ソフトウェア基礎論 31 − 8
（1989. 9. 20）

ML のモデルとしての internal ccc

佐藤周行

東京大学理学部情報科学科
東京都文京区本郷 7-3-1.

概要

プログラミング言語 ML はポリモルフィズムを実現している言語としてその重要性が認識されてきている。本稿では ML がポリモルフィズムの中でも単純な型体系を持ち、ccc と緊密な関係を持っていることを示す。つまり、ML はポリモルフィズムと typed λ-calculus との中間に位置するのである。そのために internal ccc を導入し、その基本的な性質を示す。internal ccc の等号による表現を与える。また、通常の typed λ-calculus と ccc とに対の対応が成立するのと同様の方針で、ML と internal ccc には、対の対応のあることを証明する。

An ML-theory is an INTERNAL CCC.

Sato Hiroyuki

Department of Information Science,
University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113.

Abstract

Today, the programming language ML increases its significance in computer science, as a language implementing polymorphism. In this article, we show that ML is a "simple" polymorphism and is closely related to the theory of cartesian closed category(ccc) which is familiar to every computer scientist. Internal ccc thory is defined. Its basic properties are proved. Its equational presentation is given. We show that an ML-theory and an internal ccc theory are translated to each other by $\mathcal{C}$ and $\mathcal{L}$. Using these translations, they are proved to represent the same concept. In other words, an ML-theory is a predicative polymorphism over a cartesian closed categorical structure.

1 Introduction

The programming language ML [15] is one of the most successful system implementing *polymorphism.* Although, there are a number of other stronger polymorphic systems than ML such as Girard's $\mathcal{F}$[5], or equivalently Reynold's system [12]. and Coquand's theory of constructions [2], their implementations are, if exist, less popular than ML. Therefore, in computer science, it is as important to study properties specific to ML, as to study general properties of polymorphism.

Today, in computer science, the categorical approach is increasing both in popularity and significance. We are here interested in approaches in theoretical computer science. Theoretically, the categorical approach has a number of results in the polymorphism. One approach is to develop category theory using polymorphsim. For example, Coquand suggested in [3] the possibility to interpret category theory polymorphically. (Hereafter, we call it "Coquand's approach.") Another approach is to interpret polymorphism in categoryical terms. Seely gives in [14], a model of $\mathcal{F}$ as *PL–category* which utilize the indexed category theory. Ehrhard gives the categorical semantics of the Coquand's construction in [4]. He developos it using the theory of fibration. However, these approaches are too general to serve as a model of ML. They have no counterpart to the simpleness mentioned in [3].

In this article, we distinguish ML-style-polymorphism from $\mathcal{F}$. We consider, as its model, the *internal* cartesian closed category theory. Internal products, internal exponentials, internal terminal are defined. The equational presentation of the internal ccc theory is also given. Equational presentation is important in the sense that if we can give it, the system can be handled *algebraically.* This presentation indicates that the internal ccc theory is essentially the internalization of the usual (external) ccc theory. From an internal ccc, we construct an ML-theory via the translation $\mathcal{L}$. Conversely, from an ML-theory, we construct an internal ccc using the translation $\mathcal{C}$. The construction of these transltations are proved to imply that ML is, in its essense, the language expressing the ccc theory — internally. In particular, The translation $\mathcal{L}$ is the precision of the Coquand's approach. It is natural to conclude that ML is the polymorphism enough to develop the ccc theory. The translations reveal this point in a simple and natural way. In fact, ML is a *predicative* subsystem of $\mathcal{F}$. The popularity of ML owes partially to this simpleness.

2 ML-style-polymorphism

We consider, in this article, the polymorphic system which is the basis of the type system of ML. It is a subsystem of $\mathcal{F}$[5] and presented as:

definition 2.1 An ML-theory consists of the following data.

TYPES:

1. A set of primitive types is given.
2. () is a type.
3. A denumerable set of type variables is given.
4. If A and B are types, then $A \times B$ is a type.
5. If A and B are types, then $A \rightarrow B$ is a type.

