

simply-typed λ -calculus

myuon

2017 年 10 月 13 日

目次

第 I 部	Syntax	1
1	Types	2
2	Simple Extension	2
2.1	Ground Type	2
2.2	Product Type	3
2.3	Unit Type	3
2.4	Abbreviations	3
第 II 部	Semantics	3
3	Equational Theory	3
3.1	λ^{\rightarrow}	3
3.2	$\lambda^{\times,1}$	4
4	Categorical Semantics	4
4.1	Syntactic category	4
4.2	Categorical Model	6
4.3	Soundness	7
4.4	Completeness	8
4.5	Logical Formalism	8
4.6	Categorical Semantics	9

概要

型と項の関係は、集合とそこに属する元の関係に似ている。型を導入することで、項がどんな集合に属しているかを表現することができる。型なしラムダ計算の項は自由生成されていたのでどのような項でも許されていたが、これには通常関数としての意味付けができないような異常な項も含まれている (例えば $f(f)$ など)。型システムを導入することの利点の 1 つとして、このような異常な項を含まない正常な項だけを考えることで型付きラムダ計算は正規化性などの“よい”性質をもつようにできる。

第 I 部

Syntax

型付き λ 計算の中で最も単純なものが単純型付き λ 計算 (λ^\rightarrow) である。

1 Types

定義 1 (型). 単純型付き λ 計算の型を, 次のように定義する. これを $\text{Typ}(\lambda^\rightarrow)$ などとかく.

$$\alpha ::= v \mid \alpha_1 \rightarrow \alpha_2$$

ただし v とは型変数であり, ここでは適当な型変数の集合 V_t が与えられているとする.

$\alpha_1 \rightarrow \alpha_2$ のことを関数型 (**function type**) などとよぶ. これはまさに関数を表現するためのものであり, 写像 $A \rightarrow B$ などと書いたりするのと同じことを表していると思えばよい.

さて, 項が型に属している ($x \in A$) ことを表明する関係を定義する.

定義 2 (型判断 (typing judgement)). 関係 $\vdash : \subseteq (V_t \times \text{Typ}) \times \Lambda \times \text{Typ}$ を以下のように定義する.

$$\begin{array}{c} \text{(var)} \frac{x : A \in \Gamma}{\Gamma \vdash x : A} \quad \text{(abs)} \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x. M : A \rightarrow B} \quad \text{(app)} \frac{\Gamma \vdash M_1 : A \rightarrow B \quad \Gamma \vdash M_2 : A}{\Gamma \vdash M_1 M_2 : B} \end{array}$$

図 1 simply-typed typing rules

- Γ を文脈 (**context**) と呼ぶ.
 - 関係 $\Gamma \vdash M : A$ が成り立っているとき, Γ の下で M の型は A である, などという.
 - 項 M に対して judgement $\Gamma \vdash M : A$ が存在する時, M は **well-typed** であるという.

補題 3 (coherence).

1. $\Gamma \vdash x : A$ ならば $x : A \in \Gamma$.
 - (a) $\Gamma \vdash \lambda x. M : A$ ならば, A_1, A_2 が存在して, $A = A_1 \rightarrow A_2$ かつ $\Gamma, x : A_1 \vdash M : A_2$.
 - (b) $\Gamma \vdash M_1 M_2 : B$ ならば, A が存在して, $\Gamma \vdash M_1 : A \rightarrow B$ かつ $\Gamma \vdash M_2 : A$.

Proof. typing judgement の導出に関する場合分けにより明らか. □

2 Simple Extension

2.1 Ground Type

λ^\rightarrow の型以外に, 基本となる組み込みの型を追加する. このような型を base type や ground type とよぶ.

例えば boolean expression を追加する場合は, 型と項を次のように拡張する.

$$\begin{array}{l} \alpha ::= \dots \mid \text{bool} \\ M ::= \dots \mid \text{true} \mid \text{false} \mid \text{if } M \text{ then } M \text{ else } M \end{array}$$

2.2 Product Type

product type はタプルあるいは組のことである.

