Posts and Telecommunication Institute of Technology
Faculty of Information Technology 1

Introduction to Artificial Intelligence

Predicate logic
(First-order logic)

Dao Thi Thuy Quynh

Outline

- Predicate logic
- Inference in predicate logic

Outline

- Predicate logic
  - Characteristics
  - Syntax
  - Semantics
- Inference in predicate logic

Characteristics of predicate logic

- Propositional logic
  - Representations are limited to the world of events

- Predicate logic
  - Allows to describe the world with objects, object properties, and relationships between objects
  - Objects: a table, a tree, a person, a house, a number, …
  - Properties: a table with four legs, made of wood, with drawers,…
  - Relationships: brothers, friends (between people), inside, outside, above, below (between things),…
  - Function: a particular instance of relations, for each input we have a unique function value

Syntax of predicate logic (1/4)

- Symbols
  - Constants: a, b, c, An, Ba, John, …
  - Variables: x, y, z, u, v, w, …
  - Predicates: P, Q, R, S, Like, Friend, …
    - Each predicate has n variables ($n \geq 0$)
    - Non-variable predicates are propositional symbols
  - Functions: f, g, cos, sin, mother, husband, …
    - Each function has n variables ($n \geq 0$)
  - Logical connectives: $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), $\Rightarrow$ (implication), $\Leftrightarrow$ (equivalence)
  - Quantifier symbols: $\forall$ (universal quantifier) , $\exists$ (existential quantifier)
  - Separation symbols: comma, parentheses

Syntax of predicate logic (2/4)

- Terms
  - Are expressions that describe objects, recursively defined as follows
    - Constants and variables are terms
    - If $t_1, t_2, \dots, t_n$ are terms and f denotes a function with n variables, then $f(t_1, t_2, \dots, t_n)$ is also a term
  - Non-variable terms are called ground terms
  - Two terms are equal if they correspond to the same object
    - Father(John) = Mike
- Atomic formulas (atoms)
  - Representations of an object's properties, or relationships between objects, is determined recursively as follows
    - Non-variable predicate symbols (propositions) are atomic formulas
    - If $t_1, t_2, \dots, t_n$ are terms and P denotes a predicate with n variables, $P(t_1, t_2, \dots, t_n)$ is also an atomic formula

Syntax of predicate logic (3/4)

- Formulas
  - Recursively constructed from atoms, using logical connectives and quantifiers as follows
    - Atoms are formulas
    - If G and H are formulas, the following expressions are formulas
      - $(G \wedge H)$, $(G \vee H)$, $(\neg G)$, $(G \Rightarrow H)$, $(G \Leftrightarrow H)$
    - If G is a formula and x denotes a variable, the following expressions are formulas
      - $(\forall x G)$, $(\exists x G)$
- Conventions
  - Non-atomic formulas are called complex formulas
  - Non-variable formulas are called specific formulas
  - When writing formulas, we can remove unnecessary parentheses

Syntax of predicate logic (4/4)

- Universal quantifier (∀)
  - Describe the properties of an entire class of objects, without enumerating the objects
  - $\forall x(Elephant(x) \Rightarrow Color(x, Gray))$
- Existential quantifier (∃)
  - Allows creating a sentence referring to a certain object in a class of objects, having properties or satisfying a certain relationship
  - $\exists x(Student(x) \land Inside(x, P301))$
- Literals
  - Are atoms or negation of atoms
  - $Play(x, Football)$, $\neg Like(Lan, Rose)$
- Clauses
  - Disjunction of literals
  - $Male(x) \lor \neg Like(x, Football)$

Semantics of predicate logic (1/3)

- Interpretation
  - A specification of enough information to determine whether that formula is true or false
  - The meaning of the formula in a certain world
- Semantics of atoms
  - In an interpretation, an atom corresponds to a specific event (True or False)
    - $Student(Lan)$
- Semantics of complex formulas
  - Are determined based on semantics of atoms and logical connectives
    - $Student(Lan) \land Like(An, Rose)$
    - $Like(An, Rose) \lor \neg Like(An, Tulip)$

