离散数学

Discrete Mathematics

for Computer Science

计算机学院计科系

薛思清 xuesiqing@cug.edu.cn

第3讲 谓词逻辑 Predicate Logic(1)

You are not thinking. You are merely being logical.

- Niels Bohr

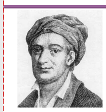
Outline

- 问题，谓词逻辑
- 谓词公式
- 谓词逻辑推理

苏格拉底三段论

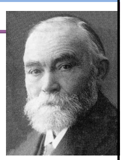
人都是要死的，
苏格拉底是人，
所以苏格拉底是要死的。

“我们从前谁也不相信这个世界还有比我们的伦理更美满、立身处世之道更进步的民族存在，现在从东方的中国，给我们以一大觉醒！东西双方比较起来，我们觉得在工艺技术上，彼此难分高低；关于思想理论方面，我们稍胜于东方一筹，而在哲学实践方面，实在不能不承认我们相形见绌.”



Leibniz

Boole



Frege

Peano

谓词逻辑?

Predicate Logic

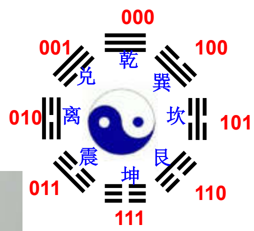
命题逻辑

谓词逻辑

(一阶逻辑First-order Logic)

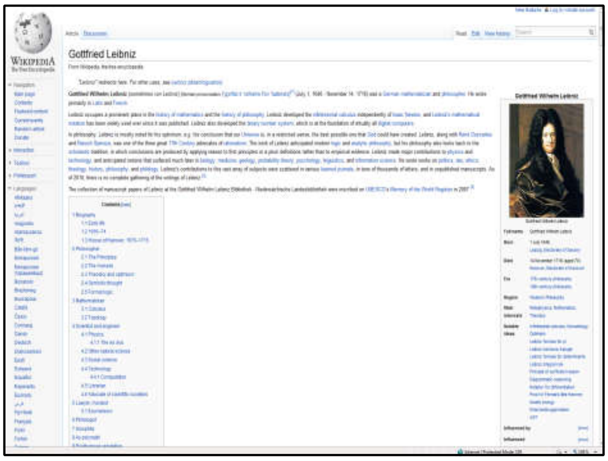
二进制——计算机的语言

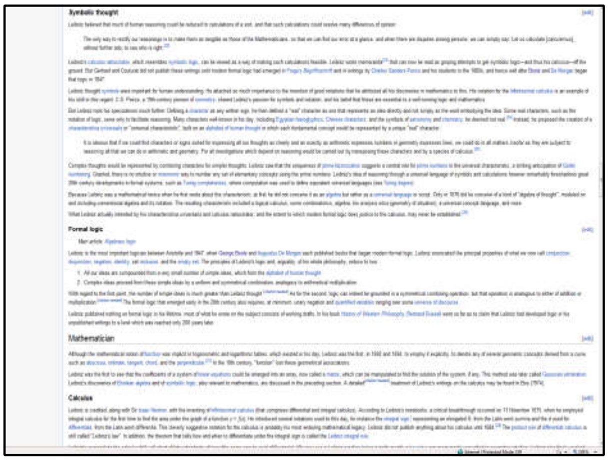
“我的这种不可思议的新发明，…… 是因为我发现了一位圣人的古代文字的秘密，这位古代圣人，就是3000年多前，中国早期的君王伏羲氏”。 -莱布尼兹

