simply-typed λ-calculus

myuon

2018 年 9 月 1 日

目次

第 I 部　Syntax 2

1 Types 2

2 Properties 2

2.1 Strong Normalization 2

3 Simple Extension 4

3.1 Base Type 4

3.2 Product Type 4

3.3 Unit Type 4

3.4 Abbreviations 5

第 II 部　Semantics 5

4 Equational Theory 5

4.1 $\lambda^{\to}$ 5

4.2 $\lambda^{\times,1}$ 5

5 Categorical Semantics 6

5.1 Syntactic category 6

5.2 Categorical Model 8

5.3 Soundness 8

5.4 Completeness 9

5.5 Logical Formalism 10

5.6 Categorical Semantics 11

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, これには通常の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 ($\lambda^{\to}$) 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 $\mathrm{Typ}(\lambda^{\to})$ などとかく.

$$\alpha ::= b \mid \alpha_1 \to \alpha_2$$

b は base type とよばれ, 言語に組み込みの値を表現するための型である. base type の集合 B と各 $b \in B$ に対応するコンストラクター c_b が与えられているとする.

$\alpha_1 \to \alpha_2$ 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 $A \to B$ などと書いたりするのとちょうど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 $- \vdash - : - \subseteq (V_t \times \mathrm{Typ}) \times \Lambda \times \mathrm{Typ}$ を以下のように定義する.

$$\text{(var)}\ \frac{x : A \in \Gamma}{\Gamma \vdash x : A} \qquad \text{(base)}\ \frac{}{\Gamma \vdash c_b : b}$$

$$\text{(abs)}\ \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B} \qquad \text{(app)}\ \frac{\Gamma \vdash M_1 : A \to B \qquad \Gamma \vdash M_2 : A}{\Gamma \vdash M_1 M_2 : B}$$

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 $x : A \in \Gamma$.

2. $\Gamma \vdash c_b : A$ ならば $A = b$.

3. $\Gamma \vdash \lambda x.M : A$ ならば, A_1, A_2 が存在して, $A = A_1 \to A_2$ かつ $\Gamma, x : A_1 \vdash M : A_2$.

4. $\Gamma \vdash M_1 M_2 : B$ ならば, A が存在して, $\Gamma \vdash M_1 : A \to B$ かつ $\Gamma \vdash M_2 : A$.

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

2 Properties

2.1 Strong Normalization

次の定理を示すのが目標である.

定理 (Strong Normalization)**.** *$\lambda^{\to}$ は SN である. すなわち, $\lambda^{\to}$ の well-typed な項は strongly normalizing(任意の簡約列が停止する).*

これを示すために, 次のように型に対し特定の条件を満たす項の集合を関係づける. このように型と項を関係づけて示す手法を logical relation などとよぶ.

定義 4. 次のように reducibility candidates RED を定義する.

- $t \in \mathrm{RED}_b$ iff t は strongly normalizing.
- $t \in \mathrm{RED}_{A \to B}$ iff $\forall u \in \mathrm{RED}_A, (t\, u) \in \mathrm{RED}_B$.

定義 5. 次のように RED を context へ拡張する.

$$\mathrm{RED}_\Gamma = \{\gamma; \mathrm{dom}\, \Gamma = \mathrm{dom}\, \gamma, \forall x : A \in \Gamma, x[\gamma] \in \mathrm{RED}_A\}$$

補題 6. *$M \in RED_A$ かつ $M \to_\beta M'$ ならば $M' \in RED_A$.*

証明. A についての帰納法.
(base) $A = b$ のとき. M は strongly normalizing なので M' もそう. よって OK.
(fun) $A = A_1 \to A_2$ のとき. $M' \in \mathrm{RED}_A$ を示すので, 任意の $N \in \mathrm{RED}_{A_1}$ に対して $M'N \in \mathrm{RED}_{A_2}$ であることを示す. $M \in \mathrm{RED}_{A_1 \to A_2}$ なので, $MN \in \mathrm{RED}_{A_2}$ である. $MN \to_\beta M'N$ であることと帰納法の仮定により $M'N \in \mathrm{RED}_{A_2}$. □

補題 7. *$M \in RED_A$ ならば, M は s.n.*

補題 8. *M が neutral(関数抽象でない項) であり, $(\forall M'.\ M \to M' \implies M' \in RED_A)$ ならば $M \in RED_A$.*

証明. この 2 つの補題は同時に A についての帰納法で示す.
(base) $A = b$ のとき. 前者は明らか. 後者は, $M \to M'$ となる任意の M' が s.n. ならば M も s.n.
(fun) $A = A_1 \to A_2$ のとき.

