

指称语义

将程序和与之相关的数学对象建立起联系，这样的语义就称为指称语义（Denotational Semantics）。

相比霍尔逻辑这样的公理化语义，它是更细粒度的（Fine-grained），而后面讲的操作语义（Operational Semantics）还要更细粒度。

指称

说是指称语义，指称是什么？简单来说，一个表达式的**指称**（Denotation），就是和其相联系的数学对象。这也反映在指称语义的一个别名，Mathematical Semantics 上。

更正式的说，一个指称是一个将表达式映射到**域**（Domain）的函数。这里的域就是某种数学对象。

下面分别讨论程序的几个要素的指称。

算术表达式的指称

对于一个算术表达式 a ，可以将其的指称定义为一个函数： $\text{aeval} : \text{aexp} \mapsto \text{state} \mapsto \mathbb{Z}$ 。即对于指定的程序状态，返回该表达式在该状态下求值的结果。这个函数可以在语法树上递归定义。

程序状态是一个将程序变量映射到具体值的全函数。也可以理解成是一个列表，列表中的元素都是程序变量、值的二元组。

布尔表达式的指称

对于一个布尔表达式 b ，可以将其的指称定义为一个函数： $\text{beval} : \text{bexp} \mapsto \mathcal{P}(S)$ ，其中 S 是所有程序状态的集合， $\mathcal{P}(S)$ 是 S 所有子集的集合（即幂集）。即对于指定的布尔表达式，返回其在哪些程序状态下为真。这个函数也是在语法树上递归定义实现的。

在 Coq 下，一个集合 A 的子集可以用一个函数 $A \mapsto \text{Prop}$ 表示，即**使用一个命题来替代集合本身**。因此， beval 的类型就是 $\text{bexp} \mapsto \text{state} \mapsto \text{Prop}$ 。

程序指令的指称

对于一个程序指令 c ，可以将其的指称定义为程序状态二元组的集合（或者说，程序状态的二元关系），即 $\text{ceval} : \text{com} \mapsto \mathcal{P}(S \times S)$ 。即对于程序指令 c ， (st_1, st_2) 在其指称内当且仅当在 st_1 下运行 c 会到达程序状态 st_2 。

因此， ceval 的类型就是 $\text{com} \mapsto \text{state} \mapsto \text{state} \mapsto \text{Prop}$ 。

程序指令的指称构造较之前两者稍显复杂。下面讨论几个比较有意思的程序指令的指称。

赋值语句

如果 $(st_1, st_2) \in \text{ceval}(X := E)$ ，那么需要满足的是：

- st_2 下 X 的值等于 st_1 下 E 的值。
- 除了 X ，对于其他任何程序变量 Y ，其在 st_1, st_2 下取值相同。

条件分支语句

为了表现出条件判断这一步，可以构造一个测试关系 $S_b = \{(st, st) : b \text{ is true on } st\}$ 。

现在, 如果 $(st_1, st_2) \in \text{ceval}(\text{If } b \text{ Then } c_1 \text{ Else } c_2 \text{ EndIf})$, 那么:

1. 要么 $(st_1, st_2) \in S_b \circ \text{ceval}(c_1)$ 。
2. 要么 $(st_1, st_2) \in S_{\neg b} \circ \text{ceval}(c_2)$ 。

循环语句

循环语句比较有趣。我们知道, 如果 $\text{While } b \text{ Do } c \text{ EndWhile}$ 执行了 n 次, 那么

$$\underbrace{\text{While } b \text{ Do } c \text{ EndWhile}}_{n \text{ times}} = \text{If } b \text{ Then } \underbrace{\text{While } b \text{ Do } c \text{ EndWhile}}_{n-1 \text{ times}} \text{ Else Skip EndIf}$$

等号表示这两者的指称应当是一样的。

于是对于给定的 n , 可以利用条件分支语句, 递归构造这样一个关系 R_n 。那么循环语句的指称就是对所有的这些 n 对应的关系取并集。用命题来表示这个命题就是

$$(st_1, st_2) \in \text{ceval}(\text{While } b \text{ Do } c \text{ EndWhile}) \iff \exists n, (st_1, st_2) \in R_n$$