一阶逻辑语言

Zero

虽然我觉得高中学的逻辑知识再加上我的直觉已经够用，但有时闯入思维的死胡同就很难走出来，强迫自己不去想也很不爽，也许还是学习一下都搞清了比较好. 参考的是郝兆宽，杨睿之，杨跃的《数理逻辑：证明及其限度》的第三章[1]，鸣谢 C 同学推荐此书!

1 一阶逻辑语言

定义 1.1 一阶逻辑语言 L 包括:

(0) 括号："(" 和 ")"；

(1) 命题联词："$\neg$"（否定符号，是一元命题联词）和 "$\to$"（蕴含符号，是二元命题联词）；

(2) 全称量词符号："$\forall$"；

(3) 变元：$v_1, v_2, v_3, ...$ 其他常用的符号有：$x, y, z, ...$；

(4) 常数符号：若干符号. 常用的符号有：$a, b, c, ...$；

(5) 函数符号：对每一自然数 n，都有若干符号，称为 n-元函数符号. 常用的符号有：$f, g, h, ...$；

(6) 谓词符号：对每一自然数 n，都有若干符号，称为 n-元谓词符号. 常用的符号有：$P, Q, R, ...$；

(7) 等词符号："$\approx$" .

一些说明:

(0)-(3) 是逻辑符号. (0) 其实是可有可无的（唯一可读性[2]），这里加上

[1] 书中说 "数学基础好的读者可以略过这一章（第一章）"（Page-1 预备知识），"第二章作为之后内容的热身也并非必要，稍作调整后可以直接从第三章开始"（Page-viii 课程大纲）.

[2] 见郝兆宽,《数理逻辑：证明及其限度》§2.4. 而保证唯一可读性的约定——也就是在没有括号的情况下如何安排优先级——见5.

是为了方便人类阅读；(1) 中的两个命题联词已经是功能完全的[3]；(2) 和 (3) 是一阶逻辑的关键，这里的变元仅代表个体，量词也只控制个体[4].

(4)-(7) 是非逻辑符号，是从数学中提炼出来的. 其中 (4)、(5) 和 (6) 中，“若干” 的含义为：逻辑语言中可以没有、或有有限多个、或有无穷多个该符号；(7) 等词符号是一个 2-元谓词符号，逻辑语言中可以没有它；把 (7) 等词符号同 (6) 谓词符号区分开来，是因为等词必须解释成等号，而谓词允许有不同的解释.

下面是两个一阶逻辑语言的例子[5]. 一般只列出非逻辑符号，因为它们才是可以变化的部分；逻辑符号则默认已经包括.

例 1.2 公理集合论的语言为：$L_{Set} = \{\approx, \in\}$. 其中：

1. 常数符号：$\{\}$；
2. 函数符号：
   (a) 0-元函数符号：$\{\}$；
   (b) 1-元函数符号：$\{\}$；
   (c) 2-元函数符号：$\{\}$；
   (d) ...
3. 谓词符号：
   (a) 0-元谓词符号：$\{\}$；
   (b) 1-元谓词符号：$\{\}$；
   (c) 2-元谓词符号：$\{\in\}$；
   (d) ...
4. 等词符号：$\{\approx\}$.

例 1.3 初等数论[6]的语言为：$L_N = \{\approx, <, 0, S, +, \cdot\}$. 其中：

1. 常数符号：$\{0\}$；
2. 函数符号：

[3]指的是这两个联词的真值表能表达所有的 4 个一元联词和 16 个二元联词. 见郝兆宽，《数理逻辑：证明及其限度》§2.5. 更多的命题联词见4

[4]这是与高阶逻辑比如二阶逻辑做一下区分，二阶逻辑语言更加丰富，但一阶逻辑有完全性，二阶逻辑则没有.

[5]虽然看上去都是数学的例子，但实际上这里还没有涉及任何与意义有关的东西，把它们都看成没有意义的符号即可. 当然实际上这些符号是有意义的，而意义的赋予会在后面7中进行.

[6]关于初等数论的公理化理论体系的建立，可以参考 (待补充...)

(a) 0-元函数符号：{}；

(b) 1-元函数符号：$\{S\}$；

(c) 2-元函数符号：$\{+,\cdot\}$；

(d) ...

3. 谓词符号：

(a) 0-元谓词符号：{}；

(b) 1-元谓词符号：{}；

(c) 2-元谓词符号：$\{<\}$；

(d) ...

4. 等词符号：$\{\approx\}$.