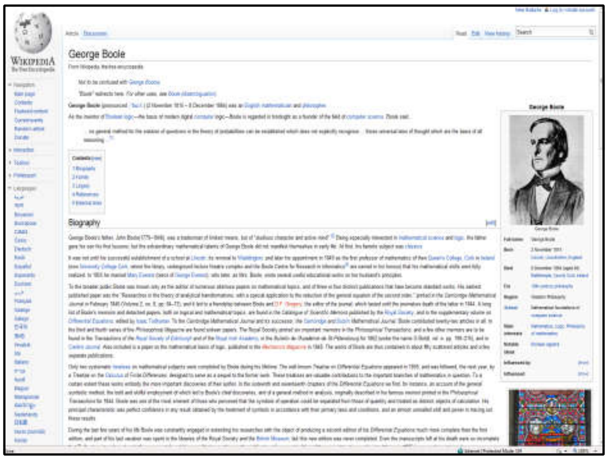


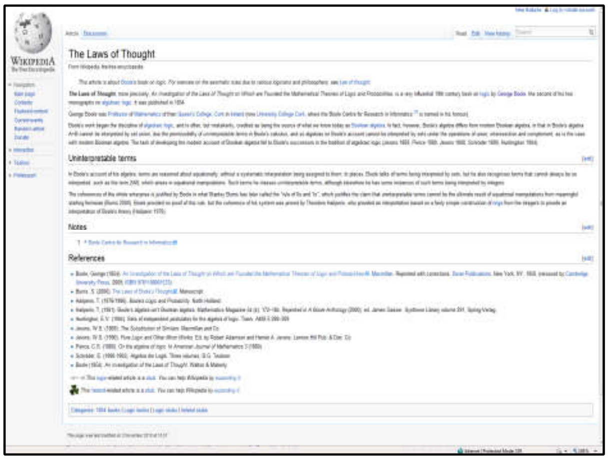
王能超 教授

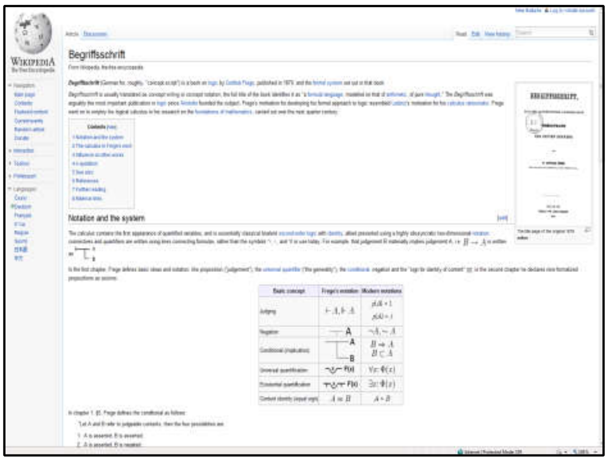
伏羲八卦图（先天八卦图）
用符号表示的二进制数


Gottfried Leibniz


Symbolic thought
Formal logic
Mathematician
Calculus


George Boole
Biography


The Laws of Thought
Uninterpretable terms
Notes
References


Begriffsschrift
Notation and the system

谓词逻辑（Predicate Logic）

1.个体、谓词和量词
2.谓词公式
3.等值演算
4.范式
5.推理理论
6.谓词逻辑在计算机科学中的应用

考虑如下3个命题或推理：

1 “有一个整数大于其他每个整数”

2 “任给ε > 0，存在δ > 0，如果|x-a|<δ，则|f(x)-b|<ε”

3 苏格拉底三段论：“人都是要死的，苏格拉底是人，所以苏格拉底是要死的。”

论域？个体域，全总个体域

量词Quantifier?

有一个整数大于其他每个整数

∃ Existential ~
∀Universal ~
(All)

个体Individual?

谓词Predicate?

个体常元(a,b,...)、个体变元(x,y,...)

表示性质、关系。
谓词常元、谓词变元(P,Q,...)，
n元谓词，(当n=0，即无变元，构成命题)
谓词填式
命题函数(如G(x,y),x,y个体变元)

量词(Quantifiers)：∀ x，□ x

D={d0, d1, d2, ...}, P(x)

(∀ x) P(x) ≡ (P(d0) ∧ P(d1) ∧ P(d2) ∧...)

(□ x) P(x) ≡ (P(d0) ∨ P(d1) ∨ P(d2) ∨ ...)