We use a lower case greek letter ($\alpha, \beta, \cdots$) for a type variable. We assume that type variables of each type are linearly ordered. A problem arises: in the construction (4) and (5), we must fix the order type variables appearing in the constructed type. However, we do not discuss this problem in detail and simple say that it is (can be, indeed) appropriately taken.

TERMS:

We use the notation *term* : *type* to denote both the fact that the term *term* has type *type* and the term itself which has type *type*. Moreover, $A[\alpha]$ denotes that the type A may contain a type variable α. $A[B/\alpha]$ denotes the type A with the type B substituted for every occurrence of its type variable α

1. $*: ()$.
2. To each type is assigned a denumerable set of variables.
3. If $a: A$ and $b: B$, then $\langle a, b\rangle: A \times B$.
4. If $t: A \times B$, then $proj_0^{A,B} t: A$ and $proj_1^{A,B} t: B$.
5. If $b: B$ under the assumption $x: A$, then $\lambda x: A.b: A \rightarrow B$ without the assumption $x: A$.
6. If $f: A \rightarrow B$ and $a: A$, then $apply^{A,B}(f, a): B$.
7. If $t: A[\alpha]$ and α does not appear in the types of free variables of t, then given a type B, $t\{B/\alpha\}: A[B/\alpha]$.

AXIOMS:

1. $t = *$ where $t: ()$.
2. $proj_0^{A,B}\langle a: A, b: B\rangle = a$
   $proj_1^{A,B}\langle a: A, b: B\rangle = b$
3. $t = \langle proj_0^{A,B} t, proj_1^{A,B} t\rangle$.
4. $apply^{A,B}(\lambda x.a, b) = a[b/x]$.
5. $\lambda x.(apply^{A,B}(a, x)) = a$

6. $t\{A/\alpha\} = t_{(A/\alpha)}$

In the above axioms, $t_{(A/\alpha)}$ denotes the term t , in which the type A is substituted for every occurrence of the type variable α. For example, $(proj_0^{\alpha,B}a)_{(A/\alpha)}$ is the term $proj_0^{A,B}a_{(A/\alpha)}$.

The axiom 6 is essential in polymorphism. If we do not require it, the type system will be called rather *overloading* than polymorphism.

The type system presented above is a variant of *explicitly-typed ML*, or *Core-XML* of [7]. In [7], the *Core-XML* is proved to be equivalent to *Core-ML*, or *implicitly-typed ML* which is a usual definition of the language ML.

Our system is defined so that it can be clear that ML is a subsystem of $\mathcal{F}$ of Girard.

proposition 2.2 The system ML presented above is a subsystem of $\mathcal{F}$.

Proof

ML is embedded in $\mathcal{F}$ as $\widetilde{\mathrm{ML}}$ in the following way.

1. Types of ML are also those of $\widetilde{\mathrm{ML}}$.
2. Terms of ML are clearly those of $\widetilde{\mathrm{ML}}$. As for $\{\cdot\}$, a term $a\{B/\alpha\}$ in ML is translated into $\widetilde{\mathrm{ML}}$ as $(\Lambda\alpha.a)\{B\}$.

This translation clearly gives the embedding of ML into $\mathcal{F}$. ∎

Lastly, we define an inconsistent ML theory.

definition 2.3 An ML theory is *inconsistent* if it has a closed term $\perp : \alpha$.

3 Internal ccc theory

Preliminaries

To express the internal category theory, we use a variant of notations of [8]. In the subsequent sections, we fix an arbitrary category $\mathcal{E}$ closed under finite limits.

notation 3.1 We use the notations $\pi_j^{p,q}$ $(j = 0, 1)$ and $\langle f, g\rangle_{p,q}$ to denote the morphisms P_0, P_1 and T respectively in the diagram below.

$$\begin{tikzcd}
D \arrow[dr, "T"] \arrow[drr, bend left, "f"] \arrow[ddr, bend right, "g"'] & & \\
& P \arrow[r, "P_1"] \arrow[d, "P_0"] \arrow[dr, phantom, "\mathrm{PB}"] & A \arrow[d, "p"] \\
& B \arrow[r, "q"] & C_0
\end{tikzcd}$$

