Artificial Intelligence

For HEDSPI Project

Lecturer 11 – Inference in First Order Logic

Lecturers :
Dr. **Le Thanh Huong**
Dr. Tran Duc Khanh
Dr. Hai V. Pham
HUST

First Order Logic

- Syntax
- Semantic
- Inference
  - Resolution

Inference in FOL

- Difficulties
  - Quantifiers
  - Infinite sets of terms
  - Infinite sets of sentences
- Examples: $\forall x.King(x) \wedge Greedy(x) \Rightarrow Evil(x)$
  - Infinite set of instances

$$King(Bill) \wedge Greedy(Bill) \Rightarrow Evil(Bill)$$
$$King(FatherOf(Bill)) \wedge Greedy(FatherOf(Bill)) \Rightarrow Evil(FatherOf(Bill))$$
$$\ldots$$

Robinson's 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

Clause

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

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

  - Example

$$P(x) \vee Q(x, a) \vee R(b)$$
$$P(y) \vee \neg Q(b, y) \vee R(y)$$

- The empty clause corresponds to a contradiction
- Any sentence can be transformed to an equi-satisfiable set of clauses

Elements of Resolution

- Resolution rule
- Unification
- Transform a sentence to a set of clauses

Resolution rule

- Resolution rule

$$\frac{A \vee B \qquad \neg C \vee D}{\theta(A \vee D)} \qquad \theta = mgu(B, C)$$

  - mgu: most general unifier
    - The most general assignment of variables to terms in such a way that two terms are equal
    - Syntactical unification algorithm
  - θ: substitution

Example of Resolution rule

- x, y are variables
- a, b are constants

$$\frac{P(x) \vee Q(x,a) \qquad \neg Q(b,y) \vee R(y)}{P(b) \vee R(a)} \qquad \theta = \{x = b, y = a\}$$

$$A \equiv P(x)$$

$$B \equiv Q(x,a)$$

$$C \equiv Q(b,y)$$

$$D \equiv R(y)$$

Example of Resolution rule

$$\frac{\neg Pet(Joe) \vee Cat(Joe) \vee Bird(Joe) \qquad Parrot(x) \vee \neg Bird(x)}{\neg Pet(Joe) \vee Cat(Joe) \vee Parrot(Joe)} \quad (1)$$

(1) $mgu(Bird(x), Bird(Joe)) = \{x/Joe\}$

$$\frac{\neg On(x,y) \vee Above(x,y) \qquad On(B,A) \vee On(A,B)}{Above(A,B) \vee On(B,A)} \quad (2)$$

(2) $mgu(On(x,y), On(A,B)) = \{x/A,\ y/B\}$

$$\frac{\neg Bird(x) \vee Feathers(x) \qquad \neg Feathers(y) \vee Flies(y)}{\neg Bird(x) \vee Flies(x)} \quad (3)$$

(3) $mgu(Feathers(x), Feathers(y)) = \{y/x\}$

Elements of Resolution

- Resolution rule
- Unification
- Transform a sentence to a set of clauses

Unification

- Input
  - Set of equalities between two terms
- Output
  - Most general assignment of variables that satisfies all equalities
  - Fail if no such assignment exists

Unification algorithm

Decompose

$$U \cup \{f(t_1, \ldots, t_n) =^? f(s_1, \ldots, s_n)\} \longrightarrow U \cup \{t_1 =^? s_1, \ldots, t_n =^? s_n\}$$

Orient.

$$U \cup \{t =^? v\} \longrightarrow U \cup \{v =^? t\}$$

Delete.

$$U \cup \{v =^? v\} \longrightarrow U$$

- Vars(U), Vars(t) are sets of variables in U and t
- v is a variable
- s and t are terms
- f and g are function symbols

Eliminate.

$$U \cup \{v =^? t\},\ v \in \mathcal{V}ars(U) \setminus \mathcal{V}ars(t) \longrightarrow U[v/t] \cup \{v =^? t\}$$

Mismatch.

$$U \cup \{f(t_1, \ldots, t_m) =^? g(s_1, \ldots, s_n)\},\ f, g \text{ distinct or } m \neq n \longrightarrow FAIL$$

Occurs.

$$U \cup \{v =^? t\},\ v \neq t \text{ but } v \in \mathcal{V}ars(t) \longrightarrow FAIL$$