¬ (∀ x) P(x) ≡ (□ x) ¬P(x)

¬ (□ x) P(x) ≡ (∀ x) ¬P(x)

有一个整数大于其他每个整数

论域为整数集合：∃x(∀yG(x,y))

论域为全总域：∃x(Z(x)∧∀y(Z(y) → G(x,y)))

∃x(Z(x)∧∀y((Z(y) ∧N(x,y)) → G(x,y)))

2 “任给ε>0，存在δ>0，如果|x-a|<δ，则|f(x)-b|<ε” (实数域)

$(\forall \varepsilon)((\varepsilon > 0) \rightarrow (\exists \delta)((\delta > 0) \wedge ((|x-a| < \delta) \rightarrow (|f(x)-b| < \varepsilon))))$

$\forall \varepsilon(G(\varepsilon, 0) \rightarrow (\exists \delta)((G(\delta, 0) \wedge ((G(\delta, |x-a|) \rightarrow G(\varepsilon, |f(x)-b|))))$

3 苏格拉底三段论：“人都是要死的，苏格拉底是人，所以苏格拉底是要死的。” (全总个体域)

令F(x)：x是人，G(x)：x是要死的，a:苏格拉底，则可以形式化为：

前提：∀x(F(x)→G(x)), F(a)

结论：G(a)

1.个体、谓词和量词

练习

1 在我们班同学中，并非所有同学都来自湖北。

2 4班有的同学准备周末去郊游。

3 凡是有理数都可写成分数。

4 对于任意的x, y，都存在唯一的z，使得x+y=z。

1.个体、谓词和量词

练习

1 在我们班同学中，并非所有同学都来自湖北（全总域）。

令S(x)：x是同学， C(x)：x在我们班中， E(x)：x来自湖北，则命题可符号化为：$\neg\forall x((S(x)\wedge C(x))\rightarrow E(x))$。$\exists x(S(x)\wedge C(x)\wedge\neg E(x))$。

2 4班有的同学准备周末去郊游。（全总域）

$\exists x(R(x)\wedge S(x)\wedge T(x))$

3 凡是有理数都可写成分数。（全总域）

$\forall x(Q(x)\rightarrow F(x))$

4 对于任意的x, y，都存在唯一的z，使得x+y=z。（实数域）

$\forall x(\forall y(\exists z((x+y=z)\wedge\forall u((u=x+y)\rightarrow(u=z)))))$

2.谓词公式

谓词公式(Predicate Formula)的构造

类于命题公式的归纳构造方式: 基于原子公式P(x1, x2, …, xn) 、联接词、量词定义

→原子公式、子公式

→量词的优先级高于任何联结词

变元?

公式的真值?

等值演算?

范式?

谓词逻辑推理?

2.谓词公式

Well-Formed Formulas

符号表(Alphabet of symbols)

Individual variables:	x, y, z
Individual constants:	a, b, c
Function letters:	f, g, h
Predicate letters:	p, q, r, P,Q,R
Connective symbols:	$\neg, \rightarrow, \wedge, \vee, \leftrightarrow$
Quantifier symbols:	$\exists, \forall$
Punctuation symbols:	(,), “,”

2.谓词公式

文字(Term) : x, a, and f (x, g(b))

原子公式(Atomic formula, atom): P(x, a) and Q(y, f (c))

wff:

1. Any atom is a wff.
2. If P and Q are wffs and x is a variable, then the following expressions are also wffs:

(P), ¬P, P∨Q, P∧Q, P→Q, P↔ Q, ∃x P, and ∀x P.

2.谓词公式

运算优先级 (the hierarchy of precedence)

$\neg, \exists x, \forall y$	(highest, do first)
$\wedge$	
$\vee$	
$\rightarrow$	(lowest, do last)

Unparenthesized Form	*Parenthesized Form*
$\forall x\neg\exists y\forall z\, p(x,y,z)$	$\forall x(\neg(\exists y(\forall z\, p(x,y,z))))$
$\exists x\, p(x)\vee q(x)$	$(\exists x\, p(x))\vee q(x)$
$\forall x\, p(x)\rightarrow q(x)$	$(\forall x\, p(x))\rightarrow q(x)$
$\exists x\neg p(x,y)\rightarrow q(x)\wedge r(y)$	$(\exists x(\neg p(x,y)))\rightarrow(q(x)\wedge r(y))$
$\exists x\, p(x)\rightarrow\forall x\, q(x)\vee p(x)\wedge r(x)$	$(\exists x\, p(x))\rightarrow((\forall x\, q(x))\vee(p(x)\wedge r(x)))$