定義 4 (product type extension). 型と項を次のように拡張する.

$$\begin{aligned}\alpha &::= \dots \mid \alpha_1 \times \alpha_2 \\ M &::= \dots \mid \text{proj}_i M \mid \langle M_1, M_2 \rangle\end{aligned}$$

それに付随して, いくつかの typing rule が追加される (図 2).

$$\begin{array}{c} \text{(pair)} \quad \frac{\Gamma \vdash M_1 : A_1 \quad \Gamma \vdash M_2 : A_2}{\Gamma \vdash \langle M_1, M_2 \rangle : A_1 \times A_2} \qquad \text{(proj)} \quad \frac{\Gamma \vdash M : A_1 \times A_2}{\Gamma \vdash \text{proj}_i M : A_i \quad (i = 1, 2)} \end{array}$$

図 2 product typing rules

2.3 Unit Type

unit type は 1 点集合のようなものである. product type を入れる時には入れることが多い.

定義 5 (unit type extension). 型と項を次のように拡張する.

$$\begin{aligned}\alpha &::= \dots \mid 1 \\ M &::= \dots \mid \star\end{aligned}$$

あるいは, unit type を **Unit**, 唯一の term を $()$ などとかくこともある.

$$\frac{}{\Gamma \vdash \star : 1}$$

図 3 unit typing rules

2.4 Abbreviations

- λ^{\rightarrow} の unit, product type extension を $\lambda^{\times,1}$ とかく.

第 II 部

Semantics

3 Equational Theory

ラムダ項同士の等式を定義しておく. 以下, calculus λ^{\rightarrow} や $\lambda^{\times,1}$ と言ったらこのような等式が使えるもの (judgement はこれらの等式で割った同値類で考えている) とする.

3.1 λ^{\rightarrow}

定義 6. $\beta\eta$ -equivalence から得られる λ^{\rightarrow} の equational theory を, λ^{\rightarrow} の標準的な equational theory $\mathcal{E}_{\lambda^{\rightarrow}}$ として採用する. 書き下すと図 4 のような規則によって equational theory が定まる. た

だしここでは、出現する項が well-typed であることなどは省略してかいた。

$$(\beta) \frac{}{\Gamma \vdash (\lambda x.M)N = M[N/x] : B} \quad (\eta) \frac{x \notin \text{FV}(M)}{\Gamma \vdash \lambda x.Mx = M : A \rightarrow B}$$

図4 equational theory of $\mathcal{E}_{\lambda \rightarrow}$

preliminaries でも言ったように, (refl), (sym), (trans), (subst) の4ルールはすでに含まれているものとして扱う。

3.2 $\lambda^{\times,1}$

定義 7. $\lambda^{\times,1}$ の標準的な equational theory $\mathcal{E}_{\lambda^{\times,1}}$ を, $\mathcal{E}_{\lambda \rightarrow}$ に加え次の rule からなるものとする。

$$(\text{prodU}) \frac{}{\Gamma \vdash \langle \text{proj}_1 M, \text{proj}_2 M \rangle = M : A \times B} \quad (\text{prodC}) \frac{}{\Gamma \vdash \text{proj}_i \langle M_1, M_2 \rangle = M_i : A_i}$$

$$(\text{unitU}) \frac{}{\Gamma \vdash M = \star : 1}$$

図5 equational theory of $\mathcal{E}_{\lambda^{\times,1}}$

4 Categorical Semantics

$\lambda^{\times,1}$ の categorical semantics を与えよう。

4.1 Syntactic category

定義 8 (syntactic category). 次のようにして定まる category $\text{Syn}(\lambda^{\times,1})$ を $\lambda^{\times,1}$ の **syntactic category** とよぶ。

- object: $\text{Typ}(\lambda^{\times,1})$
- $\text{hom}(A, B)$: judgement $x : A \vdash M : B$ (の equational theory に関する同値類)
- identity: $[x : A \vdash x : A]$ by (var)
- composition: $[y : B \vdash N : C] \circ [x : A \vdash M : B] := [x : A \vdash N[M/y] : C]$ by (app), (abs)