Example of Unification

$$\{\underline{F(G(H(y)), H(A)) =^? F(G(x), x)}\} \xrightarrow{\text{Decompose}}$$

$$\{\underline{G(H(y)) =^? G(x)},\ H(A) =^? x\} \xrightarrow{\text{Decompose}}$$

$$\{\underline{H(y) =^? x},\ H(A) =^? x\} \xrightarrow{\text{Orient}}$$

$$\{\underline{x =^? H(y)},\ H(A) =^? x\} \xrightarrow{\text{Eliminate } x}$$

$$\{x =^? H(y),\ \underline{H(A) =^? H(y)}\} \xrightarrow{\text{Decompose}}$$

$$\{x =^? H(y),\ \underline{A =^? y}\} \xrightarrow{\text{Orient}}$$

$$\{x =^? H(y),\ \underline{y =^? A}\} \xrightarrow{\text{Eliminate } y}$$

$$\{x =^? H(A),\ y =^? A\}$$

Elements of Resolution

- Resolution rule
- Unification
- Transform a sentence to a set of clauses

Transform a sentence to a set of clauses

1. Eliminate implication
2. Move negation inward
3. Standardize variable scope
4. Move quantifiers outward
5. Skolemize existential quantifiers
6. Eliminate universal quantifiers
7. Distribute and, or
8. Flatten and, or
9. Eliminate and

Eliminate implication

$$\{\forall x\, (\forall y\, P(x,y)) \rightarrow \neg(\forall y\, Q(x,y) \rightarrow R(x,y))\}$$

$$\begin{array}{lcl} \alpha \rightarrow \beta & \longrightarrow & \neg\alpha \vee \beta \\ \alpha \leftrightarrow \beta & \longrightarrow & (\neg\alpha \vee \beta) \wedge (\neg\beta \vee \alpha) \end{array}$$

$$\{\forall x\, \neg(\forall y\, P(x,y)) \vee \neg(\forall y\, \neg Q(x,y) \vee R(x,y)))\}$$

Move negation inward

$$\{\forall x\, \neg(\forall y\, P(x,y)) \vee \neg(\forall y\, \neg Q(x,y) \vee R(x,y)))\}$$

$$\begin{array}{lclclcl} \neg\neg\alpha & \longrightarrow & \alpha & & \neg\forall v\, \alpha & \longrightarrow & \exists v\, \neg\alpha \\ \neg(\alpha \vee \beta) & \longrightarrow & \neg\alpha \wedge \neg\beta & & \neg\exists v\, \alpha & \longrightarrow & \forall v\, \neg\alpha \\ \neg(\alpha \wedge \beta) & \longrightarrow & \neg\alpha \vee \neg\beta & & & & \end{array}$$

$$\{\forall x\, (\exists y\, \neg P(x,y)) \vee (\exists y\, Q(x,y) \wedge \neg R(x,y))\}$$

Standardize variable scope

$$\{\forall x\ (\exists y\ \neg P(x,y)) \vee (\exists y\ Q(x,y) \wedge \neg R(x,y))\}$$

Each variable for each quantifier

$$\{\forall x\ (\exists y\ \neg P(x,y)) \vee (\exists z\ Q(x,z) \wedge \neg R(x,z))\}$$

Move quantifiers outward

$$\{\forall x\ (\exists y\ \neg P(x,y)) \vee (\exists z\ Q(x,z) \wedge \neg R(x,z))\}$$

$$\begin{array}{llllll} (Qx\ \alpha) \wedge \beta & \longrightarrow & Qx\ (\alpha \wedge \beta) & \alpha \wedge (Qx\ \beta) & \longrightarrow & Qx\ (\alpha \wedge \beta) \\ (Qx\ \alpha) \vee \beta & \longrightarrow & Qx\ (\alpha \vee \beta) & \alpha \vee (Qx\ \beta) & \longrightarrow & Qx\ (\alpha \vee \beta) \end{array}$$

$$\{\forall x\ \exists y\ \exists z\ \neg P(x,y) \vee (Q(x,z) \wedge \neg R(x,z))\}$$

Existential Instantiation

$$\{\forall x\, \exists y\, \exists z\, \neg P(x,y) \vee (Q(x,z) \wedge \neg R(x,z))\}$$

$$\frac{\exists v \;\; \alpha}{\textsc{Subst}(\{v/k\}, \alpha)}$$

{ ∀x ¬P(x,a) ∨ (Q(x,b) ∧ ¬R(x,b) }

Skolemize existential quantifiers

$$\{\forall x\, \exists y\, \exists z\, \neg P(x,y) \vee (Q(x,z) \wedge \neg R(x,z))\}$$

$$\exists v\, \alpha \quad \longrightarrow \quad \alpha[v/\pi(v_1, \ldots, v_n)]$$

with π *new* and $v_1, \ldots, v_n$ universally quantified outside $\exists v\, \alpha$

$$\{\forall x\, \neg P(x, F_1(x)) \vee (Q(x, F_2(x)) \wedge \neg R(x, F_2(x)))\}$$

