谓词演算基础概念辨析

本课程的学习已经来到了谓词演算阶段。在解答同学们问题的过程中我发现大家对本章节初期的许多基础概念存在困惑或误解，这里给大家分享一些我的理解。

1. 谓词演算的建立

学习谓词演算，最好能够先充分理解谓词演算的建立与规定，这样后续能少走很多歪路。

首先来看谓词演算系统的基本结构：

我们由四个集出发：个体变元集 X，个体常元集 C，运算（函数）集 F，谓词集 R. 用不同的 C, F 和 R 可构造出不同的谓词演算系统.

个体变元集 $X = \{x_1, x_2, \cdots\}$ 是可数集. 个体变元 x_i 可用来表示某个个体对象. 有时为了方便，我们也用 x, y, z 等来表示个体变元.

个体常元集 $C = \{c_1, c_2, \cdots\}$ 是可数集，也可以是有限集（包括空集）. 个体常元 c_i 可用来表示确定的个体对象.

运算集 $F = \{f_1^1, f_2^1, \cdots, f_1^2, f_2^2, \cdots, f_1^3, f_2^3, \cdots\}$ 是可数集，也可以是有限集（包括空集）. f_i^n 叫做第 i 个 n 元运算符或函数词，用来表示某个体对象集上的 n 元运算. 注意符号 f_i^n 的上标 n 是该运算符的元数.

谓词集 $R = \{R_1^1, R_2^1, \cdots, R_1^2, R_2^2, \cdots, R_1^3, R_2^3, \cdots\}$ 是可数集，也可以是有限集，但不能是空集. R_i^n 叫做第 i 个 n 元谓词，用来表示某种个体对象集上的 n 元关系. R_i^n 的上标 n 是该谓词的元数.

基本的结构是四个集合，分别包含变元、常元、运算符、谓词。

接下来在此基础上构建项集：

建立谓词演算的第一步是建立项集 T，项的形成规则是：

(i) 个体变元 $x_i (\in X)$ 与个体常元 $c_i (\in C)$ 都是项.

(ii) 若 $t_1, \cdots, t_n$ 是项，则 $f_i^n(t_1, \cdots, t_n)$ 也是项 $(f_i^n \in F)$.

(iii) 任一项皆如此形成，即皆由规则 (i)，(ii) 的有限次使用形成.

当运算符集 $F = \varnothing$ 时，规定项集 $T = X \cup C$.

按上述规则，当 $F \neq \varnothing$ 时，项集 T 可如下分层：

$$T = T_0 \cup T_1 \cup T_2 \cup \cdots \cup T_k \cup \cdots$$

其中

$$\begin{aligned}
T_0 &= X \cup C = \{x_1, x_2, \cdots, c_1, c_2, \cdots\}, \\
T_1 &= \{f_1^1(x_1), f_1^1(x_2), \cdots, f_1^1(c_1), \cdots \\
&\quad f_2^1(x_1), \cdots, f_2^1(c_1), \cdots \\
&\quad \cdots\cdots \\
&\quad f_1^2(x_1, x_1), \cdots \\
&\quad \cdots\cdots \\
&\quad f_1^3(x_1, x_1, x_1), \cdots \\
&\quad \cdots\cdots\}, \\
T_2 &= \{f_1^1(f_1^1(x_1)), \cdots, f_1^2(x_1, f_1^1(x_1)), \cdots\}, \\
&\quad \cdots\cdots
\end{aligned}$$

第 k 层项 (T_k 的元素) 含有 k 个运算符. 换句话说，第 k 层项由零层项（个体常元、个体变元）经 k 次运算得来. 更简单地说：项集 T 是由 $X \cup C$ 生成的 F 型代数（即以 F 为运算集的代数系统）.

项的构建以个体变元与个体常元为基本项，再通过运算符来生成一层新的项，新的项又

能通过运算符来生成又一层新的项，类似可层层迭代。

由此我们可以初步了解运算符的含义。如“求和、平方、立方”等**运算**都可以视作运算符，类比成编程中的函数，返回结果为一个“运算值”。

注意这些“运算值”在某种意义上与基本变元/常元是同等地位的，也即此处所称的“项”。

由此还能引申出“**闭项**”的定义：

定义 1 (闭项) 只含个体常元的项叫做闭项.

