STLC 开始的类型论入门

习题答案

Jul 03, 2023

Alias Qli
alias@qliphoth.tech

练习 3.3.1: 使用相继式演算的语法形式化地定义谓词 $x : A \in \Gamma$。提示: 对语境的结构进行归纳.

答案:

$$\frac{}{x : A \in \Gamma, x : A} \qquad \frac{x : A \in \Gamma}{x : A \in \Gamma, y : B}$$

练习 3.3.1: 写出空语境下 $\lambda f\, g\, x.\, f(g\, x) : (B \to C) \to (A \to B) \to A \to C$ 的完整类型检查过程。

答案: 记 $\Gamma = f : B \to C, g : A \to B, x : A$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{f : B \to C \in \Gamma}{\Gamma \vdash f : B \to C}\ \text{Var} \quad \dfrac{\dfrac{g : A \to B \in \Gamma}{\Gamma \vdash g : A \to B}\ \text{Var} \quad \dfrac{x : A \in \Gamma}{\Gamma \vdash x : A}\ \text{Var}}{\Gamma \vdash g\, x : B}\ \text{App}}{\Gamma \vdash f(g\, x) : C}\ \text{App}}{f : B \to C, g : A \to B \vdash \lambda x.\, f(g\, x) : A \to C}\ \text{Lam}}{f : B \to C \vdash \lambda g\, x.\, f(g\, x) : (A \to B) \to A \to C}\ \text{Lam}}{\emptyset \vdash \lambda f\, g\, x.\, f(g\, x) : (B \to C) \to (A \to B) \to A \to C}\ \text{Lam}$$

练习 3.6.1: 将以下 STLC 项归约到既约形式:

1. $(\lambda x.\, \lambda y.\, x)\, y$
2. $\lambda(f : A \to B \to C).\, f$

答案:

1.
$$\begin{aligned} & (\lambda x.\, \lambda y.\, x)\, y \\ \equiv\ & (\lambda x.\, \lambda z.\, x)\, y && (\alpha \text{ conversion}) \\ \equiv\ & (\lambda z.\, x)[x \mapsto y] && (\beta \text{ reduction}) \\ \equiv\ & \lambda z.\, y && (\text{substitution}) \end{aligned}$$

2. $$\begin{aligned} & \lambda(f : A \to B \to C).\, f \\ \equiv\; & \lambda(f : A \to B \to C).\, \lambda(x : A).\, f\, x && (\eta \text{ expand } f : A \to (B \to C)) \\ \equiv\; & \lambda(f : A \to B \to C).\, \lambda(x : A).\, \lambda(y : B).\, f\, x\, y && (\eta \text{ expand } f\, x : B \to C) \end{aligned}$$

推论 4.2: 任何语境下都有 $\mathrm{yes} \not\equiv \mathrm{no} : \mathrm{Ans}$.

命题 4.3: 无法对每个类型 A 都定义函数 $\mathrm{choose} : \mathrm{Ans} \to A \to A \to A$, 使得 $\mathrm{choose}\ \mathrm{yes}\ a\, b \equiv a$, $\mathrm{choose}\ \mathrm{no}\ a\, b \equiv b$.

练习 5.1.1: 构造一个 $[\![\mathrm{yes}]\!] = [\![\mathrm{no}]\!]$ 的模型, 证明上述命题. 注意, 这和推论 4.2 不矛盾. 想一想为什么.

答案: 我们的 STLC 中的类型太少了, 这里加一个临时的类型 Ans', 有 $\mathrm{yes}', \mathrm{no}' : \mathrm{Ans}'$. 集合模型中相应的构造和 Ans 相同.

在集合模型的基础上, 做修改

$$\begin{aligned} [\![\mathrm{Ans}]\!] &= \{*\} \\ [\![\Gamma \vdash \mathrm{yes} : \mathrm{Ans}]\!] &= - \mapsto * \\ [\![\Gamma \vdash \mathrm{no} : \mathrm{Ans}]\!] &= - \mapsto * \end{aligned}$$

得到的就是一个满足 $[\![\mathrm{yes}]\!] = [\![\mathrm{no}]\!]$ 的模型. 显然这个模型是定义良好的.