Semantics of predicate logic (2/3)

- Semantics of formulas with quantifiers
  - Formula $\forall xG$ is true if and only if every formula obtained from G by replacing x by an object in the object domain is true
    - Example: For object domain {An, Ba, Lan}, the semantics of formula $\forall xStudent(x)$ is the semantics of the following formula
    - $Student(An) \land Student(Ba) \land Student(Lan)$
  - Formula $\exists xG$ is true if and only if there exists a formula obtained from G by replacing x by an object in the object domain is true
    - Example: the semantics of formula $\exists xStudent(x)$ is the semantics of the following formula
    - $Student(An) \lor Student(Ba) \lor Student(Lan)$
- Satisfiable formulas, unsatisfiable formulas, valid formulas, models: the same as in propositional logic

Semantics of predicate logic (3/3)

- Nested quantifiers
  - Can use multiple quantifiers in complex formulas

$$\forall x \forall y Sibling(x, y) \Rightarrow Relationship(x, y)$$
$$\forall x \exists y Love(x, y)$$

  - Multiple quantifiers of the same type can be grouped by usinig a quantifier symbol

$$\forall x, y Sibling(x, y) \Rightarrow Relationship(x, y)$$

  - Do not change quantifiers of different types

$\forall x \exists y Love(x, y)$ Everyone has someone to love

$\exists y \forall x Love(x, y)$ There's someone everyone loves

Logical equivalences

1. $\forall x G(x) \equiv \forall y G(y)$
2. $\exists x G(x) \equiv \exists y G(y)$
3. $\neg\big(\forall x G(x)\big) \equiv \exists x\big(\neg G(x)\big)$
4. $\neg\big(\exists x G(x)\big) \equiv \forall x\big(\neg G(x)\big)$
5. $\forall x(G(x) \wedge H(x)) \equiv \forall x G(x) \wedge \forall x H(x)$
6. $\exists x(G(x) \vee H(x)) \equiv \exists x G(x) \vee \exists x H(x)$

Examples (1/2)

- Translate the following sentences into predicate logic
  1. An không cao
  2. An và Ba là anh em
  3. Tất cả nhà nông đều thích mặt trời
  4. Mọi cây nấm đỏ đều có độc
  5. Không có nấm đỏ nào độc cả
  6. Chỉ có đúng 2 nấm đỏ
  7. Một số học sinh vượt qua kỳ thi
  8. Tất cả học sinh đều vượt qua kỳ thi trừ một bạn
  9. Hai anh em phải cùng cha cùng mẹ

Examples (2/2)

- Sentences in predicate logic

1. $\neg Tall(An)$
2. $Sibling(An, Ba)$
3. $\forall x(Farmer(x) \Rightarrow LikeSun(x))$
4. $\forall x(Mushroom(x) \land Red(x) \Rightarrow Poisonous(x))$
5. $\forall x(Mushroom(x) \land Red(x) \Rightarrow \neg Poisonous(x))$
6. $\exists x, y(Mushroom(x) \land Red(x) \land Mushroom(y) \land Red(y) \land (x \neq y) \land \forall z(Mushroom(z) \land Red(z) \Rightarrow (z = x) \lor (z = y)))$
7. $\exists x(Student(x) \land Pass(x))$
8. $\exists x((Student(x) \land \neg Pass(x)) \land \forall y(Student(y) \land (y \neq x) \Rightarrow Pass(y)))$
9. $\forall x, y(Sibling(x, y) \Rightarrow \exists p, q(Father(p, x) \land Father(p, y) \land Mother(q, x) \land Mother(q, y))$

Outline

- Predicate logic
- Inference in predicate logic
  - Inference rules
  - Forward and backward chaining
  - Reasoning using resolutions

Inference rules (1/5)