Proof. (associativity) 項 L, M, N に対し (M の context に使う変数 y は fresh にとってよい^{*1}),

$$\begin{aligned} ([z : C \vdash N : D] \circ [y : B \vdash M : C]) \circ [x : A \vdash L : B] &= [y : B \vdash N[M/z] : D] \circ [x : A \vdash L : B] \\ &= [x : A \vdash (N[M/z])[L/y] : D] \\ &= [x : A \vdash (N[L/y])[(M[L/y])/z] : D] \\ &= [x : A \vdash N[(M[L/y])/z] : D] \end{aligned}$$

$$\begin{aligned} [z : C \vdash N : D] \circ ([y : B \vdash M : C] \circ [x : A \vdash L : B]) &= [z : C \vdash N : D] \circ [x : A \vdash M[L/y] : C] \\ &= [x : A \vdash N[(M[L/y])/z] : D] \end{aligned}$$

となるので, associativity が成り立つ。

^{*1} judgement $x : A \vdash M : B$ により, M は x 以外の自由変数を含まない。よって y は出現したとしても束縛変数であるから, 適当に fresh な名前に取り替えておけばよい。ということを一々断る必要もないだろう。

(left identity) 項 M に対し,

$$\begin{aligned} [y : B \vdash y : B] \circ [x : A \vdash M : B] &= [x : A \vdash M[y/y] : B] \\ &= [x : A \vdash M : B] \end{aligned}$$

となるので, left identity が成り立つ.

right identity も同様. □

補題 9. $Syn(\lambda^{\times,1})$ は *cartesian closed*.

Proof. type constructor $1, \times, \rightarrow$ がそれぞれ terminal object, product, exponential を与えることをみる.

(terminal obj) 1 への射の存在と一意性がいえればよいが, 前者は $[\Gamma \vdash \star : 1]$ によって, 後者は (unitU) の規則によってわかる.

(product) product 自体は $\{x_1 : A_1, \dots, x_n : A_n\} \times \{y_1 : B_1, \dots, y_n : B_n\} = \text{projection}$ が $[x : A_1 \times A_2 \vdash \text{proj}_i(x) : A_i]$ ($i = 1, 2$) によって与えられる. また, $[x : P \vdash M_i : A_i]$ ($i = 1, 2$) が与えられたとき $[x : P \vdash \langle M_1, M_2 \rangle : A_1 \times A_2]$ が mediating arrow になり, 図式の可換性が (prodC) によってわかる. mediating arrow の一意性は (prodU) によりわかる.

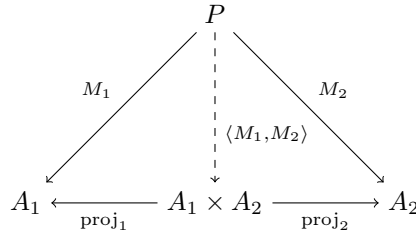


図 6 product の図式

(exponential) hom の同型は次のようにして与えられる:

$$\begin{aligned} \mathbb{S}(A \times B, C) &\cong \mathbb{S}(A, B \rightarrow C) \\ [x : A \times B \vdash M : C] &\mapsto [y : A \vdash \lambda b. M[\langle y, b \rangle / x] : C] \\ [x : A \times B \vdash (N[\text{proj}_1(x)/y])(\text{proj}_2(x)) : C] &\mapsto [y : A \vdash N : B \rightarrow C] \end{aligned}$$

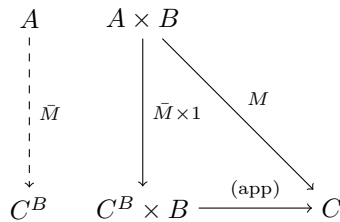


図 7 exponential の図式

上の同型*2は, 左から右が $M \mapsto \hat{M}$ になっており, 右から左が $\hat{M} \mapsto (\text{app}) \circ (\hat{M} \times 1)$ を得る操作になっている.