例 1 中写出的 $f_1^1(c)$, $f_1^2(c_1, c_1)$, $f_1^1(f_1^1(c_1))$, $f_1^1(f_1^2(c_1, c_1))$ 等就是闭项，而 $f_1^1(x_1)$, $f_1^2(c_1, x_1)$ 等都不是闭项.

由项集的生成过程可以看出，所有闭项的集是由个体常元集 C 生成的 F 型代数.

不难看出，它是项的一个子集。

之后来构建公式集，首先是原子公式：

定义 2 (原子公式集) 原子公式集是指

$$Y = \bigcup_{i,n} \left(\{R_i^n\} \times \underbrace{T \times \cdots \times T}_{n\text{个}T} \right)$$

即

$$Y = \{(R_i^n, t_1, \cdots, t_n) \mid R_i^n \in R, t_1, \cdots, t_n \in T\}.$$

以后常把原子公式 $(R_i^n, t_1, \cdots, t_n)$ 写成 $R_i^n(t_1, \cdots, t_n)$.

在谓词演算中，原子公式是用来表示命题的最小单位，而命题演算是用单个命题变元来表示简单命题的. 研究谓词演算，必须重视项集这一层次. 项是构成原子公式的基础.

每个原子公式由 n 元谓词与 n 个项构成。

注意这里原子公式的初始写法：$(R_i^n, t_1, \ldots, t_n)$，这实际是一种 **n 元关系**的表示。出于某些考虑以后我们也将它写作类似项结构的表示：$R_i^n(t_1, \ldots, t_n)$，但就如前面所说这实际是一种关系的表示而不是运算，注意不要与项混淆！

之后在此基础上正式构建公式集：

2.1.2 谓词演算公式集

建立谓词演算公式集前，先列出我们所采用的这种形式语言的字母表如下:

个体变元 $x_1, x_2, \cdots$	(可数个)
个体常元 $c_1, c_2, \cdots$	(可数或有限个)
运算符 $f_1^1, f_2^1, \cdots, f_1^2, f_2^2, \cdots$	(可数或有限个)
谓词 $R_1^1, R_2^1, \cdots, R_1^2, R_2^2, \cdots$	(可数或有限个，至少一个)
联结词 $\neg, \rightarrow$	
全称量词 $\forall$	
左右括号、逗号 (,), ,	

形成谓词演算公式集的过程与命题演算类似，不同的是:

第一，现由原子公式集 Y 出发，而不是由命题变元集出发;

第二，除了一元运算 $\neg$ (否定) 与二元运算 $\rightarrow$ (蕴涵) 外，现还增加了可数个一元全称量词运算“$\forall x_i$” $(i = 1, 2, \cdots)$.

谓词演算公式的形成规则是:

(i) 每个原子公式是公式.

(ii) 若 p, q 是公式，则 $\neg p$, $p \rightarrow q$, $\forall x_i\, p\, (i = 1, 2, \cdots)$ 都是公式.

(iii) 任一公式皆如此形成，即皆由规则 (i), (ii) 的有限次使用形成.

除了 $\neg, \rightarrow$ 和 $\forall x_i$, 还可在 $K(Y)$ 上定义新的运算 $\vee, \wedge, \leftrightarrow$ 及 $\exists x_i$ (存在量词运算), 定义式是:

$$p \vee q = \neg p \rightarrow q,$$
$$p \wedge q = \neg(p \rightarrow \neg q),$$
$$p \leftrightarrow q = (p \rightarrow q) \wedge (q \rightarrow p),$$
$$\exists x_i p = \neg \forall x_i \neg p.$$

注意公式 $\forall x(p \rightarrow q)$ 和公式 $\forall x p \rightarrow q$ 的区别. 前者 $\forall x$ 的作用范围 (简称“范围”) 是 $p \rightarrow q$, 而后者 $\forall x$ 的范围是 p.

