(a))] \wedge [\forall z \;\; \neg q(x, z) \vee p(x)]\}\}}$$

Step 7

- Move all universal quantifiers to the left

$$\forall x \quad \{\neg p(x) \vee \{[q(x, g(x)) \wedge \neg p(f(a))] \wedge [\forall z \;\; \neg q(x, z) \vee p(x)]\}\}$$

$$\downarrow$$

$$\boxed{\forall x \;\; \forall z \quad \{\neg p(x) \vee \{[q(x, g(x)) \wedge \neg p(f(a))] \wedge [\;\; \neg q(x, z) \vee p(x)]\}\}}$$

Step 8

- Distribute $\wedge$ over $\vee$

$$\forall x \ \ \forall z \ \ \{\neg p(x) \vee \{[q(x, g(x)) \wedge \neg p(f(a))] \wedge [\ \neg q(x, z) \vee p(x)]\}\}$$

$$\downarrow$$

$$\boxed{\forall x\{[\neg p(x) \vee q(x, g(x))] \wedge \neg p(f(a))\}}$$

$$\downarrow$$

$$\boxed{\{[\neg p(x) \vee q(x, g(x))] \wedge \neg p(f(a))\}}$$

Main Steps

- Standardize variables
  - Avoid confusion(step 2)
- Implication elimination
  - Remove $\Rightarrow$(step 3)
- Negation inwards(step 4)
  - $\neg \forall x \;\; p$ becomes $\exists x \;\; \neg p$
  - $\neg \exists x \;\; p$ becomes $\forall x \;\; \neg p$
- Skolemization
  - Remove $\exists$ quantifier(step 6)
- Distribute $\wedge$ over $\vee$
  - Flatten out nested conjunction and disjunction(step 8)
- Universal implicit
  - Remove $\forall$ quantifier(step 8)

Example

- $\forall x, y \;\; P(x, y) \Rightarrow \exists z \;\; Q(x, z) \wedge \neg R(y, z)$
  - Standardize variables
    - None
  - Implication elimination
    - $\forall x, y \;\; \neg P(x, y) \vee \exists z \;\; Q(x, z) \wedge \neg R(y, z)$
  - Negation inwards
    - None
  - Skolemization
    - $\forall x, y \;\; \neg P(x, y) \vee Q(x, f(x, y)) \wedge \neg R(y, f(x, y))$
  - Distribute $\wedge$ over $\vee$
    - $\forall x, y \;\; (\neg P(x, y) \vee Q(x, f(x, y))) \wedge (\neg P(x, y) \vee \neg R(y, f(x, y)))$
  - Universal implicit
    - $(\neg P(x, y) \vee Q(x, f(x, y))) \wedge (\neg P(x, y) \vee \neg R(y, f(x, y)))$

Resolution Refutation Proof

- In refutation proofs, we add the negation of the goal to the set of clauses and then attempt to deduce **FALSE**
  - Convert the set of rules and facts into clause form(conjunction of clauses)
  - Insert the negation of the goal as another clause
  - Use resolution to deduce a refutation

Example 1

- Thomas, Harry and Kate are students of computer science department
- Every student is either a programmer or is a good analyst, or both
- No programmer likes concepts and all analyst students like modeling
- Kate dislikes whatever Thomas likes and likes whatever Thomas dislikes
- Kate likes concepts and modeling
- Is there a student who is good at programming but not at modeling?

FOL formulation and Clause Forms - 1

- Thomas, Harry and Kate are students of computer science department
  - C1:Student(Thomas)
  - C2:Student(Harry)
  - C3:Student(Kate)
- Every student is either a programmer or is a good analyst, or both
  - $\forall x \;\; Student(x) \Rightarrow Programmer(x) \vee Analyst(x)$
  - C4: $\neg Student(x) \vee Programmer(x) \vee Analyst(x)$
- No programmer likes concepts and all analyst students like modeling
  - $\forall x \;\; Programmer(x) \Rightarrow \neg Likes(x, Concepts)$
  - $\forall x \;\; Analyst(x) \Rightarrow Likes(x, Modeling)$
  - C5: $\neg Programmer(x) \vee \neg Likes(x, Concepts)$
  - C6: $\neg Analyst(x) \vee Likes(x, Modeling)$