- $M \in \mathrm{RED}_{A_1 \to A_2}$ より, 任意の $N \in \mathrm{RED}_{A_1}$ に対して $MN \in \mathrm{RED}_{A_2}$ である. ところで $c_b \in \mathrm{RED}_b$ なので $Mc_b \in \mathrm{RED}_{A_2}$. 帰納法の仮定より, Mc_b は s.n. であり, よって M は s.n. である.
- 任意の $N \in \mathrm{RED}_{A_1}$ に対し, $MN \in \mathrm{RED}_{A_2}$ を示す. 帰納法の仮定により, N は s.n. である. 以下の補題を示せばよい.

  補題. *$N \in RED_{A_1}$ かつ N は s.n. ならば, $MN \in RED_{A_2}$.*

  証明. N が s.n. であることより, 最長簡約列の長さについての帰納法で示す.
  (N がすでに nf のとき) $MN \to L$ となる L を考えると, M が neutral なことより L は $M'N$ の形で, $M \to M'$ である. ここで元の補題の仮定より, $M' \in \mathrm{RED}_{A_1 \to A_2}$ なので $M'N = L \in \mathrm{RED}_{A_2}$ であることがわかる. よって元の帰納法の仮定より, $MN \in \mathrm{RED}_{A_2}$.
  (N の最大列の長さが $n+1$ のとき) $MN \to L$ となる L を任意にとると, $L = M'N$ か MN' のいずれか. 前者は上と同様の議論で $M'N \in \mathrm{RED}_{A_2}$. $N \to N'$ のとき, N' は s.n. であって, 6 により $N' \in \mathrm{RED}_{A_1}$ でもある. よってこの補題の帰納法の仮定により, $MN' \in \mathrm{RED}_{A_2}$. ゆえに元の帰納法の仮定より, $MN \in \mathrm{RED}_{A_2}$. □

□

補題 9. *$N \in RED_{A'}$, $M[N/x] \in RED_A$ ならば, $(\lambda x.M)N \in RED_A$.*

証明. 補題 7 により, N と $M[N/x]$ は s.n. である. よって M も s.n. である. このことより, $(\lambda x.M)N$ も s.n. である. $(\lambda x.M)N$ の最長簡約列の長さに関する帰納法で示す.
(最大列の長さが 0 のとき) $(\lambda x.M)N$ は nf ではないのでこれはありえない.
(最大列の長さが $n+1$ のとき) $(\lambda x.M)N \to L$ となる L は, $(\lambda x.M')N$, $(\lambda x.M)N'$, $M[N/x]$ のい

ずれかである．前 2 つが RED_A に属することは帰納法の仮定によりわかる．最後のケースが RED_A に属することは元の仮定によりわかる．よって補題 8 により, $(\lambda x.M)N \in \mathrm{RED}_A$ である． $\square$

定理 10 (Fundamental Theorem)**.** *$\Gamma \vdash M : A$ かつ $\gamma \in RED_\Gamma$ なら, $M[\gamma] \in RED_A$.*

証明． M についての帰納法で示す．

(var) $M = x$ のとき． $x : A \in \Gamma$ と $\gamma \in \mathrm{RED}_\Gamma$ であることより $x[\gamma] \in \mathrm{RED}_A$ がわかる．

(base) $M = c_b$ のとき． c_b は β 基を含まないから strongly normalizing. よって OK.

