1.4 Predicates and Quantifiers

Wenjing Li

wjli@bupt.edu.cn

School of Computer Science

Beijing University of Posts & Telecommunications

Outline

- **Predicate logic**（谓词逻辑）
- **Quantifiers** （量词）
  - Universal （全称量词）
  - Existential（存在量词）
  - Unique Existential（唯一存在量词）
- **Property of Quantifiers**
  - **Free and Bound Variables**（自由变项与绑定变项）
  - **Nesting**（嵌套）
  - **Negation**（否定）
- **Applications**

苏格拉底三段论

- 凡人都是要死的。苏格拉底是人。所以，苏格拉底是要死的。
  - 设：p:凡人都是要死的; q:苏格拉底是人; r: 苏格拉底是要死的。
  - 前提：p，q，结论：r
- 推理的形式结构：$p \wedge q \rightarrow r$ （非永真式！）

- **重新符号化：*x, a, F( ) , ∀, ∃***
  - 设：$F(x)$：x是人。$G(x)$：x是要死的。a：苏格拉底。
  - 前提：$\forall x(F(x) \rightarrow G(x))$，$F(a)$
  - 结论：$G(a)$

Predicates Logic

- A generalization of propositions - *predicates* (谓词) or *propositional functions* (命题函数) : propositions which contain variables
- Predicates become propositions once every variable is *bound* by
  - assigning it a value or an *object* (个体) from the *Universe of Discourse* (论域) U

  or
  - *quantifying* it

Formulas of Predicate Logic

- **Notation:**
  - We will use various kinds of ***individual constants*** that denote individuals/objects: ***a,b,c,…***, ***individual variables*** over objects: ***x, y, z , …***
  - ***P:*** refers to a **property** that the subject of the statement can have.
- The result of applying a ***predicate P*** to a *constant* ***a*** is the proposition ***P(a)***. Meaning: the object denoted by ***a*** has the property denoted by ***P***.
- The result of applying a ***predicate P*** to a *variable x* is the *propositional* ***form*** *P*(*x*).
  - *E.g.* if *P* = "is a prime number", then *P*(*x*) is the *propositional form* of "*x* is a prime number".

Formulas of Predicate Logic

- **Example:**
  - Let U = Z, the integers = {. . . -2, -1, 0 , 1, 2, . . .}
  - $P(x)$: $x > 0$ is the predicate. It has no truth value until the variable x is bound.
- **Examples of propositions where *x* is assigned a value:**
  - $P(-3)$ is false,
  - $P(0)$ is false,
  - $P(3)$ is true.

  The collection of integers for which $P(x)$ is true are the positive integers.
  - $P(y) \vee \sim P(0)$ is not a proposition. The variable y has not been bound. However.
  - $P(3) \vee \sim P(0)$ is a proposition which is true.

Propositional Functions

- Predicate logic generalizes the grammatical notion of a predicate to also include **propositional functions** of **any** number of arguments, each of which may take **any** grammatical role that a noun can take.
- *E.g.*
  - let *P(x,y,z)* = *"x gave y the grade z"*,
  - *then if x="Mike", y="Mary", z="A"*,
  - *then P(x,y,z) = "Mike gave Mary the grade A."*

Proposition

Proposition variable

Predicate（Propositional function）

N-place predicate

- A statement involving the n variables x_1, x_2 ,…x_n can be denoted by $P(x_1, x_2 ,\ldots x_n)$.
- A statement of the form $P(x_1, x_2 ,\ldots x_n)$ is the value of the propositional function P at the n-tuple $(x_1, x_2 ,\ldots x_n)$, and P is also called a ***n*-place predicate** or a ***n*-ary predicate**.

- Let R be the three-variable predicate
  - $R(x, y, z)$: $x + y = z$
- Find the truth value of
  - $R(2, -1, 5)$
  - $R(3, 4, 7)$
  - $R(x, 3, z)$

Quantifiers – Universal(全称量词)

- $P(x)$ is true *for every x* in the **universe of discourse**.
- Notation: *universal quantifier*

$$\forall x P(x)$$

'For all x, $P(x)$', 'For every x, $P(x)$'

- The variable x is bound by the universal quantifier producing a proposition.