2.1 自由变元与约束变元

设α是一个谓词公式，∀xβ(x)和∃xγ(x)是α的子公式，则称∀xβ(x)与∃xγ(x)是α的约束部分（Bound Part），x称为是约束出现（Bound Occurrence）的。

约束出现的变元称为约束变元（Bound Variable），不是约束出现的变元称为自由变元（Free Variable）。

β(x)称为是∀x在α中的辖域（Scope）或作用域，γ(x)称为是∃x在α中的辖域。

1 量词后的用括号括起来的子公式就是其辖域，如果子公式是原子公式，则括号可以去掉。

2 当多个量词连续出现，它们之间无括号分隔时，后面的量词在前面量词的辖域之中，且量词对变元的约束与量词的次序有关，一般不能随意改动。

示例

指出下列公式中，各量词的辖域以及变元的自由出现和约束出现：

1 ∀x(F(x,y,z)→∃yG(x,y))

2 ∃xF(x,y)∧G(x,y)

3 ∀x∀y(F(x)∧G(y)→H(x,y))

4 ∀x∃y(F(x)∧G(y)→H(x,y))

2.2 变元的改名与代入

为了清晰起见，通常运用改名(换名)规则和代入(替换)规则使得公式A满足下列条件：

- 所有变元在公式A中要么自由出现，要么约束出现，不要既有自由出现，又有约束出现。
- 所有量词后面采用的约束变元互不相同。

关于代入

对变元代入，对谓词/命题变元代入→得到更为复杂的公式，从而表示更复杂的知识。显然需要限制：

公式置换/代入后不会产生变元混淆。

约束变元改名规则和自由变元代入规则

- 改名规则：将量词中的作用变元x以及该量词的辖域中相应全部约束变元x都用相同的原公式中不出现的新个体变元y替换，得到公式与原公式等价。
- 代入规则：将公式所有自由变元x改为不在该公式中出现的新变元y，得到公式与原公式等价。

示例

对公式∀x(P(x,y)∧∃yQ(y)∧M(x,y))∧(∀xR(x)→Q(x))中的约束变元进行改名。使每个变元在公式中只以一种形式出现（即约束出现或自由出现）。

解 在该公式中，将P(x,y)和M(x,y)中的约束变元x改名为z，R(x)中的x改名为s，Q(y)中的y改名为t，改名后为：

∀z(P(z,y)∧∃tQ(t)∧M(z,y))∧(∀sR(s)→Q(x))

练习

1 $\exists x((P(x)\vee R(x))\wedge S(x))\rightarrow\forall x(P(x)\wedge Q(x))$

$\exists x((P(x)\vee R(x))\wedge S(x))\rightarrow\forall y(P(y)\wedge Q(y))$

2 $\forall x(P(x)\leftrightarrow Q(x))\wedge\exists xR(x)\vee S(x)$

$\forall x(P(x)\leftrightarrow Q(x))\wedge\exists yR(y)\vee S(z)$

3 $\forall xP(x)\wedge(\exists x)Q(x)\vee(\forall xP(x)\rightarrow Q(x))$

$\forall xP(x)\wedge(\exists y)Q(y)\vee(\forall zP(z)\rightarrow Q(u))$

2.3 公式真值

给定一个文字叙述的命题，可以符号化为谓词公式．反之，给定一个谓词公式，它表达怎样的意义？这涉及谓词逻辑的语义问题．由于谓词公式仅仅是由一些抽象符号构成，只有对它们解释和赋值后，才能讨论公式的意义，公式可能真或可能假 (重言，可满足，矛盾式：tautology, contingency, and contradiction)。

2.3 公式真值

论域

赋值

个体变元

命题变元

谓词

量化

量词量化谓词

{ 论域，(自由)个体变元，个体常元，命题变元/常元，谓词}

(自由)个体变元，命题变元，函数

解释

指 派

示例

1 Let p(x) be the statement "x is an even," where x takes values from the set of integers.

2 ∀ x p(x).

