何幻

Programming is about ideas,

languages are just a way to express them.

语言背后的代数学（三）：语义模型

2018-01-27 | Math

1. 回顾

上文我们从代数学角度重新认识了自然数，
认识了自然数是如何被编码为符号串的，
以及自然数在数学上是如何表示的。

我们的整体思路是，首先用公理化的方式建立一个形式系统，
然后为这个形式系统选择一种数学解释作为它的语义，
这样就建立了符号和数学对象之间的对应关系。

一般的，这些数学对象需要具有不同的运算性质，有不同的结构，
因此构成了不同的代数。

在《你好，类型》系列文章中，

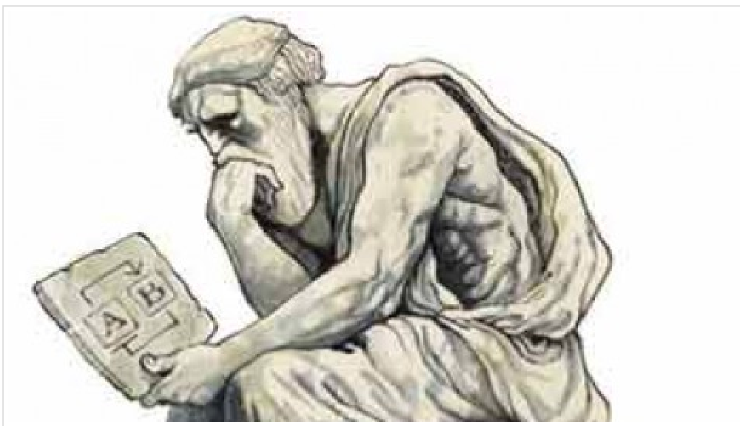
我们介绍了命题逻辑和一阶谓词逻辑，
当时，我们只是从形式系统（符号演算）的角度来介绍它们。

例如，我们只要知道公理和推导规则，就可以做出形式证明，
$\forall x(\alpha \leftrightarrow \beta) \vdash \forall x\alpha \leftrightarrow \forall x\beta$。

但是，这些符号到底代表什么含义呢？
我们当时故意没有提及。

本文从模型论角度来做出一些解释。

2. 一阶语言



首先让我们回顾以下一阶谓词逻辑有哪些符号构成，

（1）变元符号集合V，
它由可数个（包括0个）变元符号组成，用$x_1, x_1, \cdots, x_n, \cdots$表示。
（2）逻辑连接词符号集合C，
它由逻辑连接词符号$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$组成。
（3）量词符号集合Q，包括$\forall, \exists$。
（4）等词符号集合E，只包括一个符号$\doteq$。
（5）括号集合，包括$), ($。

以上这些符号称为**逻辑符号**，每个**一阶逻辑**都有这些符号。
而不同的一阶逻辑，还有属于自己的**非逻辑符号**。

（1）常元符号集合$\mathscr{L}_c$，
它由可数个（包括0个）常元符号组成，用$c_1, c_2, \cdots$表示。
（2）函数符号集合$\mathscr{L}_f$，
它由可数个（包括0个）函数符号组成，用$f_1, f_2, \cdots$表示。