注意这里全称量词和存在量词的引入及其作用范围的限定，这也是后面理解“自由出现”相关内容的基础。

在后续学习谓词演算的语义大家会对这里的定义有更深的理解，这里先大致解释一二：约定自然数集 N，

个体变元x_1, x_2, x_3...等就像式子中假设的参数，值域为 N；

个体常元c_1, c_2, c_3...等就像式子中出现的常数，均属于 N；

运算符可以视作 N 上的多元运算，得到的结果值域也为 N；

谓词则可以视作 N 上的多元关系：

对$R_i^n(t_1, \ldots, t_n)$而言，仅有$(t_1, \ldots, t_n) \in R_i^n$以及$(t_1, \ldots, t_n) \notin R_i^n$ 两种可能，若将这个原子公式视作命题，那么如果$(t_1, \ldots, t_n) \in R_i^n$则命题成立，反之亦反。

此处的说法仅是一种方便大家建立初步理解的粗略解释，具体的严格定义会在后续学习中了解。

2. 公式中变元与项的自由概念

了解公式集的建立后我们再来看公式中有关“自由”的一些概念。

首先是变元：

定义 1 (变元的自由出现与约束出现) 在一个公式里，个体变元 x 的出现如果不是在 $\forall x$ 中或在 $\forall x$ 的范围中，则叫做自由出现，否则叫做约束出现.

比如，在 $\forall x_1 (R_1^2 (x_1, x_2) \rightarrow \forall x_2 R_1^1 (x_2))$ 中，x_1 约束出现两次，x_2 约束出现两次且自由出现一次.

在这个例子中，蓝色为约束出现，红色为自由出现。

由此还能引申出“闭式”的定义：

定义 2 (闭式) 公式若不含自由出现的变元，则叫做闭式.

例如，公式 $\forall x_1(R_1^2(x_1, x_2) \rightarrow \forall x_2 R_1^1(x_2))$ 不是闭式，因为 x_2 在其中自由出现一次. 公式 $\forall x_1 (R_1^2 (x_1, c_1) \rightarrow \forall x_2 R_1^1 (x_2))$ 是闭式.

可能由于量词的存在而产生“变元干扰”，是谓词演算中的一件麻烦事. 例如，在用项去替换公式中的个体变元时就可能出现这种干扰. 项所含的变元本是自由而不受约束的，但若需要用某个项去替换一个公式中自由出现的变元时，该项中的变元可能会受约束.

注意与前面“闭项”的类比与区别：

闭代表封闭，或者说无变化。

以变元/常元为基础，由运算符我们构建了项，若参与项构成的基础元只有个体常元，则它是确定性的，无变化的，称为闭项；

以项为基础，由谓词我们构建了公式，若公式中所有的变元都不是自由出现的，这个公

式便已经是封闭的、确定的了。

这里的确定指的是公式的具体含义，以以下公式为例：

$$\forall x_1\ R_1^2(x_1, x_2)$$

我们假设 R_1^2代表的是自然数集 N 上的“大于等于”关系，则公式的**基础架构**可以理解为“x_1任意取值均大于等于x_2”，即一个数不管取值如何总是大于等于另一个数。公式中有一个自由出现的变元x_2，随着x_2的变化，公式可以表达一系列具有类似结构的含义。

当x_2的值变化时，整个公式的含义甚至真假并不相同：

若x_2定为常数 0，则公式为“自然数集 N 上x_1任意取值均大于等于 0”，直观可知为真；

若x_2定为常数 2，则公式为“自然数集 N 上x_1任意取值均大于等于 2”，这个式子的含义产生了变化，并且直观可知为假。

对一个没有自由变元出现的公式（即闭式）：

$$\forall x_1\ R_1^2(x_1, x_1)$$

同样采用上述假定，则公式的结构可以理解为“任意x_1均大于等于x_1”。公式表示“自然数集 N 上任意数均大于等于它本身”，即一个数不管取值如何总是大于等于它自己。它的架构是确定的，含义也是确定的，没有自由变元出现故无法更改，并且这是个永真式。

最后是项：

定义 3 (项 t 对公式 p 中变元 x 是自由的) 用项 t 去代换公式 p 中自由出现的个体变元 x 时，若在代换后的新公式里，t 的变元都是自由的，则说 t 对 p 中 x 是可自由代换的，简称 t 对 p 中 x 是可代换的，或简称 t 对 p 中 x 是自由的.