Let p(x) be "x has a parent," where the variable x takes values from the set of people; Let p(x) be "x has a child," again over the set of people.

示例

In the natural number interpretation, where the domain is {0, 1, 2, ...} and > has its usual meaning, we have the following:

Meaning	Formula
There is an element larger than 0.	$(\exists x)\ x > 0$
Every element is larger than itself.	$(\forall x)\ x > x$
Every element is larger than some element.	$(\forall x)(\exists y)\ x > y$
Every element has a larger element.	$(\forall x)(\exists y)\ y > x$
There is an element larger than every other.	$(\exists x)(\forall y)\ (y \mathrel{!=} x \rightarrow x > y.)$

示例　判定下列公式的类型。其中x,y的个体域为整数集I，Q为命题变元，G(x,y)表示x<y。

(1) $\forall x\exists y G(x-y,x+y)\wedge(Q\vee\neg Q)$

(2) $G(x-y,x+y)$

(3) $\forall x\forall y(G(x,y)\wedge\neg G(x,y))\wedge Q$

(4) $G(x-y,x+y)\rightarrow G(x-y,x+y)$

(5) $\forall x\forall y((G(x-y,x+y)\rightarrow G(x+y,x-y))\leftrightarrow(\neg G(x-y,x+y)\vee G(x+y,x-y)))$

Interpretation（解释）

An interpretation for a wff consists of a nonempty set D, called the domain, together with the following assignments of symbols that occur in the wff:

1. Each predicate letter is assigned a relation over D.
2. Each function letter is assigned a function over D.
3. Each free variable is assigned a value in D. All occurrences of the same variable are assigned the same value in D.
4. Each individual constant is assigned a value in D. All occurrences of the same constant are assigned the same value in D.

by James L. Hein

示例 Interpreting a Constant and a Function

$\forall x\,(P(x,c)\rightarrow P(f(x,x),x))$.

1 Let D be the set of positive rational numbers, let P(x, y) be "x < y," let c = 2, and let f(x, y) be the product x · y.
"For every positive rational number x, if x < 2 then x · x < x"

2 Let c = 1 and keep everything else the same.

Truth Value of a Wff（公式真值）

The truth value of a wff with respect to an interpretation with domain D is obtained by recursively applying the following rules:

1. An atom has the truth value of the proposition obtained from its interpretation.
2. Truth values for $\neg U$, $U\wedge V$, $U\vee V$, $U\rightarrow V$ are obtained by applying truth tables for $\neg$, $\wedge$, $\vee$, $\rightarrow$ to the truth values for U and V.
3. $\forall x\ W$ is true if and only if W(x/d) is true for every $d\in D$.
4. □ x W is true if and only if W(x/d) is true for some $d\in D$.

When a wff is true with respect to an interpretation I, we say that the wff is true for I. Otherwise the wff is false for I.

Truth Value of a Wff（公式真值）

Models（模型）

An interpretation that makes a wff true is called a model. An interpretation that makes a wff false is called a countermodel.

Validity（逻辑有效）

A wff is valid if it's true for all possible interpretations. So a wff is valid if every interpretation is a model. Otherwise,the wff is invalid.

A wff is unsatisfiable if it's false for all possible interpretations. So a wff is unsatisfiable if all of its interpretations are countermodels. Otherwise, it is satisfiable.

Truth Value of a Wff（公式真值）

Every wff satisfies exactly one of the following pairs of properties:

valid and thus also satisfiable,
invalid and satisfiable,
unsatisfiable and thus also invalid.

Respectively, the wff is tautology(重言式), contingency(可满足式), or contradiction(不可满足/矛盾式).