*2 厳密にはこれが同型であることはまだ示していない; 結果的には上の同型が exponential の同型を与えるが, 以下では図式の射の存在と一意性を示すことで exponential の存在を示している.

図式の可換性は,

$$\begin{aligned}
& [x : A \times B \vdash (\hat{M}[\text{proj}_1(x)/y])(\text{proj}_2(x)) : C] \\
&= [x : A \times B \vdash ((\lambda b. M[\langle y, b \rangle/x])[\text{proj}_1(x)/y])(\text{proj}_2(x)) : C] \\
&= [x : A \times B \vdash (\lambda b. M[\langle \text{proj}_1(x), b \rangle/x])(\text{proj}_2(x)) : C] \\
&= [x : A \times B \vdash M[\langle \text{proj}_1(x), \text{proj}_2(x) \rangle/x] : C] \\
&= [x : A \times B \vdash M[x/x] : C] \\
&= [x : A \times B \vdash M : C]
\end{aligned}$$

によってわかる. 一意性は, $[y : A \vdash N : B \rightarrow C]$ が $(\text{app}) \circ ((\lambda y. N) \times 1) = M$ を満たすとする. このとき, $y : A \vdash N = \lambda b. M[\langle y, b \rangle/x] : B \rightarrow C \in \mathcal{E}_{\lambda \times, 1}$ が言えればよいが, このことは η 変換により $y : A, b : B \vdash N b = M[\langle y, b \rangle/x] : C \in \mathcal{E}_{\lambda \times, 1}$ と同値である. そしてこれは,

$$\begin{aligned}
& [y : A, b : B \vdash M[\langle y, b \rangle/x] : C] \\
&= [y : A, b : B \vdash (N[\text{proj}_1(x)/y])(\text{proj}_2(x))[\langle y, b \rangle/x] : C] \quad (\text{app}) \circ ((\lambda y. N) \times 1) = M \text{ を代入} \\
&= [y : A, b : B \vdash (N[y/y])b : C] \\
&= [y : A, b : B \vdash N b : C]
\end{aligned}$$

によって示される. □

4.2 Categorical Model

ラムダ計算 $\lambda^{\times, 1}$ のモデルとは, category \mathbb{C} と interpretation $\llbracket - \rrbracket : \text{Syn}(\lambda^{\times, 1}) \rightarrow \mathbb{C}$ のことであった.

定義 10. \mathbb{C} を CCC とする. $\llbracket - \rrbracket_{\mathbb{C}_0} : \text{Syn}(\lambda^{\times, 1})_0 \rightarrow \mathbb{C}_0$ なる型から \mathbb{C} への対象への interpretation が与えられているとする. このとき, 次のようにして \mathbb{C} は $\lambda^{\times, 1}$ のモデルになる:

- judgement の interpretation $\llbracket x : A \vdash M : B \rrbracket_{\mathbb{C}_1}$ を定めればよい. context の interpretation を $\llbracket \{x_1 : A_1, \dots, x_n : A_n\} \rrbracket = A_1 \times \dots \times A_n$ としておく. まず次のようにして typing rule に従って $\llbracket \Gamma \vdash M : A \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket$ なる interpretation を定める.