换句话说，用 t 代换 p 中自由出现的 x 时，若 t 中有变元在代换后受到约束，则说 t 对 p 中的 x 是“不自由的”(或“不可自由代换的，”或“不可代换的”).

下面两种情形，t 对 p 中 x 是自由的:

1° t 是闭项;

2° x 在 p 中不自由出现.

此外，在任何公式中，x_i (作为项) 对 x_i 自己总是自由的.

在 $\forall x_1 R_1^1(x_2)$ 中，项 $f_1^2(x_1, x_3)$ 对 x_2 是不自由的，而项 $f_1^2(x_4, x_3)$ 对 x_2 是自由的. 在 $\forall x_1 R_1^1(x_2)$ 中，任何不含 x_1 的项对 x_2 是自由的，任何含有 x_1 的项对 x_2 是不自由的.

在 $\forall x_1 R_1^2(x_1, x_2) \to \forall x_3 R_2^2(x_3, x_1)$ 中，$f_1^2(x_1, x_4)$ 对 x_2 是不自由的；$f_2^2(x_3, x_4)$ 对 x_2 是自由的，但对 x_1 是不自由的；$f_3^2(x_1, x_3)$ 对 x_1, x_2 都是不自由的；x_2 对 x_1 是自由的；x_1 对 x_2 是不自由的. 以上列出的项对该公式中 x_3, x_4, x_5 都是自由的，因为 x_3, x_4, x_5 在该公式中没有自由出现. (x_4, x_5 根本不出现.)

定义 3 的另一种说法是: 若对项 t 中所含任一变元 y, p 中自由出现的某变元 x 全都不出现在 p 中 $\forall y$ 的范围内，则说 t 对 p 中 x 是自由的.

以后如不另加说明，$p(t)$ 表示用项 t 去代换公式 $p(x)$ 中所有自由出现的变元 x 所得结果.

还要注意: 我们写 $p(x)$, 其中 x 是指该公式中自由出现的 x, 而不是指约束出现的 x. 写 $p(x)$ 时 x 可以不在 $p(x)$ 中自由出现或根本不出现，且不排除有其他变元在 $p(x)$ 中出现.

比如，当 $p(x_1) = R_1^1(x_1) \to \forall x_1 R_1^2(x_1, c_1)$ 时，$p(t) = R_1^1(t) \to \forall x_1 R_1^2(x_1, c_1)$; 当 $p(x_1) = R_1^1(x_2) \to \forall x_2 R_1^2(x_2, c_1)$ 或当 $p(x_1) = R_1^1(x_2) \to \forall x_1 R_1^2(x_1, c_1)$ 时，都有 $p(t) = p(x_1)$.

注 定义 3 中及别处所用“代换”一词，在本书中的含义是“全部替换”而不同于“一处替换”

这个定义较为复杂拗口，其中有许多有意思的点，这里一一解释：

①项 t 代换的是公式中自由出现的个体变元 x

正如前面解释闭式时所说，约束出现的变元就像公式的基础架构，而自由变元的改变可能会使公式的具体含义产生变化。我们用项对公式的某一变元进行代换生成新的公式，可能会改变公式的具体含义，但自然不会试图改变公式的基础架构，否则代换失去了意义。

②代换后的新公式中 t 的变元都是自由的，则称 t 对 p 中 x 自由

注意用来代换的 t 是一个项！当它参与公式构造前，其中的变元还并没有自由出现或约束出现一说。

依旧用一个简单的公式来举例：

$$\forall x_1 R_1^2(x_1, x_2)$$

在这个公式里，约束变元x_1搭建起了公式的基础架构，而自由出现的变元x_2并不受$\forall x_1$影响，就像一个自变量一样。

首先用项x_3代换$p(x_2)$中x_2，我们得到$\forall x_1 R_1^2(x_1, x_3)$，注意公式的基础架构并没有被破坏，它仍然描述这样一类内容：“一个数不管其取何值，均与另一个数满足关系 R”。