定义 1.4 用一阶逻辑语言中的符号任意组成的一串任意符号串称为一个表达式（不需考虑它的意义）.

比如，“$\approx) \to \forall$” 就是 L_{Set} 中的一个表达式.

2 项

定义 2.1 令 L 是一个一阶逻辑语言. L 中所有项的集合为，满足下列条件的最小的表达式的集合：

(1) 每个变元 v_i 都是一个项；

(2) 每个常数符号都是一个项；

(3) 如果 $t_1, t_2, ..., t_n$ 是项，且 f 为一个 n-元函数符号，则 $ft_1t_2...t_n$ 也是一个项.

常用来表示项的符号有：$t_1, t_2, t_3, ...$

我的笔记: 这个定义是“自上而下”的，可以类比抽象代数中对有限生成子群或域的扩张的定义.

定义 2.2 设 K 是域 F 的一个扩域，S 是 K 的一个子集. 用 $F(S)$ 表示 K 的含 $F \cup S$ 的最小子域.

定义 2.3 设 K 是域 F 的一个扩域，S 是 K 的一个子集. 定义 K 的子环

$$F[S] = \left\{ \sum f_{n_1,n_2,...,n_m} s_1^{n_1} s_2^{n_2} ... s_m^{n_m} \middle| \forall s_i \in S, n_i \in \mathbb{Z}, m \in \mathbb{N}, f_{n_1,n_2,...,n_m} \in F \right\},$$

再定义 $T = \left\{uv^{-1} \middle| u, v \in F[S], v \neq 0\right\}$.

事实上，在抽象代数中我们证明了 $F(S) = T$. 在这个例子中，2.2是一个“自上而下”的定义，2.3是一个“自下而上”的定义.

例 2.4 $v_1, 0, S0, +v_10, \cdot S0 + 0SSS0$ 都是 L_N 中的项.

3 合式公式

定义 3.1 令 L 是一个一阶逻辑语言. L 中所有合式公式的集合为，满足下列条件的最小的表达式的集合：

(1) 如果 $t_1, t_2, ..., t_n$ 为 L 中的项，并且 P 为一个 n-元谓词符号，则 $Pt_1t_2...t_n$ 是一个合式公式. 我们称这样的公式为原子合式公式. 特别地，$\approx t_1t_2$ 是一个原子合式公式；

(2) 如果 α 和 β 是合式公式，则 $(\neg\alpha)$ 和 $(\alpha \to \beta)$ 是合式公式；

(3) 如果 α 是一个合式公式，则 $\forall v_1\alpha$ 是合式公式.

常用来表示合式公式的符号有：$\alpha, \beta, \varphi, \sigma, ...$

例 3.2 $(< v_10), (\approx (S0)(+v_10)), ((\neg(< v_10)) \to \forall v_1((\approx (\cdot S0 + SSS0)(+v_10))))$ 都是 L_N 中的合式公式.

4 新符号

定义 4.1 为了方便引入一些符号，这是数理逻辑这门学科的符号用法.

1. $(\alpha \vee \beta) := ((\neg\alpha) \to \beta)$. 符号“$\vee$”称为析取符号，可看作是二元命题联词；
2. $(\alpha \wedge \beta) := (\neg(\alpha \to (\neg\beta)))$. 符号“$\wedge$”称为合舍符号，可看作是二元命题联词；
3. $(\alpha \leftrightarrow \beta) := ((\alpha \to \beta) \wedge (\beta \to \alpha))$. 符号“$\leftrightarrow$”称为双蕴含符号，可看作是二元命题联词；
4. $(\exists x\alpha) := (\neg\forall x(\neg\alpha))$，称“$\exists$”为存在量词符号；
5. $(t_1 \approx t_2) := (\approx t_1t_2)$；
6. $(t_1P_2t_2) := P_2t_1t_2$，其中 P_2 是任意 2-元谓词符号；
7. $(t_1 \not\approx t_2) := (\neg \approx t_1t_2)$；

5 一些约定

一阶逻辑语言的合式公式有唯一可读性，在不引起混乱的情况下我们尽量少些括号. 关于括号省略的约定有：

(1) 最外的括号总是省略；

(2) 命题联词的否定符号 $\neg$ 、全称量词符号 $\forall$ 和存在量词符号 $\exists$ 的管辖范围尽可能短. 以全称量词符号为例：$\forall v_i\alpha \to \beta$ 表示 $(\forall v_i\alpha) \to \beta$ 而不是 $\forall v_i(\alpha \to \beta)$.