$$\begin{aligned}
\llbracket \frac{x : A \in \Gamma}{\Gamma \vdash x : A} \rrbracket &= \llbracket \Gamma \rrbracket \xrightarrow{\text{pr}_x} \llbracket A \rrbracket \\
\llbracket \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x. M : A \rightarrow B} \rrbracket &= \frac{\llbracket \Gamma \rrbracket \times \llbracket A \rrbracket \xrightarrow{M} \llbracket B \rrbracket}{\llbracket \Gamma \rrbracket \xrightarrow{\hat{M}} \llbracket B \rrbracket^{\llbracket A \rrbracket}} \\
\llbracket \frac{\Gamma \vdash M_1 : A \rightarrow B \quad \Gamma \vdash M_2 : A}{\Gamma \vdash M_1 M_2 : B} \rrbracket &= \frac{\llbracket \Gamma \rrbracket \xrightarrow{M_1} B^A \quad \llbracket \Gamma \rrbracket \xrightarrow{M_2} A}{\llbracket \Gamma \rrbracket \xrightarrow{\langle M_2, M_1 \rangle} A \times B^A \xrightarrow{\text{ev}} B} \\
\llbracket \frac{\Gamma \vdash M_i : A_i}{\Gamma \vdash \langle M_1, M_2 \rangle : A_1 \times A_2} \rrbracket &= \frac{\llbracket \Gamma \rrbracket \xrightarrow{M_i} A_i}{\llbracket \Gamma \rrbracket \xrightarrow{\langle M_1, M_2 \rangle} A_1 \times A_2} \\
\llbracket \frac{\Gamma \vdash M : A_1 \times A_2}{\Gamma \vdash \text{proj}_i M : A_i} \rrbracket &= \frac{\llbracket \Gamma \rrbracket \xrightarrow{M} A_1 \times A_2}{\llbracket \Gamma \rrbracket \xrightarrow{M} A_1 \times A_2 \xrightarrow{\text{pr}_i} A_i} \\
\llbracket \frac{}{\Gamma \vdash \star : 1} \rrbracket &= \frac{}{\llbracket \Gamma \rrbracket \xrightarrow{!} 1}
\end{aligned}$$

- 次に, $\llbracket x : A \vdash M : B \rrbracket_{\mathbb{C}_1} = \llbracket \{x : A\} \vdash M : B \rrbracket$ によって, interpretation を定める. 上は \mathbb{C} の CCC の構造だけを使って定まっていることに注意.

4.3 Soundness

ここでは equational theory $\mathcal{E}_{\lambda \times, 1}$ の soundness をみよう.

定理 11 (sound). $\Gamma \vdash_{\lambda \times, 1} M = N : A$ ならば, 任意の $\lambda^{\times, 1}$ のモデル (上で与えた *interpretation*) $\llbracket - \rrbracket : \text{Syn}(\lambda^{\times, 1}) \rightarrow \mathbb{C}$ に対して $\llbracket M \rrbracket = \llbracket N \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket$ となる.

Proof. judgement に関する帰納法による.

(refl, sym, trans) 明らか.

(abs) $\Gamma \vdash \lambda x. M = \lambda x. N : A \rightarrow B$ とする. 帰納法の仮定より $\llbracket \Gamma, x : A \vdash M : B \rrbracket = \llbracket \Gamma, x : A \vdash N : B \rrbracket$ であるが, 両辺を curry 化すれば欲しい等式が得られる.

(app) $\Gamma \vdash M_1 M_2 = N_1 N_2 : A$ とする. 帰納法の仮定により, $\llbracket M_i \rrbracket = \llbracket N_i \rrbracket$ ($i = 1, 2$) となり OK.

(β) $\Gamma \vdash (\lambda x. M) N = M[N/x] : B$ とする. $\Gamma = \{x_1 : A_1 \times \cdots x_n : A_n\}$ とすると, interpretation の定義によって $\llbracket \Gamma \vdash M : A \rrbracket = \llbracket x : \Gamma \vdash M[\text{pr}_i(x)/x_i] : A \rrbracket$ が成り立つことに注意して,