- Let $P(x)$ be the statement "$x+1>x$".
- What is the truth value of the quantification $\forall x P(x)$, where the universe of discourse consists of all real numbers?

Quantifiers – Universal(全称量词)

- **Example:**

  U={1, 2, 3}

  $\forall x P(x) \Leftrightarrow P(1) \wedge P(2) \wedge P(3)$

- **Example:**
  - Let $P(x)$ be the statement "$x^2>0$".
  - What is the truth value of the quantification $\forall x P(x)$, where the universe of discourse consists of all integers?

Quantifiers – Existential(存在量词)

- $P(x)$ is true for *some* x in the universe of discourse.
- Notation: *existential quantifier*

$$\exists x P(x)$$

'There is an x such that $P(x)$', 'For some x, $P(x)$', 'For at least one x, $P(x)$', 'I can find an x such that $P(x)$.'

- Let $P(x)$ be the statement "$x>3$".
- What is the truth value of the quantification $\exists x P(x)$, where the universe of discourse consists of all real numbers?

Quantifiers – Existential(存在量词)

- **Example:**

  $U=\{1, 2, 3\}$

  $\exists x P(x) \Leftrightarrow P(1) \vee P(2) \vee P(3)$

- **Example:**
  - Let *Q(x)* denote the statement "$x=x+1$".
  - What is the truth value of the quantification $\exists x P(x)$, where the universe of discourse consists of all real numbers?

Unique Existential(唯一存在量词)

- $P(x)$ is true <u>*for one and only one x*</u> in the universe of discourse.
- Notation: *unique existential quantifier*

$$\exists! x P(x)$$

'There is a unique x such that $P(x)$', 'There is one and only one x such that $P(x)$', 'One can find only one x such that $P(x)$'.

Unique Existential(唯一存在量词)

- **Example:**

  U={1, 2, 3}

  $\exists! x P(x) \Leftrightarrow ?$

P(1)	*P(2)*	*P(3)*	*∃!xP(x)*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

$$(\sim P(1) \wedge \sim P(2) \wedge P(3)) \vee (\sim P(1) \wedge P(2) \wedge \sim P(3)) \vee (P(1) \wedge \sim P(2) \wedge \sim P(3))$$

minterms in the PDNF

Uniqueness Quantifier

- **Examples:**
  - If $P(x)$ denotes "$x + 1 = 0$" and U is the integers, then $\exists! x\, P(x)$ is true.
  - But if $P(x)$ denotes "$x > 0$," then $\exists! x\, P(x)$ is false.
- **Note:** The uniqueness quantifier is not really needed as the restriction that there is a unique x such that $P(x)$ can be expressed as:

$$\exists x\, (P(x) \land \forall y\, (P(y) \rightarrow y = x))$$

Quantifiers:

$\forall x\, P(x)$:≡ "For all x, P(x)."
$\exists x\, P(x)$:≡ "There is an x such that P(x)."
$\exists! x P(x)$:≡ "There is one and only one x such that P(x)."

Thinking about Quantifiers

- When the domain of discourse is finite, we can think of quantification as looping through the elements of the domain.
- To evaluate $\forall x\, P(x)$ loop through all x in the domain.
  - If at every step P(x) is true, then $\forall x\, P(x)$ is true.
  - If at a step P(x) is false, then $\forall x\, P(x)$ is false and the loop terminates.
- To evaluate $\exists x\, P(x)$ loop through all x in the domain.
  - If at some step, P(x) is true, then $\exists x\, P(x)$ is true and the loop terminates.
  - If the loop ends without finding an x for which P(x) is true, then $\exists x\, P(x)$ is false.
- Even if the domains are infinite, we can still think of the quantifiers this fashion, but the loops will not terminate in some cases.

Thinking about Quantifiers

- The truth value of $\exists x\, P(x)$ and $\forall x\, P(x)$ depend on both the **propositional function *P(x)*** and on the **domain *U***.
- **Examples**:

1. If U is the positive integers and $P(x)$ is the statement "$x < 2$", then $\exists x\, P(x)$ is true, but $\forall x\, P(x)$ is false.