Truth Value of a Wff（公式真值）

Proving Validity(逻辑有效性证明)

Direct approach

Indirect approach

示例

Show that the properties of the following wffs(valid, invalid, or satisfiable,unsatisfiable).

a. □ x ∀ y (P(y) → Q(x, y)).
b. p(c) ∧ ¬ p(c).
c. ∀ x p(x) → □ x p(x).
d. □ y ∀ x P(x, y) → ∀ x □ y P(x, y).

示例

□ x ∀ y (P(y) → Q(x, y))

The wff □ x ∀ y (p(y) → q(x, y)) is satisfiable and invalid.
To see that the wff is satisfiable, notice that the wff is true with respect to the following interpretation:
The domain is the singleton {3}, and we define p(3) = True and q(3, 3) = True.
To see that the wff is invalid, notice that it is false with respect to the following interpretation:
The domain is still the singleton {3}, but now we define p(3) = True and q(3, 3) = False.

示例

p(c) ∧ ¬ p(c).

The wff is unsatisfiable.
Suppose the wff is satisfiable. Then there is an interpretation that assigns c a value in its domain such that p(c) ∧ ¬ p(c) = True. Of course, this is impossible. Therefore, the wff is unsatisfiable.

示例

∀ x p(x) → □ x p(x).

The wff is valid.
If the wff is invalid, then there is some interpretation making the wff false. This says that ∀ x p(x) is true and □ x p(x) is false. This is a contradiction because we can’t have p(x) be true for all x in a domain while at the same time having p(x) be false for some x in the domain.

示例

□ y ∀ x P(x, y) → ∀ x □ y P(x, y).

Let A be the antecedent and B be the consequent of the wff(denoted byW).

Direct approach:

Let M be an interpretation with domain D for W such that M is a model for A. Then there is an element $d \in D$ such that $\forall x P(x, d)$ is true. Therefore, $P(e, d)$ is true for all $e \in D$, which says that $\square y P(e, y)$ istrue for all $e \in D$. This says that M is also a model for B. Therefore, W is valid. QED.

Indirect approach:

Assume that W is invalid. Then it has a countermodel with domain D that makes A true and B false.

Therefore, there is an element $d \in D$ such that $\square y P(d, y)$ is false. Thus $P(d, e)$ is false for all $e \in D$.

Now we are assuming that A is true. Therefore, there is an element $c \in D$ such that $\forall x P(x, c)$ is true. In other words, $P(b, c)$ is true for all $b \in D$. In particular, this says that $P(d, c)$ is true.

But this contradicts the fact that $P(d, e)$ is false for all elements $e \in D$. Therefore, W is valid. QED.

Closures(闭公式)

universally/existentially quantify each free variable.

$p(x) \wedge \neg p(y)$, $p(x) \rightarrow p(y)$

$p(x) \wedge \neg p(y)$ is satisfiable, but $\forall x \forall y\, (p(x) \wedge \neg p(y))$ is unsatisfiable.

$p(x) \rightarrow p(y)$ is invalid, but $\square x \square y\, (p(x) \rightarrow p(y))$ is valid.

Closure Properties

1. A wff is valid if and only if its universal closure is valid.
2. A wff is unsatisfiable if and only if its existential closure is unsatisfiable.

The Validity Problem

Given a wff, is it valid?

Decidability

Any problem that can be stated as a question with a yes or no answer is called a decision problem.

A decision problem is called decidable if there is an algorithm that halts with the answer to the problem. Otherwise, the problem is called undecidable.

A decision problem is called partially decidable if there is an algorithm that halts with the answer yes if the problem has a yes answer, but may not halt if the problem has a no answer.

The Validity Problem

Given a wff, is it valid?

For propositional calculus can be stated as follows:

Given a wff, is it a tautology?

This problem is decidable by Quine's method/truth table.

For first-order predicate calculus is undecidable/partially decidable.

There are two partial decision procedures:

natural deduction and resolution

A formula is *valid* if it is true in every interpretation; however, as there may be an uncountable number of interpretations, it may not be possible to check this requirement in practice. M is said to be a model for a set of formulae if and only if every formula is true in M.