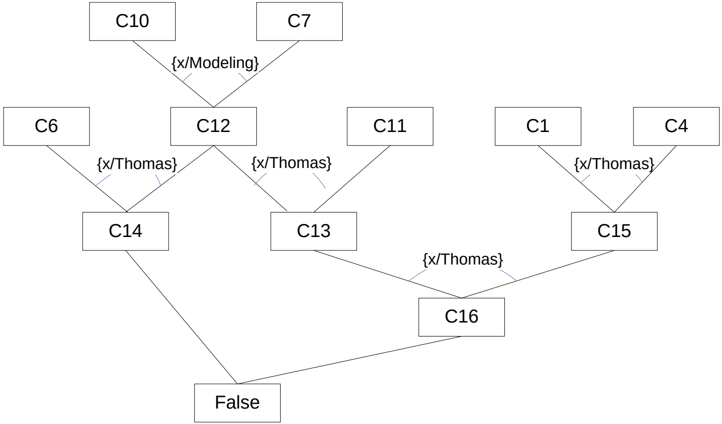
FOL formulation and Clause Forms - 2

- Kate dislikes whatever Thomas likes and likes whatever Thomas dislikes
  - $\forall x \;\; Likes(Thomas, x) \Leftrightarrow \neg Likes(Kate, x)$
  - C7: $\neg Likes(Thomas, x) \vee \neg Likes(Kate, x)$
  - C8: $Likes(Kate, x) \vee Likes(Thomas, x)$
- Kate likes concepts and modeling
  - C9: $Likes(Kate, Concepts)$
  - C10: $Likes(Kate, Modeling)$
- Is there a student who is good at programming but not at modeling?
  - Goal: $\exists x \;\; Programmer(x) \wedge \neg Likes(x, Modeling)$
  - C11(=$\neg$Goal): $\forall x \;\; \neg Programmer(x) \vee Likes(x, Modeling)$

Inference - 1

- C12: $\neg Likes(Thomas, Modeling)$
- C13: $\neg Programmer(Thomas)$
- C14: $\neg Analyst(Thomas)$
- C15: $Programmer(Thomas) \lor Analyst(Thomas)$
- C16: $Analyst(Thomas)$
- C17: $False$

Inference - 2



C10
C7
{x/Modeling}
C6
C12
C11
C1
C4
{x/Thomas}
{x/Thomas}
{x/Thomas}
C14
C13
C15
{x/Thomas}
C16
False

Resolution Strategies - Unit Resolution

- Every resolution step must involve a unit clause
  - Like $C10$, $C9$,$C1$
- Leads to a good speedup
- Incomplete in general
- Complete for Horn knowledge bases

Resolution Strategies - Input Resolution

- Every resolution step must involve an input sentence(from the query or the knowledge base)
  - Add something from the knowledge base to deduce false
- Modus Ponens is a kind of input resolution strategy in Horn knowledge bases
- Incomplete in general
- Complete for Horn knowledge bases

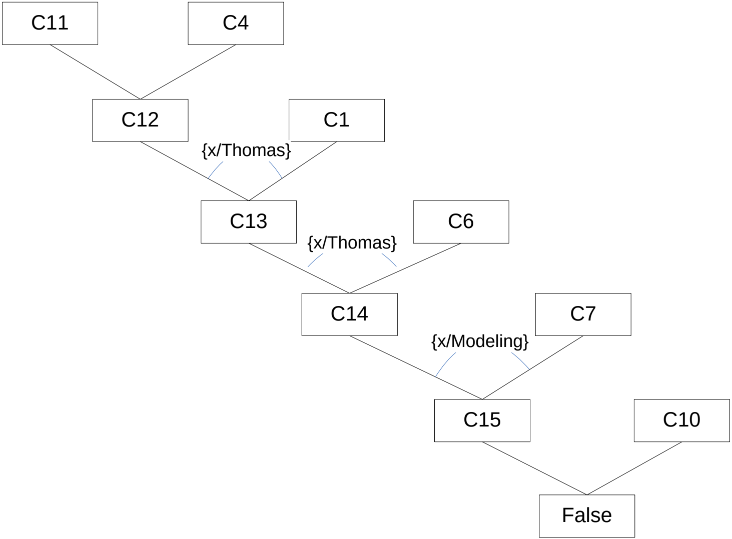
Resolution Strategies - Linear Resolution

- Slight generalization of input resolution
- Allow P and Q to be resolved together either if P is in the original knowledge, or if P is an ancestor of Q in the proof tree
  - Tree does not branch
- Complete for knowledge bases

Linear Resolution - 1

- C12: $\neg Student(x) \vee Analyst(x) \vee Likes(x, Modeling)$
- C13: $Analyst(Thomas) \vee Likes(Thomas, Modeling)$
- C14: $Likes(Thomas, Modeling)$
- C15: $\neg Kate(Kate, Modeling)$
- C16: $False$

Linear Resolution - 2



Summary

- Reduce FOL to propositional logic using EI and UI
- Forward chaining and backward chaining
- Generalized Modus Ponens
  - Unification
- Generalized Resolution rule
  - Conversion to normal form
- Resolution Strategies