（3）谓词符号集合$\mathscr{L}_P$，

它由可数个（包括0个）谓词符号组成，用$P_1, P_2, \cdots$表示。

等词符号$\doteq$实际上可以看做是一个谓词符号。

因此，一阶谓词逻辑是一种一阶逻辑。

一阶逻辑中的逻辑符号和非逻辑符号，称为**一阶语言**，记为$\mathscr{L}$。

3. 初等算术语言

初等算术语言是一个一阶语言，记为Π。

它的常元符号集合为$\{0\}$，函数符号集合为$\{S, +, \cdot\}$，

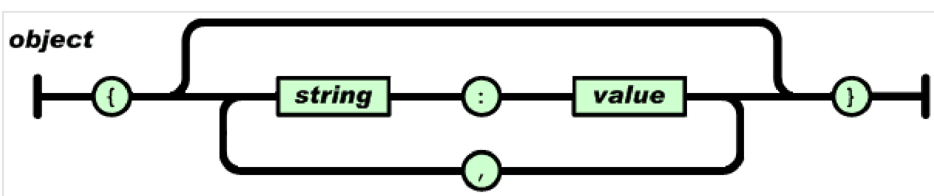
谓词符号集合为$\{<\}$。

其中，S可以表示算术中的后继函数，

而二元函数符号$+$和$\cdot$可以分别表示算术中的加法和乘法，

谓词符号$<$可以描述自然数之间的小于关系。

4. 语法项和逻辑公式



从形式语言的角度来看，除了知道语言包含哪些符号之外，还要指定语法，

习惯上，我们经常使用BNF来指定，

$t ::= c|x|ft_1 \cdots t_n$

即一个合法的**项**，可以归纳定义为，

（1）每一个常元都是合法的项，

（2）每一个变元都是合法的项，

（3）如果$t_1, t_2, \cdots, t_n$都是合法的项，而f是一个n元函数符号，

那么$ft_1 \cdots t_n$也是一个合法的项。

初等算术语言Π中的合法项，
以下符号串都是合法的，
$S0$，Sx_1，$+S0SSx$，$\cdot x_1 + Sx_1x_2$，
而$SS <$是不合法的。

我们知道逻辑证明，并不是建立在形式语法之上的，
而是建立在公理系统上面，而每一个推导规则都表明了前提和结论之间的关系，
这些前提和结论，称为**逻辑公式**。

一阶语言$\mathscr{L}$中的逻辑公式，用大写字母$A, B, \cdots$表示，定义为，
$A ::= t_1 \doteq t_2 | Rt_1 \cdots t_n | \neg A | A \wedge B | A \vee B | A \to B | A \leftrightarrow B | \forall x A | \exists x A$

即，逻辑公式可以归纳的定义为，
（1）如果t_1和t_2是合法的项，则$t_1 \doteq t_2$是公式，
（2）如果$t_1, ..., t_n$是合法的项，而R是一个n元谓词，则$Rt_1 \cdots t_n$是公式，
（3）如果A是公式，则$\neg A$是公式，
（4）若A, B是公式，则$A \wedge B, A \vee B, A \to B, A \leftrightarrow B$都是公式，
（5）若A是公式并且x是一个变元，那么$\forall x A$和$\exists x A$也是公式，x称为**约束变元**。

例，以下符号串可以看做一个初等算术公式，
$\forall x \neg (Sx \doteq 0), \forall x \forall y (< xy \to (\exists (y \doteq +xz)))$

5. 语义模型

有了一阶语言之后，我们就可以为符号选择语义了，
通常的，语言的语义有两部分组成，
其一称为**结构**，用来解释常元符号，函数符号和谓词符号，
其二称为**赋值**，用来解释变元符号。

5.1 语言结构

一阶语言$\mathscr{L}$的**结构**M是一个偶对，记为$M=(\mathbb{M},I)$，其中，
（1）$\mathbb{M}$是一个非空集合，称为**论域**，
（2）I是从$\mathscr{L}$到$\mathbb{M}$的映射，称为**解释**，记为$I:\mathscr{L}\to\mathbb{M}$，它满足下面三个条件

a. 对$\mathscr{L}$中的每一个常元符号c，$I(c)$是$\mathbb{M}$中的元素
b. 对$\mathscr{L}$中的每一个n元函数符号f，$I(f)$是$\mathbb{M}$上的n元函数
c. 对$\mathscr{L}$中的每一个n元谓词符号P，$I(P)$是$\mathbb{M}$上的一个n元关系

例，我们可以指定初等算术语言Π的结构为，偶对$N=(\mathbb{N},I)$，
其中论域$\mathbb{N}$为自然数集，
$I(S)$为自然数集上的加1函数，$I(+)$为自然数加法运算，$I(\cdot)$为自然数乘法运算。
$I(<)$为自然数集上的小于关系。

