符号化与一阶公式

情形1:

1.一个人的姥姥或者母亲是一个人的先辈

2.一个人的先辈的先辈是这个人的先辈

3.张三是李四的姥姥,李四是王五的母亲

4.所以张三是王五的先辈

情形2:

I.一个数的平方或立方是一个数的幂

II.一个数的幂的幂是一个数的幂

III.64是8的平方, 8是2的立方

IV.所以64是2的幂

两种情形可统一符号化为:

1.$\forall x\forall y((p(y,x)\vee q(y,x))\rightarrow r(y,x))$

2.$\forall x\forall y(\exists z(r(y,z)\wedge r(z,x))\rightarrow r(y,x))$

3.$p(a,b)\wedge q(b,c)$

4.$r(a,c)$

与命题逻辑相比,一阶逻辑引入了:

- 变量,如x和y,可以表示任意多的常量。
- 关系,如$p(y,x)$和$q(y,x)$.

一阶逻辑(First-Order Logic)或**谓词逻辑**(Predicate Logic):

1.逻辑联结词

- $\vee, \wedge, \rightarrow, \leftrightarrow, \cdots$.

2.个体常量

- a: 小明; b: 猫。

3.个体变量:

- x可以是小明(a),也可以是猫(b).

4.个体域(Domain of Discourse):

- 通常用D表示。
- $\exists x \exists y \exists z\,(x^3 + y^3 = z^3)$针对不同的个体域(正整数和正实数)差别明显。

5. 谓词 (Predicates, 本质上是关系)

- $S(x)$: x是大学生。
- $S(?)$: …是大学生。
- $Q(a,b)$: 小明(a)喜欢猫(b).
- $Q(?,?)$: …喜欢….
- $G(a,b,c)$: 上海位于北京和广州之间。
- $G(?,?,?)$: …位于…和…之间。
- 表示n个个体之间关系$P(x_1, x_2, \cdots, x_n)$的谓词称为n元谓词。

6.函数或函词

- $f(x, y, z) = x^2 + y^2 - z^2$.
- 有利于谓词的表达,如$P(f(x, y, z), 0)$.

7.量词(Quantifier)

- $\forall, \exists$.

8.辖域或作用域(Scope)

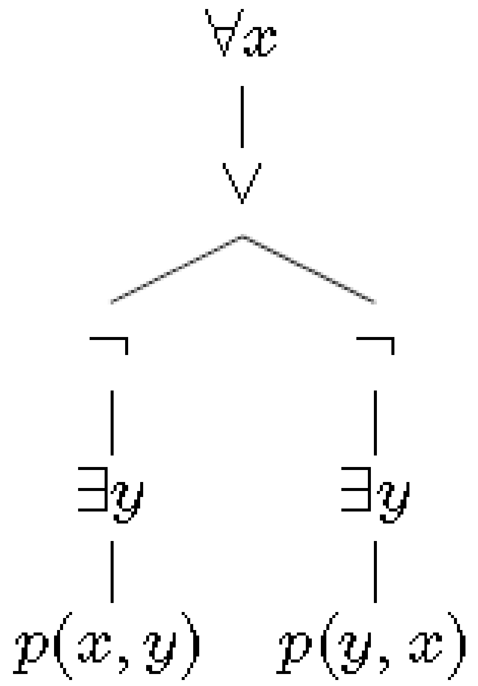
- $\forall x(P(x) \rightarrow \exists y Q(x, y, z)) \wedge R(x)$.
- $\forall x(P(x, y) \wedge Q(y)) \rightarrow \exists y R(x, y)$.

符号化应与自然语言的本意相符。

- “所有A都是B”形式的命题通常符号化为$\forall x(A(x) \rightarrow B(x))$.
- “有一些A是B”形式的命题通常符号化为$\exists x(A(x) \land B(x))$.

谓词公式实例:

$$\forall x(\neg \exists y p(x,y) \lor \neg \exists y p(y,x)).$$



谓词公式树

令P是n元谓词,$x_1, x_2, \cdots, x_n$是个体变量或个体常量,则$P(x_1, x_2, \cdots, x_n)$称为**原子谓词公式**.

特别地,当$n = 0$时,原子谓词公式$P(x_1, x_2, \cdots, x_n)$就是P,此时的原子谓词公式可以看作0元谓词。

复合谓词公式:由量词和逻辑联结词将原子谓词公式联结组成的谓词公式。

若$P(x, y, z)$: $x^3 + y^3 = z^3$, $P(x, y, z)$是原子谓词公式,而$\exists x \exists y \exists z\, P(x, y, z)$是复合谓词公式。

谓词公式形式化定义如下:

(1)原子谓词公式是谓词公式;

(2)如果A是谓词公式,则$\neg A$也是谓词公式;

(3)如果A, B是谓词公式,则$A \wedge B, A \vee B, A \rightarrow B$和$A \leftrightarrow B$也是谓词公式;

(4)如果A是谓词公式,x是个体变元,则$\forall x A$和$\exists x A$也是谓词公式;

(5)有限次使用规则(1)-(4)所得到的由原子谓词公式、逻辑联结词、量词和括号组成的符号串才是谓词公式。

形式化描述:

1.$arg ::= x \quad x \in V$

2.$arg ::= a \quad a \in A$

3.$arg\,list ::= arg$

4.$arg\,list ::= arg,\ arg\,list$

5.$atomic\,wff ::= p(arg\,list),\ \ p \in P,\ \ n \geq 1$

6.$wff ::= atomic\,wff$

7.$wff ::= \neg\,wff$

8.$wff ::= wff\ \circ\ wff,\ \ \circ\!: \vee, \wedge, \uparrow \cdots$

9.$wff ::= \forall x\,wff,\ \ x \in V$

10.$wff ::= \exists x\,wff,\ \ x \in V$

变量分**约束**(Bound)变量和**自由**(Free)变量两种。

在$\forall y\, P(x,y)$中,x是自由变量,y是约束变量。

在$\forall x \forall y (P(x) \to Q(x, f(y), z))$中, x和y是约束变量, z是自由变量。

在$R(x) \to \forall x\, T(x)$中,$x$既是自由变量又是约束变量。也可以说,第一个$x$是自由变量,第二个$x$是约束变量。

没有自由变量的一阶逻辑公式称为**一阶逻辑语句**(First-Order Sentence)。