2. If U is the negative integers and $P(x)$ is the statement "$x < 2$", then both $\exists x\, P(x)$ and $\forall x\, P(x)$ are true.

3. If U consists of 3, 4, and 5, and $P(x)$ is the statement "$x > 2$", then both $\exists x\, P(x)$ and $\forall x\, P(x)$ are true. But if $P(x)$ is the statement "$x < 2$", then both $\exists x\, P(x)$ and $\forall x\, P(x)$ are false.

Precedence of Quantifiers

- The quantifiers $\forall$ and $\exists$ have higher precedence than all the logical operators.
- **Example**
  - $\forall x\, P(x) \lor Q(x)$ means $(\forall x\, P(x)) \lor Q(x)$
  - $\forall x\, (P(x) \lor Q(x))$ means something different.
  - Unfortunately, often people write $\forall x\, P(x) \lor Q(x)$ when they mean $\forall x\, (P(x) \lor Q(x))$.

- **Remember:** A predicate is not a proposition until *all* variables have been bound either by quantification or assignment of a value!

Free and Bound Variables

- An expression like $P(x)$ is said to have a ***free variable x*** (meaning, x is undefined).

- A quantifier (either $\forall$ or $\exists$) *operates* on an expression having one or more free variables, and ***binds*** one or more of those variables, to produce an expression having one or more ***bound variables***.

Example of Binding

- $P(x,y)$ has 2 free variables, x and y.
- $\forall x\, P(x,y)$ has 1 free variable and 1 bound variable. [Which is which?]
- An expression with zero free variables is a bona-fide (actual) proposition.
- An expression with one or more free variables is similar to a predicate:

e.g. let $Q(y) = \forall x\, P(x,y)$

Nesting of Quantifiers

- **Example:**
  - Let the u.d. of x and y be people.
  - Let $L(x,y)$=*"x likes y"*
  - Then $\exists y\, L(x,y)$ = *"There is someone whom x likes."*
  - Then $\forall x\, (\exists y\, L(x,y))$= *"Everyone has someone whom they like."*

(a real proposition; no free variables left)

Binding and Nesting

- **Examples:**
  - $\forall x\, \exists x\, P(x)$ - x is not a free variable in $\exists x\, P(x)$

    Therefore the $\forall x$ binding isn't used.

  - $(\forall x\, P(x)) \wedge Q(x)$ - The variable x is outside of the *scope* of the $\forall x$ quantifier, and is therefore free.

    Not a complete proposition!

  - $(\forall x\, P(x)) \wedge (\exists x\, Q(x))$ – A complete proposition, and no superfluous quantifiers

Binding and Nesting

- If $R(x,y)$= "*x relies upon y,*" express the following in unambiguous English (*U is all people*):
  - $\forall x(\exists y\ R(x,y))$= Everyone has *someone* to rely on.
  - $\exists y(\forall x\ R(x,y))$= There's an overburdened soul whom *everyone* relies upon (including himself)!
  - $\exists x(\forall y\ R(x,y))$= There's some needy person who relies upon *everybody* (including himself).
  - $\forall y(\exists x\ R(x,y))$= *Everyone* has someone who relies upon them.
  - $\forall x(\forall y\ R(x,y))$= *Everyone* relies upon everybody, (including themselves)!

Natural language is ambiguous!

- Let $L(x,y)$ = "*x likes y*"
- "Everybody likes somebody."
  - For everybody, there is somebody they like,
    - $\forall x\ \exists y\ L(x,y)$
  - or, there is somebody (a popular person) whom everyone likes?
    - $\exists y\ \forall x\ L(x,y)$
- "Somebody likes everybody."
  - Same problem: Depends on context, emphasis.

$\exists x\ \forall y\ L(x,y)$

$\forall y\ \exists x\ L(x,y)$

Negations

- Every student in the class has taken a course in Calculus.

$$\sim\forall x P(x) \Leftrightarrow \exists x \sim P(x)$$

  - **Negation:** There is a student in this class who has not taken a course in calculus.
- There is a student in this class who has taken a course in calculus.

$$\sim\exists x P(x) \Leftrightarrow \forall x \sim P(x)$$

  - **Negation:** Every student in this class has not taken a course in calculus.
- **Remember:**
  - Distributing a negation operator across a quantifier changes a universal to an existential and vice versa.

