λD 中的数学

Mathematics in λD

报告人：许博

目录

引入类型构造子

$\lambda\!\to$

$\lambda x : \alpha . x$

$\lambda 2$, $\lambda\!\to$ + terms depending on types

$\lambda \alpha : * . \lambda x : \alpha . x$

$(\lambda \alpha : * . \lambda x : \alpha . x)\beta \to_\beta \lambda x : \beta . x$

$\lambda \underline{\omega}$, $\lambda\!\to$ + types depending on types

$\beta \to \beta, \gamma \to \gamma, \ldots$ 等具有结构 $\Diamond \to \Diamond$ 的类型，其中箭头的左右是一样的类型

$\lambda \alpha : * . \alpha \to \alpha$，类型构造子

$(\lambda \alpha : * . \alpha \to \alpha)\beta \to_\beta \beta \to \beta$

类型构造子的类型，超级类型

类型构造子的类型

$\alpha : *$
$\alpha \to \alpha : *$
$\lambda\alpha : *.\alpha \to \alpha : * \to *$
$\lambda\alpha : *.\lambda\beta : *.\alpha \to \beta : * \to (* \to *)$

超级类型，记为类 (kind)，$\mathbb{K}$

$*, * \to *, * \to (* \to *), \ldots$
$\mathbb{K} = *|(\mathbb{K} \to \mathbb{K})$

超级超级类型，$\square$，仅有一个

$* : \square, * \to * : \square, * \to (* \to *) : \square, \ldots$

类型构造子

定义 4.1.4（构造子，真构造子，类别）

(1) 如果 $\kappa : \square$ 且 $M : \kappa$，则 M 是一个构造子，如果 $\kappa \neq *$，则 M 是一个真构造子
(2) 类别（s）的集合为 $\{*, \square\}$

类型构造子的定义

如果 κ 是一个类 (kind)，则对于每个类型是 κ 的 M(即 $M : \kappa$)，M 被称作是一个类型构造子，简称为构造子
因为 $* : \square, \alpha : *$，所以 α 或者 $\alpha \to \alpha$ 也都是构造子

真构造子

使用真构造子（proper constructor）表示不是类型的构造子（即类型不是 $*$ 的构造子）。

类别 (sort), s

使用类别 (sort) 或符号 s 表示 $*$ 或 $\square$。

四个层级 (level)

定义 4.1.6（层级，level）

第 1 层：项（terms）；
第 2 层：构造子（包括类型和真构造子）；
第 3 层：类（kinds）；
第 4 层：超级超级类型 $\square$。

对于语句 $A : B$，可以得出 B 所处的层级一定比 A 高一级，比如当 A 是一个项时，B 是一个类型，或者 A 是一个类型时，$B \equiv *$。

类别规则

定义 4.2.1（类别规则，sort-rule）

(sort) $\emptyset \vdash * : \square$

形式化 $*$ 的类型是 $\square$，$* : \square$ 可以由空的上下文推导出。

良构 (well-formed) 的类型定义

给定的上下文中所有的声明都需要是可推导的。

定义 2.4.5（$\lambda\!\rightarrow$ 的 var-rule）

如果 $x : \sigma \in \Gamma$，则 $\Gamma \vdash x : \sigma$

对于 $\lambda\!\rightarrow$ 上下文中出现的类型，合法类型的集合已经给出，可以直接判断合法与否。

定义 3.4.6（$\lambda 2$ 的 var-rule）

如果 Γ 是一个 $\lambda 2$-上下文且 $x : \sigma \in \Gamma$，则 $\Gamma \vdash x : \sigma$

$\lambda 2$ 的类型复杂一点，需要类型中的自由类型变量在上下文中已声明，因此类型的合法与否依赖于推定的上下文。

$\lambda\underline{\omega}$ 中的变量规则

定义 4.2.2（变量规则，var-rule）

(var) 如果 $x \notin \Gamma$，则 $\dfrac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}$

推定中的类型的合法与否取决于是否可以被形式化地推导出。
$x \notin \Gamma$ 保证了变量 x 未在 Γ 中出现，因此声明在一个上下文中的所有的变量都是不同的，避免了变量名相同（类型不同）时造成的混淆。

应用 (sort) 和 (var) 规则的一个例子

$$\cfrac{\cfrac{(1)\ \emptyset \vdash * : \Box}{(2)\ \alpha : * \vdash \alpha : *}\ (var)}{(3)\ \alpha : *, x : \alpha \vdash x : \alpha}\ (var)$$

只能推导出上下文中新添加的最后一个声明。

不能推导的例子

$\alpha : *, x : \alpha \vdash \alpha : *$

$\alpha : *, \beta : * \vdash \alpha : *$

$\alpha : *, \beta : * \vdash \beta : *$

最后一个例子中，不能得到 **premiss**： $\alpha : * \vdash * : \Box$

$\lambda\underline{\omega}$ 中的弱化规则 (weakening rule)

定义 4.3.1（弱化规则，weakening rule）

(weak) 如果 $x \notin \Gamma$，则 $\dfrac{\Gamma \vdash A : B \quad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B}$

若上下文 Γ 已经可以推导出 $A : B$，在 Γ 的尾部添加了一个任意的声明（弱化）后仍然可以推导出 $A : B$。需要注意的是添加的声明的类型需要是可推导出的。

