目录

第一章　一阶逻辑的语言 **3**
1.1　引入 3
1.2　一阶逻辑的语言的定义 3
1.2.1　一阶语言的例子 5
1.3　量词的辖域 (the scope of quantifiers) 5
1.3.1　定义 6
1.3.2　Binding a variable over multiple quantifiers 6
1.4　自由变元和约束变元 8

第一章　一阶逻辑的语言

1.1　引入

与其说是引入, 不如说是一个介绍, 这里我们介绍的便是一阶逻辑的语言. 什么是语言, 就是说, 在我们讲述一阶逻辑的定义和性质的之前, 我们先是要介绍 1. 一阶逻辑的定义; 2. 一阶逻辑的记号.

1.2　一阶逻辑的语言的定义

Definition 1.2.1. 语言 L, 包括了:

1. 括号: nothing special.
2. 联结词: $\{\neg, \to\}$, 也是常规, 就如同我们之前所用到的.
3. 量词: $\forall$, 一阶逻辑之中一个比较重要东西, 说的那么模糊是因为我也不懂.
4. 变元: $v_1, \cdots$, 不知道, 相当于变量 x ?
5. 常数: 和变元相对应的概念.
6. 函数: 可以理解为比函数更为抽象的一个概念.
7. 谓词: 可以理解 be , '是', 当然目前我是这么理解的.
8. 等词: $\approx$, 也是一个谓词, 但是地位非常特殊.

★ **Example 1.**

1. 集合论的**语言**是集合 $\mathcal{L}_{Set} = \{\in, \approx\}$. 其中, $\in$ 是一个二元谓词. 比如说 $a \in A$, $\in$ 有两个 entry, 所以说是二元的.
2. 初等数论的**语言**是集合 $L = \{<, 0, S, +, \cdot, \approx\}$. S 的意思是后继, 是一个一元函数, $+, \cdot$ 均是二元函数. 0 是一个常元.

3. 序关系的**语言**是 $L = \{R\}$.

啊, 我们从这些例子之中, 是否就能够稍微地感受出, 什么是谓词以及什么是函数了.

Definition 1.2.2 (项)**.** 设 L 是一个一阶语言. L 之中的项 (term) 是如下递归地定义的.

1. 变元都是项;
2. 常数都是项;
3. 如果说 $t_1, t_2, \cdots, t_n$ 是项, f 是一个 n 元函数, 那么 $f(t_1, t_2, \cdots, t_n)$ 也是项.

Remark 1. $f(t_1, \cdots, t_n)$ 也可以写为 $ft_1 \cdots t_n$, 我看不出有什么区别. 后者难看一点.

★ **Example 2.** $S0, +v_1SSS0$ 和 $\times S0 + 0SSS0$ 是项. 当然, 我们可以将 $+t_1t_2$ 写为 $t_1 + t_2$, 不会有什么歧义.

$S0$ 就是 0 的后继, viz. 1.
$+v_1SSS0$ viz. 变量 v_1 加上 3.
$\times S0 + 0SSS0$ viz. $(0+3) \times 1$

接下来我们开始定义命题, 或者说公式, 或者说合式公式, 或者说 Proposition.

Definition 1.2.3. L 是一个一阶语言. 其中的 Proposition 定义为:

1. 如果说 $t_1, t_2, \cdots, t_n$ 是项, P 是一个 n 元谓词, 那么 $P(t_1, t_2, \cdots, t_n)$ 是一个 Proposition.
2. 如果说 α, β 是命题, 那么 $(\neg\alpha), (\alpha \to \beta)$ 也是公式.
3. 如果说 α 是公式, 那么 $\forall v_i \alpha$.

Remark 2. 其中 $P(t_1, \cdots, t_n)$ 称为原子公式. 并且, $\forall$ 目前只是一个符号, 现在不能对其做一个含义上的推断. $\forall v_i \alpha$ 并不能理解为 '对于全部的 v...' 什么的.

Remark 3.

1. 我们使用 $\exists x \alpha$ 作为 $\neg(\forall x (\neg\alpha))$ 的简写. 称 $\exists$ 为存在量词.
2. 我们使用 (不约等于) 来表示 $\neg(\alpha \approx \beta)$
3. 我们通常使用 P, Q, R 来表示谓词; 使用 x, y, z 或者 v 来表示变量; 使用 f, g, h 来表示函数; 使用 a, b, c 来表示常数; 使用 $\alpha, \beta, \varphi, \sigma, \tau$ 来表示公式; t 来表示项; Γ, Δ, Σ 来表示公式集合.

1.2.1 一阶语言的例子

例子有很多, 比如我们在学习那个集合论与图论的时候涉及到了一些一阶语言的表述. 所以说是非常脑瘫, 这你吗, 应该先学这个啊! 另外一个例子便是大名鼎鼎的卓里奇的数学分析, 其中运用了大量的一阶逻辑语言来简化表述. 好像没有我想得那么多, 但还是有.

哲学语言 由于分析哲学的原因, 虽然我也不知道是什么原因, 但总之确实是这个原因, 很多哲学家喜欢使用一阶语言来重述一些命题. 总之我们来看

★ **Example 3.** 当今的法国国王是一个秃子.

$$\forall x \left(P_1^1(x) \to P_2^1(x)\right)$$

其中, P 的下标代表的这是第几个谓词, 而上标代表这是几元谓词.