但当用项$f_1^2(x_1, x_3)$代换$p(x_2)$中x_2时，问题出现了：若此时用项 t 代换x_2得到 $p(t)$，我们的得到的是$\forall x_1 R_1^2(x_1, f_1^2(x_1, x_3))$，注意这里$f_1^2$参与公式构造后，其变元$x_1$受到了$\forall x_1$影响，变成了一个约束变元！实际上，公式的基础架构也已经随之改变了。这里所谓的“另一个数”与第一个数x_1有关，需要更细致的描述，是一种不同的基础架构。

因此，后续我们常要求代换后 t 的变元都是自由的，也即使用满足“t 对 p 中 x 自由”的项 t 来进行代换，否则代换失去了意义。

③几种特殊情况

在几种特殊情况下，t 对 p 中 x 自由：

1. t 是闭项

此时 t 中没有变元，代换后自然不会受约束。

2. x 在 p 中不自由出现

此时无自由变元 x 被代换，公式不产生实际变化，项 t 没有参与公式，其中变元自然不受到约束。

3. t 即 x 本身

注意 p 中仅有自由的 x 被代换，而用来代换 x 的项 t 即 x 本身，我们已知这些位置上 x 是自由的，项 t 中变元（即 x）不会受到约束。

④为什么要这样进行定义？

为什么要给出“t 对 p 中 x 自由”这样一个定义呢？

我们可以注意到，定义始终在维护公式的基础架构不变，我们用 p(x)表示一种基础架构的公式，同时允许有类似自变量的 x 存在，来使公式拥有架构类似但含义不同的变化，如果破坏了这种基础架构，公式未必就是错误的，但它失去了意义：我们完全可以用另一个符号 q(x)来表示，这就混淆了界限。

同时，这样的定义是为后续内容服务的。比如，学习谓词演算 K 的公理时，会学到(K4)：

(K4) $\forall x p(x) \to p(t)$, 其中项 t 对 $p(x)$ 中的 x 是自由的；

直觉上，这样的公理我们当然是可以接受的：对于 p(x)，如果 x 任意取值都使之成立，那么我们不改变公式的基础架构，也即使用满足“t 对 p 中 x 自由”的项 t 来代入，所得的 p(t)自然也成立。

最后以一个更为典型的例子来结尾，设 $p(x_1)$：

$$\exists x_2 R_1^2(x_1, x_2)$$

仍然考虑自然数集，R 为“大于”关系。

公式 p 具有这样的基础架构：“给定一个数，存在另一个数大于它”。直觉上我们可以知道，只要x_1不是一个与x_2有关的值，则无论x_1取值如何公式都成立，有$\forall x_1 \exists x_2\ R_1^2(x_1, x_2)$。

假如违反“t 对 p 中 x 自由”这个定义规则，用项x_2代换其中自由出现的x_1，则得到$\exists x_2\ R_1^2(x_2, x_2)$，它表示“存在一个数，它一定大于它本身”，基础架构产生改变，并且它是错误的！

究其原因，我们不妨先试着进行一下$\exists x_2\ R_1^2(x_1, x_2)$的直观证明：首先固定$x_1$的值，然后遍历$x_2$的所有可能值，只要有一个值使得关系成立，则式子成立。

注意跟踪观察这个过程！我们要先固定自由变元x_1的值，然后一次次遍历一个由约束变元x_2构造的基础架构，来判断正误。

如果代换不满足“t 对 p 中 x 自由”的定义，得到$\exists x_2\ R_1^2(x_2, x_2)$呢？注意这里$x_1$用$x_2$进行代换后，受到原公式中$\exists x_2$的约束，成为了基础架构的一部分，公式本质产生了变化！

所以，以下公式是违反(K4)定义的，不能由(K4)得到它（当然它本身也是错误的）：

$$\forall x_1 \exists x_2\ R_1^2(x_1, x_2) \rightarrow \exists x_2\ R_1^2(x_2, x_2)$$

在不涉及后续内容的基础上，我也只能尝试用较形象的语言给大家粗略解释一二。将来的学习过程中，大家对于谓词演算的了解将会越来越充分、全面，对于这些定义出发点的理解自然也就会更加深刻了。

By　贺阳槐安