(3) 二元命题联词反复出现时，以右为先. 例如：$\alpha \to \beta \wedge \gamma$ 指的是 $(\alpha \to (\beta \wedge \gamma))$.

例 5.1 *3.2*中最复杂的一个合式公式

$$((\neg(< v_1 0)) \to \forall v_1 ((\approx (\cdot S0 + SSS0)\,(+v_1 0))))$$

去掉所有括号后为 $\neg < v_1 0 \to \forall v_1 \approx \cdot S0 + SSS0 + v_1 0$. 在上述约定下对它进行解读是没有歧义的.

上面的讨论完全是形式化的处理，下面赋予意义. 首先定义真值指派：

6 真值指派

定义 6.1 令 L 是一个一阶逻辑语言，$\Gamma = \{\alpha, \beta, ...\}$ 是 L 中所有合式公式的集合，给定集合 $\{0,1\}$，可称之为真假值集合. L 的一个真值指派是一个原像为合式公式集合、像为真假值集合的映射 $\imath\colon \Gamma \to \{0,1\}$，满足如下条件：

(1) $\imath(\neg\alpha) = \begin{cases} 1, & \text{如果} \imath(\alpha) = 0; \\ 0, & \text{如果} \imath(\alpha) = 1. \end{cases}, \forall \alpha \in \Gamma;$

(2) $\imath(\alpha \to \beta) = \begin{cases} 0, & \text{如果} \imath(\alpha) = 1 \text{ 且} \imath(\beta) = 0; \\ 1, & \text{其他情况.} \end{cases}, \forall \alpha, \beta \in \Gamma;$

(3) 关于量词 $\forall$ 的规则，待补充.

条件只限制在 2 个命题联词和全称量词符号上. 4中用这 2 个命题联词和全称量词符号又组合出了 3 个新的命题联词和存在量词符号，当然这 3 个新的命题联词和存在量词符号的真值指派规则也是唯一永恒的. 真值指派

要满足的关于命题联词的条件可以用真值表来表达，这是一种高效的，易于查阅的方法. 我们现在给出真值表，见1. 注意这是真值指派的定义的一部分，是任何真值指派必须满足的条件，其中前 2 列是定义，后 3 列是满足前 2 列就一定会同时满足的，给出只是为了方便使用. 其中 $\forall \alpha, \beta \in \Gamma$.

表 1: 真值表

	$\imath(\neg\alpha)$	$\imath(\alpha \to \beta)$	$\imath(\alpha \vee \beta)$	$\imath(\alpha \wedge \beta)$	$\imath(\alpha \leftrightarrow \beta)$
$\imath(\alpha)=1, \imath(\beta)=1$	0	1	1	1	1
$\imath(\alpha)=1, \imath(\beta)=0$	0	0	1	0	0
$\imath(\alpha)=0, \imath(\beta)=1$	1	1	1	0	0
$\imath(\alpha)=0, \imath(\beta)=0$	1	1	0	0	1

一个一阶逻辑语言的真值指派不唯一.

到这里其实仍然是完全形式上的定义，但有了真值指派后，我们就可以赋予这些符号一些意义了. 比如，我们可以赋予自然语言的意义：一个合式公式代表一句话，一个真值指派代表一个判断标准，某个真值指派把一个合式公式映射到 0 代表那句话是假话，映射到 1 则代表那句话是真话...

下面赋予数学语言的意义.

7 赋予数学上的意义

几乎所有数学命题皆可以在某种一阶语言中表达[7]. 先简单复习一下一阶逻辑语言.

a. 一阶逻辑语言的基本符号构件有：一元命题联词 $\neg$，二元命题联词 $\to, \vee, \wedge, \leftrightarrow$；量词符号 $\forall, \exists$；变元 $v_1, v_2, ..., x, y, z, ...$；常数符号 $a, b, c, ...$，函数符号 $f, g, h, ...$；谓词符号 $P, Q, R, ..., \approx$. （见1.1和4）

b. 变元、常数符号和函数符号组成项. （见2）

c. 项加上谓词符号、命题联词和量词符号组成合式公式. （见3）

d. 每一个合式公式映射到 0 或 1 完成真值指派. （见6）

大致按构造顺序进行对应如下：

(1) 定义某个一阶逻辑语言，对应：公理化某个数学体系.

[7]书上是这样说的，但这个“几乎”二字没有细说，待考察反例. 另外这个论断有待加强（也许满足某些性质的所有数学命题皆可以在某种一阶语言中表达），且这个加强版的命题有待确定那个具体的性质和给出证明. 这些问题可能与书中后面有关哥德尔完全性定理的部分有关，未来学习后再补充.