(abs) $M = \lambda x.M'$ のとき (ただし x は束縛変数なので γ に fresh になるように選んでよい). $\Gamma \vdash \lambda x.M' : A_1 \to A_2$ すなわち $\Gamma, x : A_1 \vdash M' : A_2$ である．このとき, $M[\gamma] \in \mathrm{RED}_{A_1 \to A_2}$ であること, すなわち任意の $N \in \mathrm{RED}_{A_1}$ に対して $M[\gamma]\, N = (\lambda x.M'[\gamma])N \in \mathrm{RED}_{A_2}$ であることを示せばよい．ところで, $x[\gamma, N/x] = N$ であるから $\gamma[N/x] \in \mathrm{RED}_{\Gamma, x:A_1}$ を満たす．よって $\gamma[N/x] \in \mathrm{RED}_{\Gamma, x:A_1}$ となって帰納法の仮定より, $M'[\gamma[N/x]] = M'[\gamma][N/x] \in \mathrm{RED}_{A_2}$ である．ゆえに補題 9 より $(\lambda x.M'[\gamma])N \in \mathrm{RED}_{A_2}$.

(app) $M = M_1 M_2$ のとき． $\Gamma \vdash M_1 : A \to B$ かつ $\Gamma \vdash M_2 : A$ とする．帰納法の仮定より, $M_1[\gamma] \in \mathrm{RED}_{A_1 \to A_2}$ かつ $M_2[\gamma] \in \mathrm{RED}_{A_1}$ である．RED の定義により, $M[\gamma] = M_1[\gamma] M_2[\gamma] \in \mathrm{RED}_{A_2}$ なので OK. $\square$

定理 11 (Strong Normalization)**.** *$\Gamma \vdash M : A$ ならば, M は s.n.*

証明． 補題 8 により, $x \in \mathrm{RED}_A$ である．よって identity substitution について $id \in \mathrm{RED}_\Gamma$ となる．これと定理 10 により, $M \in \mathrm{RED}_A$ となり, 補題 7 により s.n. である． $\square$

3 Simple Extension

3.1 Base Type

上ではラムダ計算に base type を追加したが, 定義自体最低限しかしていないので, 具体的な型を追加する時にどのようにするかを例として挙げておく．

base type を $b \in B$ で, その時の constructor を C_b でかくことにする．

- boolean: $B = \{\mathrm{bool}\}, C_{\mathrm{bool}} = \{\mathrm{true}, \mathrm{false}\}$
- nat: $B = \{\mathrm{nat}\}, C_{\mathrm{nat}} = \{\mathrm{zero}, \mathrm{succ}(C_{\mathrm{nat}})\}$

3.2 Product Type

product type はタプルあるいは組のことである．

定義 12 (product type extension)**.** 型と項を次のように拡張する．

$$\alpha ::= \ldots \mid \alpha_1 \times \alpha_2$$
$$M ::= \ldots \mid \mathrm{proj}_i M \mid \langle M_1, M_2 \rangle$$

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

3.3 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．

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

図 2　product typing rules

定義 13 (unit type extension)**.** 型と項を次のように拡張する.

$$\alpha ::= \ldots \mid 1$$
$$M ::= \ldots \mid \star$$

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

3.4　Abbreviations

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

第 II 部

Semantics

4　Equational Theory

ラムダ項同士の等式を定義しておく. 以下, calculus $\lambda^{\to}$ 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.

4.1　$\lambda^{\to}$

定義 14. $\beta\eta$-equivalence から得られる $\lambda^{\to}$ の equational theory を, $\lambda^{\to}$ の標準的な equational theory$\mathcal{E}_{\lambda^{\to}}$ 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.

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

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

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.

4.2　$\lambda^{\times,1}$

定義 15. $\lambda^{\times,1}$ の標準的な equational theory$\mathcal{E}_{\lambda^{\times,1}}$ を, $\mathcal{E}_{\lambda^{\to}}$ に加え次の rule からなるものとする.

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

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

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

5　Categorical Semantics

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

5.1　Syntactic category

定義 16 (syntactic category)**.** 次のようにして定まる category $\mathrm{Syn}(\lambda^{\times,1})$ を $\lambda^{\times,1}$ の **syntactic category** とよぶ.

- object: $\mathrm{Typ}(\lambda^{\times,1})$
- 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)

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

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

[*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.