$$\begin{aligned}
& \llbracket \Gamma \vdash (\lambda x. M) N : B \rrbracket \\
&= (\text{app}) \circ \langle \llbracket \Gamma \vdash \lambda x. M : A \rightarrow B \rrbracket, \llbracket \Gamma \vdash N : A \rrbracket \rangle \\
&= (\text{app}) \circ \langle \text{curry}(\llbracket \Gamma, x : A \vdash M : B \rrbracket), \llbracket \Gamma \vdash N : A \rrbracket \rangle \\
&= (\text{app}) \circ \text{curry}(\llbracket \Gamma, x : A \vdash M : B \rrbracket \times 1) \circ \langle 1, \llbracket \Gamma \vdash N : A \rrbracket \rangle \\
&= \text{uncurry}(\text{curry}(\llbracket \Gamma, x : A \vdash M : B \rrbracket)) \circ \langle 1, \llbracket \Gamma \vdash N : A \rrbracket \rangle \\
&= \llbracket \Gamma, x : A \vdash M : B \rrbracket \circ \langle 1, \llbracket \Gamma \vdash N : A \rrbracket \rangle \\
&= \llbracket z : \llbracket \Gamma \rrbracket \times A \vdash M[\text{pr}_i(\text{pr}_1(z))/x_i, \text{pr}_2(z)/x] : B \rrbracket \circ \llbracket y : \llbracket \Gamma \rrbracket \vdash \langle y, N[\text{pr}_i(y)/x_i] \rangle : \llbracket \Gamma \rrbracket \times A \rrbracket \\
&= \llbracket y : \llbracket \Gamma \rrbracket \vdash M[\text{pr}_i(\text{pr}_1(z))/x_i, \text{pr}_2(z)/x][\langle y, N[\text{pr}_i(y)/x_i] \rangle/z] : B \rrbracket \\
&= \llbracket y : \llbracket \Gamma \rrbracket \vdash M[\text{pr}_i(y)/x_i, N[\text{pr}_i(y)/x_i]/x] : B \rrbracket \\
&= \llbracket \Gamma \vdash M[x_i/x_i, N[x_i/x_i]/x] : B \rrbracket \\
&= \llbracket \Gamma \vdash M[N/x] : B \rrbracket
\end{aligned}$$

よって OK.

(η) $\Gamma \vdash \lambda x. (M x) = M : A \rightarrow B$ とする. このとき,

$$\begin{aligned}
& \llbracket \Gamma \vdash \lambda x. (M x) : A \rightarrow B \rrbracket \\
&= \text{curry}(\llbracket \Gamma, x : A \vdash M x : B \rrbracket) \\
&= \text{curry}((\text{app}) \circ \langle \llbracket \Gamma, x : A \vdash M : A \rightarrow B \rrbracket, \llbracket \Gamma, x : A \vdash x : A \rrbracket \rangle) \\
&= \text{curry}((\text{app}) \circ \langle \llbracket \Gamma, x : A \vdash M : A \rightarrow B \rrbracket, \text{pr}_2^{\llbracket \Gamma \rrbracket, A} \rangle) \\
&= \text{curry}((\text{app}) \circ (\llbracket \Gamma \vdash M : A \rightarrow B \rrbracket \times 1) \circ \langle \text{pr}_1^{\llbracket \Gamma \rrbracket, A}, \text{pr}_2^{\llbracket \Gamma \rrbracket, A} \rangle) \\
&= \text{curry}((\text{app}) \circ (\llbracket \Gamma \vdash M : A \rightarrow B \rrbracket \times 1)) \\
&= \text{curry}(\text{uncurry}(\llbracket \Gamma \vdash M : A \rightarrow B \rrbracket)) \\
&= \llbracket \Gamma \vdash M : A \rightarrow B \rrbracket
\end{aligned}$$

により OK.

(prodU), (prodC), (unitU) これは unit type が \mathbb{C} の 1 へ, product type が \mathbb{C} の product に対応していることから明らか.

□

4.4 Completeness

定理 12 (complete). 任意の $\lambda^{\times,1}$ のモデル $\llbracket - \rrbracket : \text{Syn}(\lambda^{\times,1}) \rightarrow \mathbb{C}$ に対して $\llbracket M \rrbracket = \llbracket N \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket$ が成り立つならば, $\Gamma \vdash_{\lambda^{\times,1}} M = N : A$ である.

Proof. $\text{Syn}(\lambda^{\times,1})$ が $\lambda^{\times,1}$ のモデルでもあることから明らか. \square

completeness は, syntactic category(term model) が “equational theory 以外に余分な構造を持たない” モデルであることが効いている.

4.5 Logical Formalism

categorical semantics の話をするにあたって, モデルなどの概念をもう少し形式的に定義しておこう. 本当に正しく議論をするには以下では不十分であるが, ここでは詳細には立ち入らないことにする.