Eliminate universal quantifiers

$$\{\forall x\ \neg P(x, F_1(x)) \vee (Q(x, F_2(x)) \wedge \neg R(x, F_2(x)))\}$$

$$\boxed{\forall v\ \alpha \quad \longrightarrow \quad \alpha}$$

$$\{\neg P(x, F_1(x)) \vee (Q(x, F_2(x)) \wedge \neg R(x, F_2(x)))\}$$

Distribute and, or

$$\{\neg P(x, F_1(x)) \vee (Q(x, F_2(x)) \wedge \neg R(x, F_2(x)))\}$$

$$\boxed{\begin{array}{lcl} \alpha \vee (\beta \wedge \gamma) & \longrightarrow & (\alpha \vee \beta) \wedge (\alpha \vee \gamma) \\ (\beta \wedge \gamma) \vee \alpha & \longrightarrow & (\beta \vee \alpha) \wedge (\gamma \vee \alpha) \end{array}}$$

$$\{(\neg P(x, F_1(x)) \vee Q(x, F_2(x))) \wedge (\neg P(x, F_1(x)) \vee \neg R(x, F_2(x)))\}$$

Flatten and, or

$$\{(\neg P(x, F_1(x)) \vee Q(x, F_2(x))) \wedge (\neg P(x, F_1(x)) \vee \neg R(x, F_2(x)))$$

$$\boxed{\begin{array}{lcl} (\alpha \wedge (\beta \wedge \gamma)) & \longrightarrow & (\alpha \wedge \beta \wedge \gamma) \\ (\alpha \vee (\beta \vee \gamma)) & \longrightarrow & (\alpha \vee \beta \vee \gamma) \\ ((\alpha \wedge \beta) \wedge \gamma) & \longrightarrow & (\alpha \wedge \beta \wedge \gamma) \\ ((\alpha \vee \beta) \vee \gamma) & \longrightarrow & (\alpha \vee \beta \vee \gamma) \end{array}}$$

$$\{(\neg P(x, F_1(x)) \vee Q(x, F_2(x))) \wedge (\neg P(x, F_1(x)) \vee \neg R(x, F_2(x)))\}$$

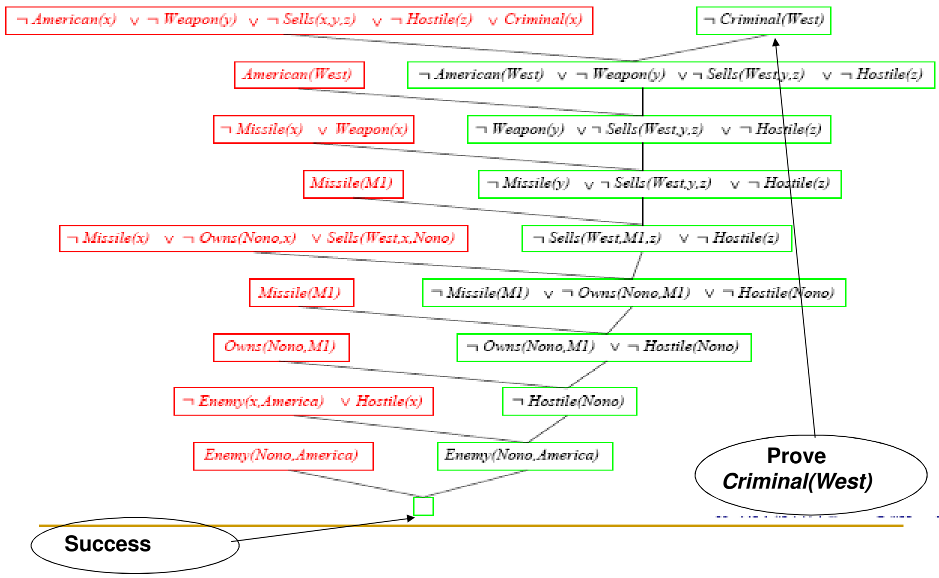
Eliminate and

$$\{(\neg P(x, F_1(x)) \vee Q(x, F_2(x))) \wedge (\neg P(x, F_1(x)) \vee \neg R(x, F_2(x)))\}$$

$$\boxed{\{\alpha \wedge \beta\} \quad \longrightarrow \quad \{\alpha,\ \beta\}}$$

$$\{\neg P(x, F_1(x)) \vee Q(x, F_2(x)),\ \neg P(x, F_1(x)) \vee \neg R(x, F_2(x))\}$$

Example of proof by Resolution



Summary of Resolution

- Refutation-based procedure
  - *S* |= *A* if and only if $S \cup \{\neg A\}$ is unsatisfiable
- 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* |= *A*
    - Otherwise
      - No conclusion

Summary of Resolution

- Theorem
  - A set of clauses S is unsatisfiable if and only if upon the input S, Resolution procedure finds the empty clause (after a finite time).