5.2 变元赋值

赋值σ是一个映射，$\sigma:V\to\mathbb{M}$，
它将$\mathscr{L}$中的每一个变元，赋以论域$\mathbb{M}$中的一个元素a，
记为$\sigma(x)=a$，其中$x\in V,a\in\mathbb{M}$。

有了赋值运算之后，公式中的变元就固定下来了，
我们就可以谈论，在某一指定赋值运算下公式的语义了。

5.3 模型和语义

给定一阶语言$\mathscr{L}$，并指定结构M和赋值σ，

我们称(M,σ)是，我们为语言$\mathscr{L}$选择的一个**模型**。

项的语义

选择了模型(M,σ)之后，

$\mathscr{L}$中的合法项t的**语义**，

就可以归纳的定义为论域$\mathbb{M}$中的元素了，记为$t_{M[\sigma]}$。

（1）$x_{M[\sigma]}=\sigma(x)$，x为变元符号

（2）$c_{M[\sigma]}=c_M$，c为常元符号

（3）$(ft_1\cdots t_n)_{M[\sigma]}=f_M((t_1)_{M[\sigma]},\cdots(t_n)_{M[\sigma]})$

例，初等算术Π中项的语义，

$(+x_1Sx_7)_{N[\sigma]}=(x_1)_{N[\sigma]}+(Sx_7)_{N[\sigma]}=1+((x_7)_{N[\sigma]}+1)=1+(7+1)=9$

逻辑公式的语义

公式A在模型(M,σ)下的语义是一个真假值，用$A_{M[\sigma]}$表示，归纳定义如下，

（1）$(Pt_1\cdots t_n)_{M[\sigma]}=P_M((t_1)_{M[\sigma]},\cdots,(t_n)_{M[\sigma]})$

（2）$(t_1\doteq t_2)_{M[\sigma]}=\begin{cases}T, & \text{if } (t_1)_{M[\sigma]}=(t_2)_{M[\sigma]} \\ F, & \text{otherwise}\end{cases}$

（3）$(\neg A)_{M[\sigma]}=B_\neg(A_{M[\sigma]})$

（4）$(A\vee B)_{M[\sigma]}=B_\vee(A_{M[\sigma]},B_{M[\sigma]})$

（5）$(A\wedge B)_{M[\sigma]}=B_\wedge(A_{M[\sigma]},B_{M[\sigma]})$

（6）$(A\to B)_{M[\sigma]}=B_\to(A_{M[\sigma]},B_{M[\sigma]})$

（7）$(A\leftrightarrow B)_{M[\sigma]}=B_\leftrightarrow(A_{M[\sigma]},B_{M[\sigma]})$

（8）$(\forall x_iA)_{M[\sigma]}=\begin{cases}T, & \forall a\in M, A_{M[\sigma[x_i:=a]]}=T \\ F, & \text{otherwise}\end{cases}$

（9）$(\exists x_iA)_{M[\sigma]}=\begin{cases}T, & \exists a\in M, A_{M[\sigma[x_i:=a]]}=T \\ F, & \text{otherwise}\end{cases}$

其中，二元真值函数$B_\wedge,B_\vee,B_\to,B_\leftrightarrow$，

分别逻辑连接词符号$\wedge,\vee,\to,\leftrightarrow$的语义。

至此，我们通过为一阶语言指定模型，
将语言中所有的符号串都进行了解释。

6. 总结

本文以一阶逻辑为例，从逻辑学角度给出了语义模型的定义，
由此，一阶逻辑系统中的符号串，都有了一个数学对象与之对应，
它们是论域，论域集合上的函数和运算。

可想而已，这些数学对象是有代数性质的，
下文我们将继续深入了解。

参考

你好，类型（五）：Predicate logic
数理逻辑
一阶逻辑

语言背后的代数学（二）：初等代数
语言背后的代数学（四）：哥德尔定理