Negations

- **Example 20：** What are the negation of the statements "*there is an honest politician*" and "*All Americans eat cheeseburgers*"?
  - ***H (x) : "x is honest."***

    $\exists xH(x)$, where the domain consists of all politicians.

    The negation is $\neg\exists xH(x)$, which is equivalent to $\forall x\neg H(x)$: *Every politician is dishonest."*

  - ***C(x)： "x eats cheeseburgers."***

    $\forall xC(x)$, where the domain consists of all Americans.

    The negation is $\neg\forall xC(x)$, which is equivalent to $\exists x\neg C(x)$: *"Some American does not eat cheeseburgers" or "There is an American who does not eat cheeseburgers."*

NEGATIONS

- **Example 21：** What are the negation of the statements $\forall x(x^2>x)$ and $\exists x(x^2=2)$?

  - $\sim\forall x(x^2>x) \Leftrightarrow \exists x\sim(x^2>x) \Leftrightarrow \exists x\,(x^2\leq x)$
  - $\sim\exists x(x^2=2) \Leftrightarrow \forall x\sim(x^2=2) \Leftrightarrow \forall x\,(x^2\neq 2)$

Translating from English to Logic

- **Example 1:**

  Translate the following sentence into predicate logic: "*Every student in this class has taken a course in Java.*"

- **Solution:**

  First decide on the domain U.

  Solution 1: If U is all students in this class,

  J(x) denoting "x has taken a course in Java"

  $\forall x\ J(x)$.

  Solution 2: But if U is all people,

  S(x) denoting "x is a student in this class"

  $\forall x\ (S(x) \rightarrow J(x))$

  $\forall x\ (S(x) \wedge J(x))$ is not correct. What does it mean?

Translating from English to Logic

- **Example 2:**

  Translate the following sentence into predicate logic: "*Some student in this class has taken a course in Java.*"

- **Solution:**

  First decide on the domain U.

  Solution 1: If U is all students in this class,

  $$\exists x\, J(x)$$

  Solution 2: But if U is all people,

  $$\exists x\, (S(x) \land J(x))$$

  $\exists x\, (S(x) \rightarrow J(x))$ is not correct. What does it mean?

Excersice-Example 24

- Some student in this class has visited Guangzhou.
- Every student in this class has visited Chengdu or Guangzhou.

  - ***G(x): "x has visited Guangzhou."，C(x):"x has visited Chengdu."***
  - ***The U.D. for the variable x consists of the students in this class,***
    - $\exists x G(x)$.
    - $\forall x (C(x) \lor G(x))$
  - ***The U.D. for the variable x consists of all people.***
    - ***S(x): "x is a student in this class."***
    - $\exists x (S(x) \land G(x))$
    - $\forall x (S(x) \rightarrow (C(x) \lor G(x)))$

Some common shorthands

- Sometimes the universe of discourse is restricted within the quantification.
- *E.g.*,
  - **$\forall x>0\ P(x)$ is shorthand for**
    "For all x that are greater than zero, $P(x)$."
    $\forall x\ (x>0 \rightarrow P(x))$
  - **$\exists x>0\ P(x)$ is shorthand for**
    "There is an x greater than zero such that $P(x)$."
    $\exists x\ (x>0 \land P(x))$

Some common shorthands

- Consecutive quantifiers of the same type can be combined:

  $\forall xyz\ P(x,y,z) \Leftrightarrow_{\text{def}} \forall x\ \forall y\ \forall z\ P(x,y,z)$

  $\exists xyz\ P(x,y,z) \Leftrightarrow_{\text{def}} \exists x\ \exists y\ \exists z\ P(x,y,z)$

- One way of precisely defining the calculus concept of a limit, using quantifiers:

$$\left(\lim_{x\to a} f(x) = L\right) \Leftrightarrow$$
$$\left(\forall \varepsilon > 0 \exists \delta > 0 \forall x \left(0 < |x-a| < \delta\right) \to \left(|f(x) - L| < \varepsilon\right)\right)$$