(2) 在定义某个一阶逻辑语言时会给出若干常数符号和若干 1-元函数符号、2-元函数符号等，对应：公理化某个数学体系时给出的常数和一元函数、二元函数等，它们的具体意义是在公理化的过程中规定的，函数的使用方法见 (5).

(3) 项，对应：数学对象.

(4) 变元，对应：不具体确定的泛指的某个数学对象.

(5) 变元、常数符号和函数符号组成项，对应：变元是数学对象，常数是数学对象，函数作用到数学对象上得到的还是数学对象.

(6) 合式公式，对应：命题，数学中常用的词语有公理、定理、推论、猜想、错误的陈述等.

(7) 每一个合式公式映射到 0 或 1 完成真值指派，对应：在该数学体系中确定命题是真还是假. 由于一阶逻辑语言中真值指派不唯一，而数学体系要求命题的真假是唯一的，所以在数学体系公理化的过程中会给出一些公理，直接认为公理是真命题. 反映到一阶逻辑语言中就是规定一些合式公式映射的像是 1. 而这种做法导致了多少个不同的真值指派呢？如果恰好有 1 个就是我们想要的，如果有 0 个说明公理出现了矛盾，如果多于 1 个说明公理还不完善[8]. 数学中用公理、命题、定理、推论等表示真命题，用猜想表示人类还不知道是真是假的命题，用谬误表示假命题，当然也直接使用命题、真命题、假命题这些词语.

(8) 在定义某个一阶逻辑语言时会给出若干 1-一元谓词符号、2-元谓词符号等，对应：公理化某个数学体系时给出的一些数学对象之间的关系，它们的具体意义是在公理化的过程中规定的.

(9) 在定义某个一阶逻辑语言时会给出[9]等词符号 $\approx$，对应：等于号，在数学中用符号 $=$.

(10) 命题联词，对应的意义其实已经通过真值指派的定义（也就是真值表）确定好了. $\neg\alpha$ 对应数学中对 α 这个命题的否定；$\alpha\to\beta$ 对应数学中命题 α 蕴含命题 β，在数学中用符号 $\alpha\implies\beta$ 或 $\beta\impliedby\alpha$，也说命题 β 蕴含于命题 α，或，命题 α 是命题 β 的必要条件，或，命题 β 是命题 α 的充分条件；$\alpha\vee\beta$ 对应数学中命题 α 或命题 β；$\alpha\wedge\beta$ 对应数学中命题 α 且命题 β；$\alpha\leftrightarrow$ 对应数学中命题 α 等价于命题 β，在数学中用符号 $\alpha\iff\beta$，

[8]哥德尔的定理好像是在说不可能证明出某个一阶逻辑语言恰好有 1 个真值指派，有可能有一些合式公式该如何映射到真假值集合是不能确定的. 不过我还没看先不瞎说了.

[9]在描述数学体系的一阶逻辑语言中是一定会给出的!

也说命题 α 当且仅当命题 β，也说命题 α 是命题 β 的充分必要条件.[10]

(11) 全称量词符号 $\forall$，对应：全称量词任意.

(12) 存在量词符号 $\exists$，对应：存在量词存在.

(13) 项加上谓词符号、命题联词和量词符号组成合式公式，对应：有关数学对象之间的关系的论断都是命题，命题的否定、命题的蕴含关系、命题的或、命题的且、命题的等价关系都是命题，把命题中的某些数学对象用存在或任意限制住得到的也是命题.

至此我们就说清了数学是如何被一阶逻辑语言表达的. 数学会用一个又一个定义来简化数学对象或数学对象之间的关系，反映到一阶逻辑语言上就是定义一些新的项和新的谓词符号. 如果对一阶逻辑语言的运行方式不了解，就有可能对某些新的谓词符号感到困惑，而这正是我学习一阶逻辑语言的初衷.

8 数学被还原为一阶逻辑语言的例子

见郝兆宽，《数理逻辑：证明及其限度》§3.1.2. 这一节给了很多数学的例子，包括集合论语言、集合论语言、群的语言和环的语言.

9 其它问题

可参考任意数理逻辑的教材. 包括自由出现和约束出现的问题，形式证明和数学证明的问题，逻辑方法的局限性问题（不完全性定理）等等.

[10]在数学中它们的意义是与真值指派密切相关的，因此把真值指派写在了前面，实际上，真值指派的合式公式对应到数学中的命题是如何构造出来的还没有说，见 (13).