★ **Example 4.** 金山不存在.

$$\neg\exists x \left(P_3^1(x) \wedge P_4^1(x)\right)$$

算术语言 $L = \{\approx, <, 0, S, +, \cdot\}$

★ **Example 5.** 0 不是任何自然数的后继

$$\forall x \left(\neg\left(Sx \approx 0\right)\right)$$

或者说

$$\neg x \left(Sx \approx 0\right)$$

差不多得了, 稍微体验一下就行了.

群的语言 $L = \{e, +\}$

★ **Example 6.** 群加法满足结合律

$$\forall a$$

1.3 量词的辖域 (the scope of quantifiers)

这是个棘手的问题, 比如说下面这个:

★ **Example 7.** Senario: $(\forall y H(y) \rightarrow \exists z W(z, y)) \rightarrow \exists z G(z)$

Question: $\exists z W(z, y)$ 中的 y 是否是包含于 $\forall y$ 的辖域之中呢?

辖域就是这个量词 '管辖' 的范围, 在辖域之中的变量被这个量词限制的.

1.3.1 定义

我们还需要一个正经的定义, 我们给出两种定义:

Definition 1.3.1. L 是一个一阶语言, A 是 L 中的公式, 设 Q 是量词的在 A 之中的一个 occurence, 设其后面接着的变量是 v.

B 是 Q 的辖域, 如果 B 以 Qv 开头, 且其任意的真截断不是公式.

其中 B 的截断 w 满足 $B = ww'$ (B 毕竟是一个字符串), 当 $w' \neq \epsilon$ 的时候, w 称为 B 的真截断, 即, 真截断 w 满足 $w \neq B$

Comment. *ϵ 是空字符串. 这里还是建议不要遗忘这点.*

其实还有等价的定义, B 是以 Qv 开头的, 最大的一个, 公式的子字符串. 接下来给出第二种定义:

Definition 1.3.2. A is a well-formed formula in L, and Q is an occurrence of a quantifier in A.

Let B be a well-formed part of A such that B begins with Qx.

B is called the scope of the quantifier Q.

这里涉及到了非常多的概念, well-formed formula, occurrence, well-formed part. 这些定义还涉及到了 collations 什么的东西. 总之, 啊, 前者比较好理解吧, 毕竟是中文写的.

但是我们这里进行另一个表述, 这个表述要简单得多, 我们只需要记住, 量词 Q 的辖域是紧接其后的一个原子公式.

Definition 1.3.3. A is a formula in L, Q is an occurrence of a quantifier in A, let B be an atomic formula right after Qx, then B is the scope of Q.

1.3.2 Binding a variable over multiple quantifiers

如题, 在同一个变量上面作用了多个量词, 这个时候该怎么进行分析?

★ **Example 8.** 我们来看这个:

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

我甚至读不懂这个东西! 我们根据网络上搜索到的资料来解答.

问题 1

这里是问题来源[1]. 面对上面 Example 4 的时候, 即面对

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

的时候, 我们宣称, 最里面的那个量词才是真正限制变量的那个量词. 这里的话便是 $\forall$ 限制了 x. 就是说,

$$\exists x \forall x P(x) \equiv \exists x \forall y P(y) \equiv \forall y P(y) \equiv \forall x P(x)$$

最外圈那一层的量词, viz. $\exists$ 跟没有一样. 这个问题就和我们程序之中变量名称和类似, 甚至 scope 这个用语都是通用的.

一个回答进一步宣称, $\exists x \forall x P(x)$ 甚至不是一个 well formed formula, 这样的话直接避免处理这种东西, 直接将其视为非法的. 虽然说 $\exists x \forall x P(x)$ 其实是 WFF, 他这个说法其实并不是很能站得住脚, 但是也未必没有参考价值.

问题 2

这里是问题来源[2]. I can't read this: $\forall x \exists x P(x)$. Help me.

这个公式是 well formed 的, 只不过是有点反人类. 当然最好的方式就是避免对同一个变量使用嵌套的量词, 我们应该做的, 就是稍微更改一下使用的变量名称, 使得其更为清晰. 重点便是搞清楚一个量词的辖域, 量词是不能干涉到辖域之中的, 同一个变量名称的辖域的.

以 $\forall x \exists x P(x)$ 为例, $\forall x$ 的辖域是 $\forall x (\exists x P(x))$, 这个辖域之中还有另一个辖域, 并且还是同一个变量, viz. $\exists x P(x)$, 那么外边 $\forall$ 就不能限制 $\exists x P(x)$ 之中的 x.

$$\forall x (\exists x P(x)) \equiv \forall x (\exists y P(y)) \equiv \exists y P(y)$$

进一步, 我们给出一个更为清晰的例子.

$$\forall x \ \Big(Q(x) \wedge \exists x (P(x) \vee R(x))\Big)$$

$\exists x$ 的辖域是 $\exists x (P(x) \vee R(x))$, 这个公式和下面这个是等价的,

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

$\forall x$ 管到了 $Q(x)$ 却没有管辖 $\exists x$ 内部的东西.

[1] https://math.stackexchange.com/questions/3535109/binding-a-variable-over-multiple-quantifiers

[2] https://math.stackexchange.com/questions/1100506/more-than-one-quantifiers-for-one-variable-forall-x-exists-x-px

1.4 自由变元和约束变元