There is a distinction between proof theoretic and model theoretic approaches in predicate calculus. *Proof theoretic* is essentially syntactic, and there is a list of axioms with rules of inference. The theorems of the calculus are logically derived (i.e. $\vdash A$) and the logical truths are as a result of the syntax or form of the formulae, rather than the *meaning* of the formulae. *Model theoretical*, in contrast is essentially semantic. The truth derives from the meaning of the symbols and connectives, rather than the logical structure of the formulae. This is written as $\vdash_M A$.

A calculus is *sound* if all of the logically valid theorems are true in the interpretation, i.e. proof theoretic ⇒ model theoretic. A calculus is *complete* if all the truths in an interpretation are provable in the calculus, i.e. model theoretic ⇒ proof theoretic. A calculus is *consistent* if there is no formula A such that $\vdash A$ and $\vdash \neg A$.

The predicate calculus is sound, complete and consistent. *Predicate calculus is not decidable*: i.e. there is no algorithm to determine for any well-formed formula A whether A is a theorem of the formal system. The undecidability of the predicate calculus may be demonstrated by showing that if the predicate calculus is decidable then the halting problem (of Turing machines) is solvable.

by Gerard O'Regan

2.谓词公式-小结

1.命题是陈述句，在自然语言中，通常陈述句有主语、谓语和宾语，主语和宾语都可能是个体，而谓语及其相连的宾语通常被看成是谓词。(谓词填式)

2. 在谓词逻辑中还可描述对个体所进行的某种变换，即引入所函词，函词与谓词不同，函词作用在个体上，而产生另一个个体，而谓词作用在个体上之后产生的是一个命题。

3.个体域不同时，谓词逻辑公式的含义不同。为了使公式有一致的含义，可引入一个全总域，表示宇宙间所有个体所组成的域。在某些情况下，全总域也可指所讨论的问题范围内的所有个体。

4.量词与个体域总是联系在一起的，因此使用不同的个体域，同一命题可能在一阶逻辑中有不同的符号化形式，但总可使所有的个体变量的个体域为全总域，并通过引入合适的特性谓词来从全总域中分离出可使用量词限制的合适个体域来。

5.量词是有顺序的，不能将量词的顺序随意改变。若一个谓词公式中所有个体变元都量化了，则该谓词公式就变成了命题（闭公式）。

2.谓词公式-小结

6.关于"一阶" 逻辑（FOL）

1）所谓一阶逻辑的"一阶"的含义就是指其中的函数只能作用在个体上，或者说其中的变量只能代表取值于个体，如果允许函数作用在谓词上，则变成二阶或高阶逻辑。

$$\forall P((0 \in P \wedge \forall i(i \in P \to i+1 \in P)) \to \forall n(n \in P))$$

2）在一阶逻辑中，谓词变量和谓词常量的区别并不重要，正如在命题逻辑中，命题常量和命题变项的区别不重要一样，这是因为在一阶逻辑中不能对谓词做某些变换（操作），或者说在一阶逻辑上不能在谓词集合上定义函数。但在个体域上可定义函数，因此个体常量与个体变量的区别显得比较重要。

3）可以说，一阶逻辑一个很重要的特点是在个体域上引入了变量。个体变量既可作为谓词作用的对象也可作为函数作用的对象。一阶逻辑中的阶的含义在某种意义上是指个体处于0阶，而对个体的判断（即命题）处于一阶，一阶逻辑中的函数作用于0阶的个体而得到个体，而谓词作用于个体得到处于一阶的命题。

2.谓词公式-小结

7 一阶逻辑公式的"解释"

给定一个文字叙述的命题，可以符号化为谓词公式．反之，给定一个谓词公式，它表达怎样的意义？这涉及谓词逻辑的语义问题。只有对谓词公式解释和赋值后，才能讨论公式的意义，公式可能真或可能假。