4.5.1 Category of Theories

$\lambda^{\times,1}$ の signature とは, ラムダ計算としての構成要素を集めたもののことであった. 具体的に書き下すと以下ようになる.

定義 13. 次のものを $\lambda^{\times,1}$ の **signature** とよぶ.

- sort $\{p, m, v_p, v_m\}$: p は type, m は term, v_- は variable の意味である. それぞれ型, 項, 変数 (変数には項変数と型変数とがある) であることを表す.
- constructor symbols: これは以下のようなものからなる. また, variable を term と別の sort として考える必要はあまりないが, あとの証明の都合とわかりやすさを考慮の上でこのように定義している.
 - type variables: $\alpha, \beta, \gamma, \dots : v_p$
 - term variables: $x, y, z, \dots : v_m$
 - type constructor symbols: $\{\text{TVar} : v_p \rightarrow p, (\rightarrow) : p \rightarrow p \rightarrow p, 1 : p, (\times) : p \rightarrow p \rightarrow p\}$
 - term constructor symbols: $\{\text{Var} : v_m \rightarrow m, \text{Abs} : v_m \rightarrow m \rightarrow m, \text{App} : m \rightarrow m \rightarrow m, \star : m, \langle -, - \rangle : m \rightarrow m \rightarrow m, \text{proj}_i(-) : m \rightarrow m\}$
- typing rules 前述のもの: sort p と m を関係づける typing rule である. 各 term constructor が満たすべきルールともいえる.
- equational theory $\mathcal{E}_{\lambda^{\times,1}}$: すでに見たように, $\beta\eta$ -equality と product, 1 に関する equation を集めたものである.

同様にして, 一般のラムダ計算の signature も, このような sort, constructor symbols, typing rules, equational theory の組として定めることができる.

定義 14. category **Theory**($\lambda^{\times,1}$) を, 次のように定める.

- object: signature Σ であって, sort が $\{p, m, v_p, v_m\}$, type/term constructor symbols と typing rules と equational theory は $\Sigma(\lambda^{\times,1})$ を含むようなもの^{*3}

^{*3} sort はこれらを含む一般の形と弱めることもできるはずだが, 複数の sort を含むような constructor symbol の扱いをどうするかという問題がある. 後に signature に沿ってモデルを定義する必要がある関係で, ここではこのような定義を採用した.

- hom : signature の各要素ごとの包含

これは, product type と unit type をもつ calculus とその間の拡大のなす category(poset) だと思つてよい. このとき, $\Sigma(\lambda^{\times,1})$ は initial object である.

4.5.2 Category of Models

定義 15. category $\mathbf{Model}(\lambda^{\times,1})$ を, 次のように定める.

- object: $\lambda^{\times,1}$ のモデル $\llbracket - \rrbracket_{\mathbb{C}} : \text{Syn}(\lambda^{\times,1}) \rightarrow \mathbb{C}$
- $\text{hom}(\mathbb{C}, \mathbb{D})$: category の constructor とその構造^{*4}を保つ functor $F : \mathbb{C} \rightarrow \mathbb{D}$ であつて, interpretation と可換なもの: $F \circ \llbracket - \rrbracket_{\mathbb{C}} = \llbracket - \rrbracket_{\mathbb{D}}$.

このとき, $\text{Syn}(\lambda^{\times,1})$ も当然この category の object であり, さらに任意のモデルへの morphism が存在するので, $\text{Syn}(\lambda^{\times,1})$ は $\mathbf{Model}(\lambda^{\times,1})$ の initial object である.

4.6 Categorical Semantics

定義 16. functor $\text{Syn} : \mathbf{Theory}(\lambda^{\times,1}) \rightarrow \mathbf{Model}(\lambda^{\times,1})$ が次のようにして定まる.

- $\text{Syn}(\Sigma)$ は, Σ から作ったラムダ計算の syntactic category で, 次のようにして定まる:
 - object: type constructor を組み合わせて得られる文字列
 - $\text{hom}(A, B)$: term constructor を組み合わせて得られる文字列 M で, typing rule $x : A \vdash M : B$ を満たすようなもの.
- $\text{Syn}(\Sigma \subseteq \Sigma')$ は, inclusion functor.