- Reasoning with predicate logic is more difficult than propositional logic because variables can take on infinite values
  - Cannot use truth tables
- The inference rules for propositional logic are also true for predicate logic
  - Modus Ponens, Modus Tollens, duouble-negation elimination, and-introduction, or-introduction, and-elimination, or-elimination, resolution
- Besides:
  - There are some inference rules with quantifiers

Inference rules (2/5)

- Substitution
  - **Notation**: $SUBST(\theta, \alpha)$
  - **Meaning**: substitute value θ into formula α
  - Example
    - $SUBST\ (\{x/Nam, y/An\}, Like(x,y))\ = Like(Nam, An)$

- Universal elimination

$$\frac{\forall x\ \alpha}{SUBST(\{x/g\}, \alpha)}$$

Example:

$$\forall x\ Like(x, IceCream) \xrightarrow{\{x/Nam\}} Like(Nam, IceCream)$$

Inference rules (3/5)

- Existential elimination

$$\frac{\exists x\ \alpha}{SUBST(\{x/k\}, \alpha)}$$ k has not appeared in KB

Example:

$$\exists x\ GoodAtMath(x) \xrightarrow{\{x/C\}} GoodAtMath(C)$$

k is called a Skolem constant and can be given a name

- Existential introduction

$$\frac{\alpha}{\exists x\ SUBST(\{g/x\}, \alpha)}$$

Example:

$$Like(Nam, IceCream) \xrightarrow{\{Nam/x\}} \exists x\ Like(x, IceCream)$$

Example of reasoning (1/3)

- Problem

Bob là trâu		
Pat là lợn		
Trâu to hơn lợn		
Bob to hơn Pat?		

Example of reasoning (2/3)

- Problem

Bob là trâu	$Buffalo(Bob)$	(1)
Pat là lợn	$Pig(Pat)$	(2)
Trâu to hơn lợn	$\forall x, y\ Buffalo(x) \wedge Pig(y) \Rightarrow Bigger(x, y)$	(3)
Bob to hơn Pat?	$Bigger(Bob, Pat)?$	

Example of reasoning (3/3)

- Problem

Bob là trâu	$Buffalo(Bob)$	(1)
Pat là lợn	$Pig(Pat)$	(2)
Trâu to hơn lợn	$\forall x, y\ Buffalo(x) \land Pig(y) \Rightarrow Bigger(x, y)$	(3)
Bob to hơn Pat?	$Bigger(Bob, Pat)?$	

- Reasoning

And-Introduction (1)(2)	$Buffalo(Bob) \land Pig(Pat)$	(4)
Universal elimination (3)	$Buffalo(Bob) \land Pig(Pat) \Rightarrow Bigger(Bob, Pat)$	(5)
Modus Ponens, (4)(5)	$Bigger(Bob, Pat)$	

Inference rules (4/5)

- Unification
  - Unification is a process of making two different atoms identical by finding a substitution
  - **Notation**: $UNIFY(p, q) = (\theta)$

    $SUBST(\theta, p) = SUBST(\theta, q)$

    θ is called unifier
  - If there are multiple unifiers, we often use the most general unifier (MGU), the unifier that uses the fewest substitutions for variables
  - Unification can be performed automatically using an algorithm of linear complexity (in term of number of variables)

Unification example

p	q	θ
$Know(Nam, x)$	$Know(Nam, Bắc)$	$\{x/Bắc\}$
$Know(Nam, x)$	$Know(y, MotherOf(y))$	$\{y/Nam, x/MotherOf(Nam)\}$
$Know(Nam, x)$	$Know(y, z)$	$\{y/Nam, x/z\}$ $\{y/Nam, x/Nam, z/Nam\}$

Inference rules (5/5)

- General Modus Ponens (GMP)
  - Suppose that we have atoms p_i, p'_i, q, and substitution θ such that $UNIFY(p_i, p'_i) = \theta$, for all i
  - Then:

$$\frac{p'_1, p'_2, \dots, p'_n, (p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q)}{SUBST(\theta, q)}$$