(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 $\{x_1 : A_1, \ldots, x_n : A_n\} \times \{y_1 : B_1, \ldots, y_n : B_n\}$ =projection が $[x : A_1 \times A_2 \vdash \mathrm{proj}_i(x) : A_i]$ $(i = 1, 2)$ 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.

$$
\begin{tikzcd}
 & P \arrow[dl, "M_1"'] \arrow[d, dashed, "\langle M_1, M_2 \rangle"] \arrow[dr, "M_2"] & \\
A_1 & A_1 \times A_2 \arrow[l, "\mathrm{proj}_1"] \arrow[r, "\mathrm{proj}_2"'] & A_2
\end{tikzcd}
$$

図 6　product の図式

(exponential) hom の同型は次のようにして与えられる:

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

$$
\begin{tikzcd}
A \arrow[d, dashed, "\bar{M}"] & A \times B \arrow[d, "\bar{M} \times 1"] \arrow[dr, "M"] & \\
C^B & C^B \times B \arrow[r, "(\mathrm{app})"] & C
\end{tikzcd}
$$

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 $\hat{M} \mapsto (\mathrm{app}) \circ (\hat{M} \times 1)$ を得る操作になっている.

図式の可換性は,

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

によってわかる. 一意性は, $[y : A \vdash N : B \to C]$ が $(\mathrm{app}) \circ ((\lambda y. N) \times 1) = M$ を満たすとする. このとき, $y : A \vdash N = \lambda b.\, M[\langle y, b \rangle / x] : B \to C \in \mathcal{E}_{\lambda^{\times,1}}$ 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,

*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.

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

によって示される. □

5.2 Categorical Model

ラムダ計算 $\lambda^{\times,1}$ のモデルとは, category $\mathbb{C}$ と interpretation $[\![-]\!]:\mathrm{Syn}(\lambda^{\times,1})\to\mathbb{C}$ のことであった.

定義 18. $\mathbb{C}$ を CCC とする. $[\![-]\!]_{\mathbb{C}_0}:\mathrm{Syn}(\lambda^{\times,1})_0\to\mathbb{C}_0$ なる型から $\mathbb{C}$ への対象への interpretation が与えられており, $\mathbb{C}$ は $\lambda^{\times,1}$ の base type constructor c_b に対応する射 $c_b:1\to[\![b]\!]$ があるとする. 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$[\![x:A\vdash M:B]\!]_{\mathbb{C}_1}$ を定めればよい. context の interpretation を $[\![\{x_1:A_1,\ldots,x_n:A_n\}]\!]=A_1\times\cdots\times A_n$ としておく. まず次のようにして typing rule に従って $[\![\Gamma\vdash M:A]\!]:[\![\Gamma]\!]\to[\![A]\!]$ なる interpretation を定める.

$$
\left[\!\!\left[\frac{x:A\in\Gamma}{\Gamma\vdash x:A}\right]\!\!\right]=[\![\Gamma]\!]\xrightarrow{\mathrm{pr}_x}[\![A]\!]
$$

$$
\left[\!\!\left[\frac{}{\Gamma\vdash c_b:b}\right]\!\!\right]=[\![\Gamma]\!]\xrightarrow{!}1\xrightarrow{c_b}[\![b]\!]
$$

$$
\left[\!\!\left[\frac{\Gamma,x:A\vdash M:B}{\Gamma\vdash\lambda x.M:A\to B}\right]\!\!\right]=\frac{[\![\Gamma]\!]\times[\![A]\!]\xrightarrow{M}[\![B]\!]}{[\![\Gamma]\!]\xrightarrow[\hat{M}]{}[\![B]\!]^{[\![A]\!]}}
$$

$$
\left[\!\!\left[\frac{\Gamma\vdash M_1:A\to B\qquad\Gamma\vdash M_2:A}{\Gamma\vdash M_1M_2:B}\right]\!\!\right]=\frac{[\![\Gamma]\!]\xrightarrow{M_1}B^A\qquad[\![\Gamma]\!]\xrightarrow{M_2}A}{[\![\Gamma]\!]\xrightarrow{\langle M_2,M_1\rangle}A\times B^A\xrightarrow{\mathrm{ev}}B}
$$