给定 $[\![\Gamma \vdash x : \mathrm{Ans}]\!], [\![\Gamma \vdash a, b : A]\!]$, 我们尝试在模型中构造 $[\![\Gamma \vdash \mathrm{choose}\ x\, a\, b : A]\!]$, 满足

$$\frac{[\![\Gamma \vdash x : \mathrm{Ans}]\!] = [\![\Gamma \vdash \mathrm{yes} : \mathrm{Ans}]\!]}{[\![\Gamma \vdash \mathrm{choose}\ x\, a\, b : A]\!] = [\![\Gamma \vdash a : A]\!]}$$

$$\frac{[\![\Gamma \vdash x : \mathrm{Ans}]\!] = [\![\Gamma \vdash \mathrm{no} : \mathrm{Ans}]\!]}{[\![\Gamma \vdash \mathrm{choose}\ x\, a\, b : A]\!] = [\![\Gamma \vdash b : A]\!]}$$

然而 $[\![\Gamma \vdash \mathrm{yes} : \mathrm{Ans}]\!] = - \mapsto * = [\![\Gamma \vdash \mathrm{no} : \mathrm{Ans}]\!]$, 因此 $[\![\Gamma \vdash a : A]\!] = [\![\Gamma \vdash b : A]\!]$. 取 $A = \mathrm{Ans}'$, $a = \mathrm{yes}'$, $b = \mathrm{no}'$, 就有 $(- \mapsto 1) = (- \mapsto 0)$, 矛盾. 因此不存在这样的函数 choose.

最后说明为什么这与推论 4.2 不矛盾.推论 4.2 说的是在语法中 $\mathrm{yes} \equiv \mathrm{no}$ 不成立, 不妨碍存在一个模型, 性质比语法丰富, 使得 $\mathrm{yes} \equiv \mathrm{no}$ 成立. 模型中成立的性质在语法中不必要成立.

练习 5.1.2: 构造满足 $[\![\mathrm{true}]\!] = [\![\mathrm{false}]\!]$ 的所有模型.

答案: 为了满足计算规则, 模型中一定有

$$\frac{[\![\Gamma \vdash b : \mathbb{B}]\!] = [\![\Gamma \vdash \mathrm{true} : \mathbb{B}]\!]}{[\![\Gamma \vdash \mathrm{elim}_{\mathbb{B}}(b, c_t, c_f) : C]\!] = [\![\Gamma \vdash c_t : C]\!]}$$

$$\frac{[\![\Gamma \vdash b : \mathbb{B}]\!] = [\![\Gamma \vdash \mathrm{false} : \mathbb{B}]\!]}{[\![\Gamma \vdash \mathrm{elim}_{\mathbb{B}}(b, c_t, c_f) : C]\!] = [\![\Gamma \vdash c_f : C]\!]}$$

如果 $[\![\mathrm{true}]\!] = [\![\mathrm{false}]\!]$, 那么对任意的 $\Gamma \vdash c_t, c_f : C$, 都有 $[\![\Gamma \vdash c_t : C]\!] = [\![\Gamma \vdash c_f : C]\!]$, 这里 C 是任意类型. 因此, 满足 $[\![\mathrm{true}]\!] = [\![\mathrm{false}]\!]$ 的是这样的模型: 所有相同语境下相同类型的项在模型中都相等.

练习 5.3.1: 给出余积类型的计算规则. 唯一性规则比较复杂, 不作要求.

答案:

$$\frac{a : A \quad c_l : A \to C \quad c_r : B \to C}{\mathrm{elim}_{A+B}(\mathrm{inl}(a), c_l, c_r) \equiv c_l\ a : C} \quad \frac{b : A \quad c_l : A \to C \quad c_r : B \to C}{\mathrm{elim}_{A+B}(\mathrm{inr}(b), c_l, c_r) \equiv c_r\ b : C}$$

练习 5.4.1: 定义自然数上的乘法.

答案:

$$\begin{aligned} &\mathrm{mul} : \mathbb{N} \to \mathbb{N} \to \mathbb{N} \\ &\mathrm{mul} :\equiv \lambda m\, n.\ \mathrm{elim}_{\mathbb{N}}(m, \mathrm{zero}, \lambda \_.\ \lambda m'.\ \mathrm{add}\ n\ m') \end{aligned}$$

$\mathrm{add} : \mathbb{N} \to \mathbb{N} \to \mathbb{N}$ 是讲义中定义的自然数加法.

练习 5.5.1: 将练习 5.4.1 中定义的自然数乘法改写成模式匹配的形式.

答案:

$$\begin{aligned} &\mathrm{mul}\ \mathrm{zero} \quad\quad\ n :\equiv \mathrm{zero} \\ &\mathrm{mul}\ \mathrm{suc}(m)\ n :\equiv \mathrm{add}\ n\ (\mathrm{mul}\ m\ n) \end{aligned}$$