Lewis Carroll Example

<u>***Charles Lutwidge Dodgson***</u>
<u>***(AKA Lewis Caroll)***</u>
<u>***(1832-1898)***</u>

- The first two are called *premises* and the third is called the *conclusion*.
  1. *"All lions are fierce."*
  2. *"Some lions do not drink coffee."*
  3. *"Some fierce creatures do not drink coffee."*
- Let:
  - *P(x): "x is a lion" ;*
  - *Q(x): "x is fierce" ;*
  - *R(x) : "x drinks coffee".*

1. $\forall x\,(P(x) \rightarrow Q(x))$

2. $\exists x\,(P(x) \land \neg R(x))$

3. $\exists x\,(Q(x) \land \neg R(x))$

- Later we will see how to prove that the conclusion follows from the premises.

Excersice--Example 27

- All hummingbirds are richly colored. $\forall x(P(x) \to S(x))$
- No large birds live on honey. $\neg \exists x(Q(x) \wedge R(x))$
- Birds that do not live on honey are dull in color. $\forall x(\neg R(x) \to \neg S(x))$
- Hummingbirds are small. $\forall x(P(x) \to \neg Q(x))$

- **Let:**
  - *P (x) :“x is a hummingbird,”* ,
  - *S(x) :“x is richly colored,”.*
  - *Q(x): “x is large,”* ,
  - *R(x): “x lives on honey,”*
  - Assuming that the domain consists of all birds.

Bonus Topic: Logic Programming

- There are some programming languages that are based entirely on predicate logic!
- The most famous one is called Prolog.
- Prolog (from *Pro*gramming in *Log*ic) is a programming language developed in the 1970s by researchers in artificial intelligence (AI).
- A Prolog program is a set of propositions (“facts”) and (“rules”) in predicate logic.
- The input to the program is a “query” proposition.
  - Want to know if it is true or false.
- The Prolog interpreter does some automated deduction to determine whether the query follows from the facts.

Logic Programming (cont)

- example of a set of Prolog facts consider the following:
  - *instructor(p,c)*： professor *p* is the instructor of course *c*;
  - *enrolled(s,c)*： student *s* is enrolled in course *c*.

instructor(chan, math273).

instructor(patel, ee222).

instructor(grossman, cs301).

enrolled(kevin, math273).

enrolled(juna, ee222).

enrolled(juana, cs301).

enrolled(kiko, math273).

enrolled(kiko, cs301).

Logic Programming (cont)

- In Prolog, names beginning with an uppercase letter are variables.
- If we have a predicate *teaches(p,s)* representing "professor *p* teaches student *s*," we can write the rule:

  teaches(P,S) :- *instructor(P,C)*, *enrolled(S,C)*.

- This Prolog rule can be viewed as equivalent to the following statement in logic (using our conventions for logical statements).

$$\forall p\ \forall c\ \forall s((I(p,c) \wedge E(s,c)) \rightarrow T(p,s))$$

Logic Programming (cont)

- Prolog programs are loaded into a *Prolog interpreter*. The interpreter receives *queries* and returns answers using the Prolog program.
- For example, using our program, the following query may be given:

  ?enrolled(kevin,math273).

- Prolog produces the response:

  yes

- Note that the ? is the prompt given by the Prolog interpreter indicating that it is ready to receive a query.

Logic Programming (cont)

- The query:

  ?enrolled(X,math273).

  produces the response:

  X = kevin;
  X = kiko;
  no

- The query:

  ?teaches(X,juana).

  produces the response:

  X = patel;
  X = grossman;
  no

• ***The Prolog interpreter tries to find an instantiation for X.***

• ***It does so and returns X=kevin. Then the user types the ; indicating a request for another answer.***

• ***When Prolog is unable to find another answer it returns no.***

Logic Programming (cont)

- The query:

  ?teaches(chan,X).

  produces the response:

  X = kevin;

  X = kiko;

  no

- A number of very good Prolog texts are available. *Learn Prolog Now!* is one such text with a free online version at http://www.learnprolognow.org/
- There is much more to Prolog and to the entire field of logic programming.

Homework

- § 1.4
  - 10, 18, 38, 42, 64