推导 $\alpha : *, x : \alpha \vdash \alpha : *$

$$\dfrac{\dfrac{(1)\ \emptyset \vdash * : \square}{(2)\ \alpha : * \vdash \alpha : *}(var) \qquad \dfrac{(1)\ \emptyset \vdash * : \square}{(2)\ \alpha : * \vdash \alpha : *}(var)}{(3)\ \alpha : *, x : \alpha \vdash \alpha : *}(weak)$$

定义 4.4.1（形成规则，formation rule）

$$\text{(form)} \ \frac{\Gamma \vdash A : s \quad \Gamma \vdash B : s}{\Gamma \vdash A \to B : s}$$

定义 3.4.7（$\lambda\!\!\to$ 中的形成规则，formation rule in $\lambda\!\!\to$）

(form) 如果 Γ 是一个 $\lambda 2$-上下文，$B \in \mathbb{T}2$ 且 B 中所有的自由类型变量已经在 Γ 中声明，则 $\Gamma \vdash B : *$

三个 s 同时表示 * 或 □。
在引入新的 (form) 规则之前，由 (var) 规则与 (weak) 规则只能推导出 $* : \square, \alpha : *, x : \alpha$ 而不能推出箭头类型 $* \to * : \square$ 以及 $\alpha \to \beta : *$ 等。

例子

$$\frac{\emptyset \vdash * : \square \quad \emptyset \vdash * : \square}{\emptyset \vdash * \to * : \square} \ (form)$$

应用规则，application rule

$$(\text{appl})\ \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}$$

抽象规则，abstraction rule

$$(\text{abst})\ \frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash A \to B : s}{\Gamma \vdash \lambda x : A.M : A \to B}$$

$\lambda\underline{\omega}$ 中的这两个规则与之前略有不同：

- 用于类型的元变量（meta-variable）的名字不同，因为 $\lambda\underline{\omega}$ 中的类型更为通用
- 需要保证类型是良构的（即可以由上下文推导出）

当 $s \equiv *$ 时，$A \to B$ 是第二层的类型，如 $(\alpha \to \beta) \to \gamma$，
当 $s \equiv \square$ 时，$A \to B$ 就是一个第三层的类 (kind)，如 $(* \to *) \to *$。

简化 (shortened) 推导

推导 $\alpha : *, \beta : * \vdash \alpha \to \beta : *$，需要以下推定：

$\emptyset \vdash * : \square$ (sort),
$\alpha : * \vdash \alpha : *$ (var),
$\alpha : * \vdash * : \square$ (weak),
$\alpha : *, \beta : * \vdash \alpha : *$ (weak),
$\alpha : *, \beta : * \vdash \beta : *$ (var)。

允许跳过以下推导过程：
(i) 当使用规则 (sort),(var) 和 (weak) 时，
(ii) 当使用规则 (form) 时，以及
(iii) 当确定 (abst) 规则的第二个 **premiss** 的合法性时。
因此 $\alpha : *, \beta : * \vdash \alpha \to \beta : *$ 现在可以直接使用。

定义 4.7.1（变换规则，conversion rule）

(conv) 如果 $B =_\beta B'$，则 $\dfrac{\Gamma \vdash A : B \quad \Gamma \vdash B' : s}{\Gamma \vdash A : B'}$

B 作为已知推定 $\Gamma \vdash A : B$ 的部分是良构的，$\Gamma \vdash B' : s$ 保证 B' 也是良构的。而 $B =_\beta B'$ 无法保证 B' 是良构的，比如 $\beta \to \gamma =_\beta (\lambda\alpha : *.\beta \to \gamma)M$。

例子

令 $\Gamma \equiv \beta : *, x : (\lambda\alpha : *.\alpha \to \alpha)\beta$：

$$\frac{\Gamma \vdash x : (\lambda\alpha : *.\alpha \to \alpha)\beta \quad \Gamma \vdash \beta \to \beta : *}{\Gamma \vdash x : \beta \to \beta} \text{(conv)}$$

$\lambda\underline{\omega}$-规则

(sort) $\emptyset \vdash * : \square$

(var) 如果 $x \notin \Gamma$，则 $\dfrac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}$

(weak) 如果 $x \notin \Gamma$，则 $\dfrac{\Gamma \vdash A : B \quad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B}$

(form) $\dfrac{\Gamma \vdash A : s \quad \Gamma \vdash B : s}{\Gamma \vdash A \to B : s}$

(appl) $\dfrac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}$

(abst) $\dfrac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash A \to B : s}{\Gamma \vdash \lambda x : A.M : A \to B}$

(conv) 如果 $B =_\beta B'$，则 $\dfrac{\Gamma \vdash A : B \quad \Gamma \vdash B' : s}{\Gamma \vdash A : B'}$

$\lambda\underline{\omega}$ 的性质

$\lambda\underline{\omega}$ 满足之前几章所提到的大部分性质，但是类型唯一性引理需要进行调整：

引理 4.8.1（类型唯一性引理）

如果 $\Gamma \vdash A : B_1$ 且 $\Gamma \vdash A : B_2$，则 $B_1 =_\beta B_2$

引理 2.10.9（适用于 $\lambda\!\to$ 的类型唯一性引理）

假设 $\Gamma \vdash M : \sigma$ 且 $\Gamma \vdash M : \tau$，则 $\sigma \equiv \tau$