一阶逻辑公式的解释显然比命题逻辑公式要复杂得多，因为一阶逻辑公式有非逻辑的符号。对于一阶逻辑公式的解释依赖于一阶逻辑公式所基于的非逻辑符号。设有非逻辑符号集L，它由三部分组成L = C ∪ F ∪ P:

(1). 个体常量所组成的集合[C] = {c_1, c_2, ..., c_n, ...};

(2). 函数符号所组成的集合[F] = { f_1, f_2, ..., f_n, ...}，每个函数f有一个元数n，表明它是n元函数;

(3). 谓词符号所组成的集合[P] = {F1, F2, ..., Fn, ...}，每个谓词F有一个元数n，表明它是n元谓词。

由该非逻辑符号集L生成的项可记为Term(L)，生成的公式可记为Form(L)。为了确定Form(L)中公式的真值，先要给出非逻辑符号集L的解释。

非逻辑符号集L的一个解释[L]由四个部分组成:

[1].一个非空集合D，D称为解释[L]的论域;

[2].对于[C]的个体常量c，其解释为c∈ D是D中的某个元素;

[3].对于[F]的n元函数f，其解释是D上的一个n元函数：$f : D^n \to D$;

[4].对于[P]的n元谓词F，其解释是D上的一个n元关系：$F \subseteq D^n$

(1) 上述定义所给出的解释方法是对非逻辑符号集的一种最直观的解释，称为非逻辑符号集L的塔斯基(Tarski)语义，塔斯基是研究语义学的一个最有名的学者，这种语义解释方法在各种自然语言及形式语言的语义研究中也被广泛使用。

(2) 对L的一个解释也可看成是为L构造了一个模型，研究一个形式语言的模型的有关内容构成了数理逻辑的一个重要分支：模型论(Model Theory)。

(3) 给定一阶语言，我们可以构造它的一个解释，我们也可以给定一个解释所需的东西，然后研究公式的真值，这种研究实际上从某种意义说是对解释的形式化研究。

(4) 只给出解释还不能确定Form(L)中的公式的真值，因为公式中可能存在自由变元，必需为这些自由变元指派具体的个体，不指定具体的个体，则带有自由变元的公式还不能成为命题逻辑的公式。

2.谓词公式-小结

8.关于"一阶逻辑语言"

① 一阶逻辑语言符号包括:

② 个体常量:通常用排在前面的小写字母表示, $a, b, c, ..., a_i, b_i, c_i$ ...

③ 个体变项:通常用排在后面的小写字母表示, $x, y, z, ..., x_i, y_i, z_i$ ...

④ 函数符号:通常用排在中间的小写字母表示, $f, g, h, ..., f_i, g_i, h_i$ ...

⑤ 谓词符号:通常用排在中间的大写字母表示, $F, G, H, ..., F_i, G_i, H_i$ ...

⑥ 量词符号:全称量词∀、存在量词∃

⑦ 联结符号:¬、∧、∨、↔、→

⑧ 辅助符号:(、)、,(逗号)

⑨ 进而可以基于语言符号表定义项(个体单词),谓词公式(原子公式以及更复杂的公式)

2.谓词公式-小结

9.关于谓词公式的书写

一般地,可以用括号表示优先级,但可通过假设联结符号及量词之间的优先级而去掉一些括号,使得公式的书写更为简洁,约定:

(1). 公式的最外层括号可省略;

(2). 联结词¬的优先级高于∧,而∧高于∨,∨高于→,→高于↔,量词的优先级高于任何联结符号, 公式:

$\neg F(x, y) \wedge Q(y, z) \vee \neg F(y, z) \rightarrow G(y, x) \leftrightarrow Q(x, z) \rightarrow F(y, z)$表示:

$(((((\neg F(x, y)) \wedge Q(y, z)) \vee (\neg F(y, z))) \rightarrow G(y, x)) \leftrightarrow (Q(x, z) \rightarrow F(y, z)))$,

但下面的书写既比较简洁,又比较容易理解:

$((\neg F(x, y) \wedge Q(y, z) \vee \neg F(y, z)) \rightarrow G(y, x)) \leftrightarrow (Q(x, z) \rightarrow F(y, z))$