- GMP allows us to build automatic reasoning algorithms, forward and backward chaining

Forward chaining (1/4)

- When new formula p is added to KB:
  - For each rule q that p can be unified with a left-hand side part:
    - If the remaining parts of the left hand side already exist, add the right-hand side to KB and continue

Forward chaining (2/4)

- When new formula p is added to KB:
  - For each rule q that p can be unified with a left-hand side part:
    - If the remaining parts of the left hand side already exist, add the right-hand side to KB and continue
- Example

Cho KB như sau:

1. Mèo thích cá
2. Mèo ăn gì nó thích
3. Có con mèo tên là Tom

Hỏi: Tom có ăn cá không?

Forward chaining (3/4)

- When new formula p is added to KB:
  - For each rule q that p can be unified with a left-hand side part:
    - If the remaining parts of the left hand side already exist, add the right-hand side to KB and continue
- Example

Cho KB như sau:

1. Mèo thích cá
2. Mèo ăn gì nó thích
3. Có con mèo tên là Tom

Hỏi: Tom có ăn cá không?

Translate into predicate logic:

1. $\forall x\ Cat(x) \Rightarrow Like(x, Fish)$
2. $\forall x, y\ Cat(x) \wedge Like(x, y) \Rightarrow Eat(x, y)$
3. $Cat(Tom)$

Hỏi: $Eat(Tom, Fish)$?

Forward chaining (4/4)

- When new formula p is added to KB:
  - For each rule q that p can be unified with a left-hand side part:
    - If the remaining parts of the left hand side already exist, add the right-hand side to KB and continue
- Example

Cho KB như sau:
1. Mèo thích cá
2. Mèo ăn gì nó thích
3. Có con mèo tên là Tom

Hỏi: Tom có ăn cá không?

Translate into predicate logic:
1. $\forall x\ Cat(x) \Rightarrow Like(x, Fish)$
2. $\forall x, y\ Cat(x) \land Like(x, y) \Rightarrow Eat(x, y)$
3. $Cat(Tom)$

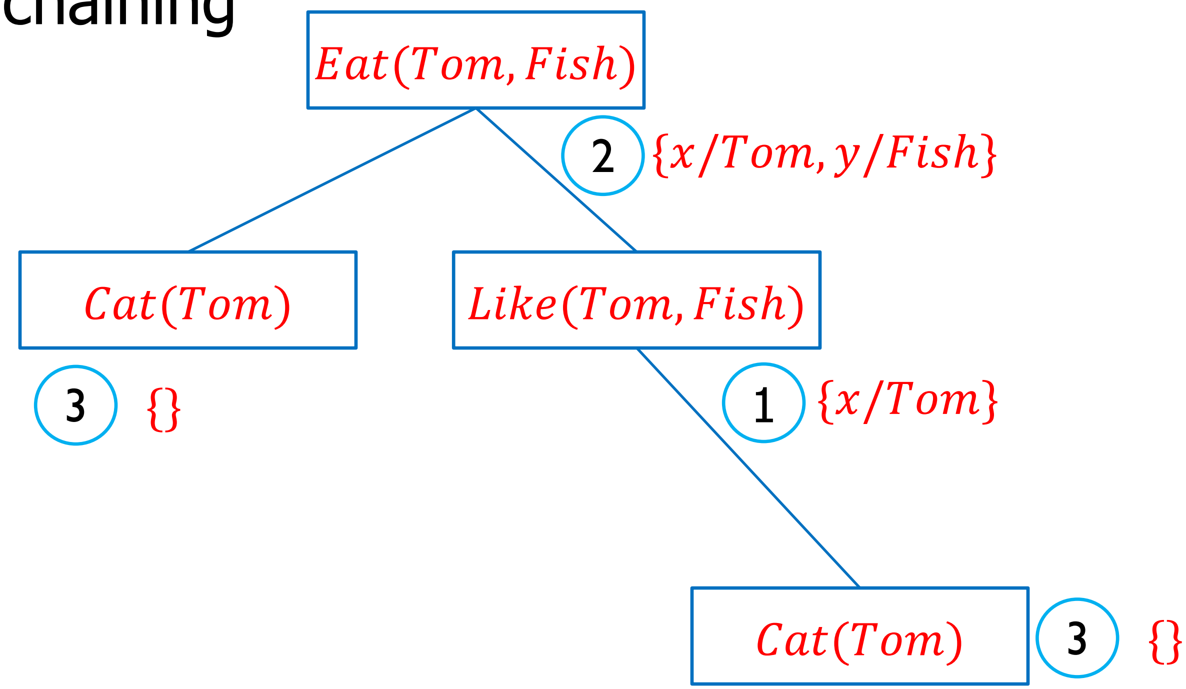
Hỏi: $Eat(Tom, Fish)$?