$\text{Syn}(\Sigma)$ が CCC になることなどは補題 9 と同様にして示せる.

また, $\text{Syn}(\Sigma(\lambda^{\times,1}))$ が $\text{Syn}(\lambda^{\times,1})$ に一致することも明らか.

定理 17. $\mathbf{Theory}(\lambda^{\times,1})$ と $\mathbf{Model}(\lambda^{\times,1})$ は categorical semantics を与える.

Proof. Syn の右随伴が存在することをみる. さて, 任意のモデル \mathbb{C} と interpretation $\llbracket - \rrbracket : \text{Syn}(\lambda^{\times,1}) \rightarrow \mathbb{C}$ に対して $\text{Lan}(\mathbb{C})$ を,

- sort: $\lambda^{\times,1}$ と同じ
- constructor symbol: \mathbb{C} の object/morphism の constructor^{*5}.
- typing rules: morphism の constructor に関するルールを記述する.
- equational theory: \mathbb{C} の morphism とみて等しい等式を全て集めてきて, それを $\text{Lan}(\mathbb{C})$ のコンストラクタを組み合わせて表現したもの.

と定める. これが Syn の右随伴であることを示せばよい. つまり,

$$\mathbf{Model}(\text{Syn}(\Sigma), \mathbb{C}) \cong \mathbf{Theory}(\Sigma, \text{Lan}(\mathbb{C}))$$

なる natural iso があることを示す. ところで, \mathbf{Theory} は poset なのでこれは次と同値.

$$\mathbf{Model}(\text{Syn}(\Sigma), \mathbb{C}) \iff \Sigma \subseteq \Sigma(\text{Lan}(\mathbb{C}))$$

^{*4} とりあえず CCC の構造を保つ, と思つておいてよい.

^{*5} ここでは \mathbb{C} の constructor と書いたが, 厳密にやるのであれば cartesian closed category という概念自体を, constructor とそれらのルールにわけて形式化するというをしなければならない. つまり, “ $\lambda^{\times,1}$ の Model の category とは, \sim を満たす signature からなる category であり, さらにそれらが CCC の axiom を満たしているようなもの” のような書き方で形式化して定義を行う必要がある. しかしそれはあまりに煩雑なのでここでは曖昧な書き方をすることにした.

(\implies) $\llbracket - \rrbracket : \text{Syn}(\lambda^{\times,1}) \rightarrow \mathbb{C}$ を任意のモデルとし, $F : \text{Syn}(\Sigma) \rightarrow \mathbb{C}$ をモデルの morphism とする. $\Sigma \subseteq \Sigma(\text{Lan}(\mathbb{C}))$ をみればよいが, sort は共通しており, $F : \text{Syn}(\Sigma) \rightarrow \mathbb{C}$ がモデルの morphism なので constructor と equational theory が保たれることもわかる. よって OK.

(\impliedby) $\Sigma \subseteq \Sigma(\text{Lan}(\mathbb{C}))$ のとき, \mathbb{C} がモデルになることをみる. これは $\llbracket - \rrbracket : \text{Syn}(\Sigma) \rightarrow \mathbb{C}$ を, type, constructor をそのまま移すようなものとする. このとき typing rule の包含から $\text{Syn}(\Sigma)$ の constructor には \mathbb{C} でも同じように型がつくこと, そして equational theory の包含から $\text{Syn}(\Sigma)$ の等しい射は \mathbb{C} でも等しいこと (つまり mapping part に関して well-defined なこと) がわかる. これにより $\llbracket - \rrbracket$ はモデルの morphism となるので, $\text{Syn}(\lambda^{\times,1})$ の initiality によりモデルの interpretation $\llbracket - \rrbracket : \text{Syn}(\lambda^{\times,1}) \rightarrow \mathbb{C}$ が得られる. \square

系 18. CCC の internal language は $\lambda^{\times,1}$ であり, $\lambda^{\times,1}$ の syntactic category は CCC である.