Exercice

- 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
- Is West a criminal?

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

Modeling

"... it is a crime for an American to sell weapons to hostile nations":

$$\forall x, y, z \;\; American(x) \wedge Weapon(y) \wedge Nation(z) \wedge Hostile(z) \wedge Sells(x, z, y) \Rightarrow Criminal(x)$$

"Nono ... has some missiles":

$$\exists x \;\; Owns(Nono, x) \wedge Missile(x)$$

"All of its missiles were sold to it by Colonel West":

$$\forall x \;\; Owns(Nono, x) \wedge Missile(x) \Rightarrow Sells(West, Nono, x)$$

We will also need to know that missiles are weapons:

$$\forall x \;\; Missile(x) \Rightarrow Weapon(x)$$

and that an enemy of America counts as "hostile":

$$\forall x \;\; Enemy(x, America) \Rightarrow Hostile(x)$$

"West, who is American ...":

$$American(West)$$

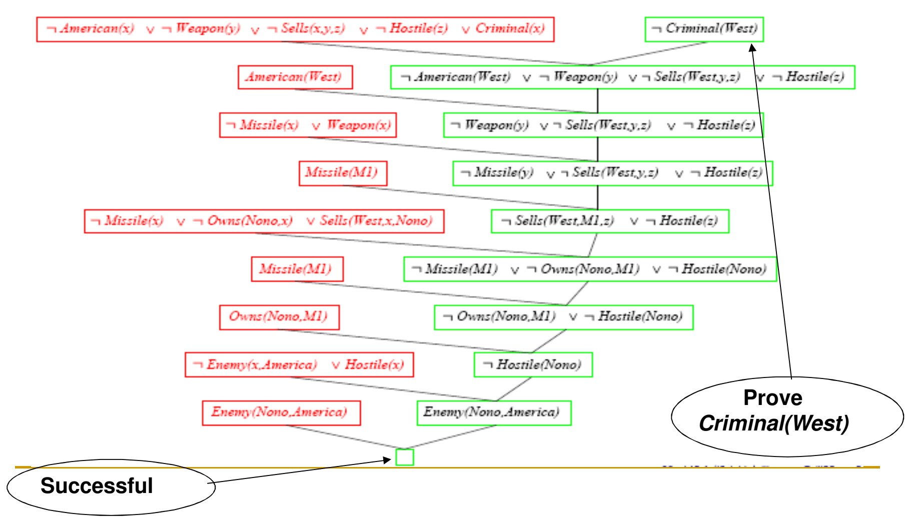
"The country Nono ...":

$$Nation(Nono)$$

"Nono, an enemy of America ...":

$$Enemy(Nono, America)$$
$$Nation(America)$$

Transform the problem to set of clauses and Resolution



Exercice

- Jack owns a dog   own(Jack, dog)
- Every dog owner is an animal lover
- No animal lover kills an animal
- Either Jack or Curiosity killed the cat, who is named Tuna
- Did Curiosity kill the cat?

Jack owns a dog   own(Jack, dog)
Every dog owner is an animal lover
No animal lover kills an animal
Either Jack or Curiosity killed the cat, who is named Tuna
Did Curiosity kill the cat? Kills(Curiosity,Tuna)

$$\exists x. Dog(x) \wedge Owns(Jack, x)$$
$$\forall x \forall y. (Dog(y) \wedge Owns(x, y)) \Rightarrow AnimalLover(x)$$
$$\forall x \forall y. (AnimalLover(x) \wedge Animal(y) \Rightarrow \neg Kills(x, y))$$
$$Kills(Jack, Tuna) \vee Kill(Curiosity, Tuna)$$
$$Cat(Tuna)$$
$$\forall x. Cat(x) \Rightarrow Animal(x)$$

Transform the problem to set of clauses

$Dog(D)$

$Owns(Jack, D)$

$\neg Dog(y) \vee \neg Owns(x, y) \vee AnimalLover(x)$

$\neg AnimalLover(x) \wedge \neg Animal(y) \vee \neg Kills(x, y)$

$Kills(Jack, Tuna) \vee Kill(Curiosity, Tuna)$

$Cat(Tuna)$

$\neg Cat(x) \vee Animal(x)$

¬Kills(Curiosity,Tuna)