Reasoning:

4. GMP (1) (3) $\Rightarrow Like(Tom, Fish)$
5. GMP (2) (3) (4) $\Rightarrow Eat(Tom, Fish)$

Backward chaining

- For question q, if exists q' that can be unified with q, return unifier
- For each rule with the right-hand side q' that can be unified with q, try to prove the left-hand side parts using backward chaining



Reasoning using resolutions

- Resolution in predicate logic
  - Given two formulas, where P_i and Q_i are literals
    - $P_1 \vee P_2 \vee \ \dots \vee P_n$
    - $Q_1 \vee Q_2 \vee \ \dots \vee Q_m$
  - If P_j and $\neg Q_k$ can be unified by θ, we have the following resolution

$$\frac{P_1 \vee P_2 \vee \ \dots \vee P_n, Q_1 \vee Q_2 \vee \ \dots \vee Q_m}{SUBST(\theta, P_1 \vee \ \dots \vee P_{j-1} \vee P_{j+1} \vee \ \dots \vee \ P_n \vee Q_1 \vee \ \dots \vee Q_{k-1} \vee Q_{k+1} \vee \ \dots \vee Q_m)}$$

Example:

$$\frac{\forall x\big(Rich(x) \vee Good(x)\big), \neg Good(Nam) \vee Handsome(Nam)}{Rich(Nam) \vee Handsome(Nam)}$$

Resolution and Reductio ad absurdum (1/4)

- Need to prove $KB \vdash Q$?
- Method:
  - Add $\neg Q$ to KB, prove that there exists a subset of KB that has False value
  - $(KB \vdash Q) \Leftrightarrow (KB \wedge \neg Q \vdash False)$

Resolution and Reductio ad absurdum (2/4)

- Algorithm
  - $KB = UNION\ (KB, \neg Q)$
  - **while** (KB does not contain False) **do**
    - 1. Select two formulas S_1, S_2 from KB so that we can apply resolution
      - Add the result of the resolution to KB
    - 2. If does not exist two such formulas
      - **return** False
  - **end while**
  - **return** Success

Resolution and Reductio ad absurdum (3/4)

- Example

KB:

$\neg A \vee \neg B \vee P$ (1)

$\neg C \vee \neg D \vee P$ (2)

$\neg E \vee C$ (3)

A (4)

E (5)

D (6)

Prove that: $KB \vdash P$

Resolution and Reductio ad absurdum (4/4)

- Example

KB:
$\neg A \vee \neg B \vee P$ (1)
$\neg C \vee \neg D \vee P$ (2)
$\neg E \vee C$ (3)
A (4)
E (5)
D (6)
Prove that: $KB \vdash P$

Proof:
Add to KB the following formula:
$\neg P$ (7)
Apply resolution to formulas (2) and (7)
$\neg C \vee \neg D$ (8)
Apply resolution to formulas (6) and (8)
$\neg C$ (9)
Apply resolution to formulas (3) and (9)
$\neg E$ (10)
Formula (10) has False value.
Conclusion: $KB \vdash P$