$$
\left[\!\!\left[\frac{\Gamma\vdash M_i:A_i}{\Gamma\vdash\langle M_1,M_2\rangle:A_1\times A_2}\right]\!\!\right]=\frac{[\![\Gamma]\!]\xrightarrow{M_i}A_i}{[\![\Gamma]\!]\xrightarrow{\langle M_1,M_2\rangle}A_1\times A_2}
$$

$$
\left[\!\!\left[\frac{\Gamma\vdash M:A_1\times A_2}{\Gamma\vdash\mathrm{proj}_iM:A_i}\right]\!\!\right]=\frac{[\![\Gamma]\!]\xrightarrow{M}A_1\times A_2}{[\![\Gamma]\!]\xrightarrow{M}A_1\times A_2\xrightarrow{\mathrm{pr}_i}A_i}
$$

$$
\left[\!\!\left[\frac{}{\Gamma\vdash\star:1}\right]\!\!\right]=[\![\Gamma]\!]\xrightarrow{!}1
$$

- 次に, $[\![x:A\vdash M:B]\!]_{\mathbb{C}_1}=[\![\{x:A\}\vdash M:B]\!]$ 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.

5.3 Soundness

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

定理 19 (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)* $[\![-]\!]:\mathit{Syn}(\lambda^{\times,1})\to\mathbb{C}$ *に対して* $[\![M]\!]=[\![N]\!]:[\![\Gamma]\!]\to[\![A]\!]$ *となる.*

証明. judgement に関する帰納法による.

(refl, sym, trans) 明らか.

(abs) $\Gamma \vdash \lambda x.M = \lambda x.N : A \to B$ とする. 帰納法の仮定より $[\![\Gamma, x : A \vdash M : B]\!] = [\![\Gamma, x : A \vdash N : B]\!]$ 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, $[\![M_i]\!] = [\![N_i]\!]$ $(i = 1, 2)$ 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 $[\![\Gamma \vdash M : A]\!] = [\![x : \Gamma \vdash M[\mathrm{pr}_i(x)/x_i] : A]\!]$ が成り立つことに注意して,

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

よって OK.

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

$$\begin{aligned}
&[\![\Gamma \vdash \lambda x.\,(M\ x) : A \to B]\!] \\
&= \mathrm{curry}([\![\Gamma, x : A \vdash M\ x : B]\!]) \\
&= \mathrm{curry}((\mathrm{app}) \circ \langle [\![\Gamma, x : A \vdash M : A \to B]\!], [\![\Gamma, x : A \vdash x : A]\!] \rangle) \\
&= \mathrm{curry}((\mathrm{app}) \circ \langle [\![\Gamma, x : A \vdash M : A \to B]\!], \mathrm{pr}_2^{[\![\Gamma]\!], A} \rangle) \\
&= \mathrm{curry}((\mathrm{app}) \circ ([\![\Gamma \vdash M : A \to B]\!] \times 1) \circ \langle \mathrm{pr}_1^{[\![\Gamma]\!], A}, \mathrm{pr}_2^{[\![\Gamma]\!], A} \rangle) \\
&= \mathrm{curry}((\mathrm{app}) \circ ([\![\Gamma \vdash M : A \to B]\!] \times 1)) \\
&= \mathrm{curry}(\mathrm{uncurry}([\![\Gamma \vdash M : A \to B]\!])) \\
&= [\![\Gamma \vdash M : A \to B]\!]
\end{aligned}$$

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.

□

5.4 Completeness

定理 20 (complete)**.** 任意の $\lambda^{\times,1}$ のモデル $[\![-]\!] : \mathit{Syn}(\lambda^{\times,1}) \to \mathbb{C}$ に対して $[\![M]\!] = [\![N]\!] : [\![\Gamma]\!] \to [\![A]\!]$ が成り立つならば, $\Gamma \vdash_{\lambda^{\times,1}} M = N : A$ である.

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

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.

5.5 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.

5.5.1 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のようになる.

定義 21. 次のものを $\lambda^{\times,1}$ の **signature** とよぶ.

