构造演算（The Calculus of Constructions）

读书笔记

许博

1 λC 系统

λC 组合了第二章到第五章中介绍的系统，拥有四种可能的选择，即依赖于项/类型的项/类型。

λP 与 λC 只有一处不同，但足以扩展 λP 到 $\lambda\text{C} = \lambda 2 + \lambda\underline{\omega} + \lambda\text{P}$：

$$(form_{\lambda\text{P}})\ \frac{\Gamma \vdash A : * \quad \Gamma, x : A \vdash B : s}{\Gamma \vdash \Pi x : A.B : s}$$

在这条规则中，关键点是 $A : *$，为了保证类型 $\Pi x : A.B$ 的成员（inhabitant）是项或者依赖于项的类型。但在舍弃了这个限制之后，我们就获得了我们想要的泛化：依赖于项/类型的项/类型。

看起来将 $A : *$ 替换为 $A : s$，其中 s 为 $*$ 或 $\square$，就足够了，但是规则中已经出现了 s，观察 $\lambda\underline{\omega}$ 的 $(form)$-规则：

$$(form_{\lambda\underline{\omega}})\ \frac{\Gamma \vdash A : s \quad \Gamma \vdash B : s}{\Gamma \vdash A \to B : s}$$

只能表示依赖于项的项和依赖于类型的类型，而不能相互交叉（cross-over）。

因此在 λC 的 $(from)$-规则中，使用了两个 s：s_1 和 s_2：

$$(form_{\lambda\text{C}})\ \frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash \Pi x : A.B : s_2}$$

$\Pi x : A.B$ 的类型继承自 B，也即依赖于项/类型（1）的项/类型（2）依然是项/类型（与 2 统一）。因此有一个有趣的事实是：假设 A 中不存在与

抽象的类型变量相同的自由类型变量，$* \to A$ 是一个类型，而 $A \to *$ 是一个种类 (kind)。