Conjunctive Normal Form (CNF) and Clause Form

- A clause is the disjunction of literals

  $A_1 \vee A_2 \vee \ \ldots \vee A_m$, where A_i are literals

- Conjunctive Normal Form (CNF): conjunction of clauses
  - $A \wedge (B \vee C) \wedge (D \vee E \vee F)$

- We can convert any formula into an equivalent formula that is in CNF

Conversion into CNF and Clause form (1/3)

- Step 1: Eliminate equivalences
  - Replace $P \Leftrightarrow Q$ by $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$
- Step 2: Eliminate implications
  - Replace $P \Rightarrow Q$ by $\neg P \vee Q$
- Step 3: Move negations close to predicates
  - Apply De Morgan's laws and replace $\neg(\neg A)$ by A:

$$\neg(\neg P) \equiv P$$

$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$

$$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$$

$$\neg(\forall x Q) \equiv \exists x(\neg Q)$$

$$\neg(\exists x Q) \equiv \forall x(\neg Q)$$

Conversion into CNF and Clause form (2/3)

- Step 4: Standardize variable names so that each quantifier has its own variable
  - Example

$$\boxed{\begin{array}{l}\forall x \neg P(x) \vee Q(x)\\ \forall x \neg R(x) \vee Q(x)\end{array}} \Rightarrow \boxed{\begin{array}{l}\forall x \neg P(x) \vee Q(x)\\ \forall \boldsymbol{y} \neg \boldsymbol{R}(\boldsymbol{y}) \vee \boldsymbol{Q}(\boldsymbol{y})\end{array}}$$

- Step 5: Eliminate existential quantifiers by using Skolem constants and Skolem functions
  - Replace $\exists x\, P(x)$ by $P(C)$, where C is a new constant (Skolem)
  - If $\exists$ is inside $\forall$, replace by a new function whose variable is a variable of $\forall$ (Skolem function)
  - Example:

  Replace $\forall x\, \exists y\, P(x, y)$ by $\forall x\, P(x, f(x))$, $f(x)$ is a Skolem function

Conversion into CNF and Clause form (3/3)

- Step 6: Eliminate universal quantifiers (∀)
  - Move universal quantifiers to the left-hand side and remove them
  - Example: transform $\forall x\,(P(x,y)\ \vee\ Q(x))$ into $(x,y)\ \vee\ Q(x)$
- Step 7: Apply distributive laws
  - $(P \wedge Q) \vee R\ \equiv\ (P \vee R)\ \wedge (Q \vee R)$
  - $(P \vee Q) \vee R\ \equiv\ (P \vee Q \vee R)$
- Step 8: Eliminate conjunctions
  - Eliminate conjunctions to from clauses
  - Example: transform $(P \vee R \vee S) \wedge (Q \vee \neg R)$ into two formulas: 1) $P \vee R \vee S$ 2) $Q \vee \neg R$
- Step 9: Standardize variable names so that each formula has its own variables

Exercise 1

- Cho các câu sau

1. Mọi bé trai đều thích chơi bóng đá
2. Ai thích chơi bóng đá đều có giày đá bóng
3. Nam là một bé trai

Câu hỏi

a) Biểu diễn các câu trên ở dạng logic vị từ
b) Chuyển các câu logic vị từ vừa viết về dạng chuẩn tắc hội
c) Viết câu truy vấn "Nam có giày đá bóng" dưới dạng logic vị từ và chứng minh sử dụng phép giải

Exercise 2

- Giả sử ta biết các thông tin sau

1. Ông Ba nuôi một con chó
2. Hoặc ông Ba hoặc ông Am đã giết con mèo Bibi
3. Mọi người nuôi chó đều yêu động vật
4. Ai yêu quý động vật cũng không giết động vật
5. Chó mèo đều là động vật

Hỏi ai đã giết con mèo Bibi?