- sort $\{p, m, v\}$: p は type, m は term, v は variable の意味である. それぞれ型, 項, 変数であることを表す.
- 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.
  - term variables: $x, y, z, \cdots : v_m$
  - type constructor symbols: $\{B : p, \mathrm{TVar} : v_p \to p, (\to) : p \to p \to p, 1 : p, (\times) : p \to p \to p\}$
  - term constructor symbols: $\{c_b : m, \mathrm{Var} : v_m \to m, \mathrm{Abs} : v_m \to m \to m, \mathrm{App} : m \to m \to m, \star : m, \langle -, - \rangle : m \to m \to m, \mathrm{proj}_i(-) : m \to m\}$
- 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 22. category $\mathbf{Theory}(\lambda^{\times,1})$ を, 次のように定める.

- object: signature Σ であって, sort が $\{p, m, v\}$, type/term constructor symbols と typing rules と equational theory は $\Sigma(\lambda^{\times,1})$ を含むようなもの[*3]
- 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) だと思ってよい. このとき, $\Sigma(\lambda^{\times,1})$ は initial object である.

5.5.2 Category of Models

定義 23. category $\mathbf{Model}(\lambda^{\times,1})$ を, 次のように定める.

- object: $\lambda^{\times,1}$ のモデル $[\![-]\!]_{\mathbb{C}} : \mathrm{Syn}(\lambda^{\times,1}) \to \mathbb{C}$

[*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($\mathbb{C}$,$\mathbb{D}$): category の constructor とその構造[*4]を保つ functor $F : \mathbb{C} \to \mathbb{D}$ であって, interpretation と可換なもの: $F \circ [\![-]\!]_{\mathbb{C}} = [\![-]\!]_{\mathbb{D}}$.

このとき, $\mathrm{Syn}(\lambda^{\times,1})$ 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, $\mathrm{Syn}(\lambda^{\times,1})$ は $\mathbf{Model}(\lambda^{\times,1})$ の initial object である.

5.6 Categorical Semantics

定義 24. functor $\mathrm{Syn} : \mathbf{Theory}(\lambda^{\times,1}) \to \mathbf{Model}(\lambda^{\times,1})$ が次のようにして定まる.

- $\mathrm{Syn}(\Sigma)$ 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
  - 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.
- $\mathrm{Syn}(\Sigma \subseteq \Sigma')$ は, inclusion functor.

$\mathrm{Syn}(\Sigma)$ が CCC になることなどは補題 17 と同様にして示せる.

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

定理 25. $\mathbf{Theory}(\lambda^{\times,1})$ と $\mathbf{Model}(\lambda^{\times,1})$ は *categorical semantics* を与える.

証明. 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 $[\![-]\!] : \mathrm{Syn}(\lambda^{\times,1}) \to \mathbb{C}$ に対して $\mathrm{Lan}(\mathbb{C})$ 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 $\mathrm{Lan}(\mathbb{C})$ 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,

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

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

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

($\Longrightarrow$) $[\![-]\!] : \mathrm{Syn}(\lambda^{\times,1}) \to \mathbb{C}$ を任意のモデルとし, $F : \mathrm{Syn}(\Sigma) \to \mathbb{C}$ をモデルの morphism とする. $\Sigma \subseteq \Sigma(\mathrm{Lan}(\mathbb{C}))$ をみればよいが, sort は共通しており, $F : \mathrm{Syn}(\Sigma) \to \mathbb{C}$ 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.

($\Longleftarrow$) $\Sigma \subseteq \Sigma(\mathrm{Lan}(\mathbb{C}))$ のとき, $\mathbb{C}$ がモデルになることをみる. これは $[\![-]\!] : \mathrm{Syn}(\Sigma) \to \mathbb{C}$ 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 $\mathrm{Syn}(\Sigma)$ 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 $\mathrm{Syn}(\Sigma)$ の等

[*4] とりあえず CCC の構造を保つ, と思っておいてよい.

[*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ことをしなければならない. つまり, "$\lambda^{\times,1}$ の Model の category とは, ∼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.

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 $[\![-]\!]$ はモデルの morphism となるので, $\mathrm{Syn}(\lambda^{\times,1})$ の initiality によりモデルの interpretation $[\![-]\!] : \mathrm{Syn}(\lambda^{\times,1}) \to \mathbb{C}$ が得られる. $\square$

系 26. *CCC の internal language は* $\lambda^{\times,1}$ *であり,* $\lambda^{\times,1}$ *の syntactic category は CCC である.*