Other notation: if the diagram is explicitly labelled D, we sometimes write π_0^D, π_1^D, $\langle f, g\rangle_D$ to denote the above morphisms.
The same notations apply to equalizer diagrams.
We sometimes omit suffixes if they are clear from context.

definition 3.2 An internal category $\mathbf{C} = (C_0, C_1)$ in a category $\mathcal{E}$ consists of:

1. a pair of objects C_0 (the object of objects) and C_1 (the object of morphisms).
2. distinguished morphisms $C_1 \xrightarrow{dom} C_0$, $C_1 \xrightarrow{cod} C_0$, $C_0 \xrightarrow{\iota} C_1$, and $C_2 \xrightarrow{\bullet} C_1$, where C_2 satisfies

$$\begin{tikzcd}
C_2 \arrow[r, "\pi_0"] \arrow[d, "\pi_1"'] \arrow[dr, phantom, "\mathrm{PB}"] & C_1 \arrow[d, "dom"] \\
C_1 \arrow[r, "cod"'] & C_0
\end{tikzcd} \qquad (C_2)$$

3. $dom \cdot \iota = cod \cdot \iota = id_{C_0}$,
   $dom \cdot \bullet = dom \cdot \pi_0$,
   $cod \cdot \bullet = cod \cdot \pi_1$,
   $\bullet(id \times \bullet)_{C_3} = \bullet(\bullet \times id)_{C_3}$,
   $\bullet(id \times \iota) = \pi_0^{C_2}$,
   $\bullet(\iota \times id) = \pi_1^{C_2}$

remark 3.3 The arrow ι corresponds to the identity morphism and '$\bullet$' to the composition of morphisms in the internal category.

Internal product

definition 3.4 An internal category $\mathbf{C} = (C_0, C_1)$ is said to have the internal product if there exist the internal product operator $C_0^2 \xrightarrow{\wedge} C_0$ and the pairing operator $\Delta C_1 \xrightarrow{\langle\!\langle -,- \rangle\!\rangle} C_1$ which satisfy:

1. ΔC_1 is the equalizer of the diagram below.

$$\begin{tikzcd} \Delta C_1 \arrow[r, tail, "e"] & C_1^2 \arrow[r, shift left, "dom \cdot \pi_0"] \arrow[r, shift right, "dom \cdot \pi_1"'] & C_0 \end{tikzcd}$$

(E_Δ)

2.

$$\begin{tikzcd} \Delta C_1 \arrow[r, "\langle\!\langle -,- \rangle\!\rangle"] \arrow[dr, "dom \cdot \pi_0 \cdot e"'] & C_1 \arrow[d, "dom"] \\ & C_0 \end{tikzcd} \qquad \begin{tikzcd} \Delta C_1 \arrow[r, "\langle cod \cdot \pi_0 \cdot e,\, cod \cdot \pi_1 \cdot e \rangle"] \arrow[d, "\langle\!\langle -,- \rangle\!\rangle"'] \arrow[dr, phantom, "\mathrm{PB}"] & C_0^2 \arrow[d, "\wedge"] \\ C_1 \arrow[r, "cod"'] & C_0 \end{tikzcd}$$

(D_Δ)

3. The following diagram commutes, where $s = \langle \bullet\langle \pi_0 \cdot \pi_0, \pi_1 \rangle, \bullet\langle \pi_1 \cdot \pi_0, \pi_1 \rangle \rangle$.

$$\begin{tikzcd} \Delta C_1 \times_{C_0} C_1 \arrow[r, "s"] \arrow[d, "\langle\!\langle -,- \rangle\!\rangle \times id"'] & \Delta C_1 \arrow[d, "\langle\!\langle -,- \rangle\!\rangle"] \\ C_2 \arrow[r, "\bullet"'] & C_1 \end{tikzcd}$$

definition 3.5 (projection) Consider a morphism $F: DomF \longrightarrow C_0^2$ with codomain C_0^2. Let the morphism $T = \langle F, \iota \cdot \wedge \cdot F \rangle_{D_\Delta}: DomF \longrightarrow \Delta C_1$ be defined as in the diagram below.

$$\begin{tikzcd} DomF \arrow[dr, "T"] \arrow[drr, "F"] \arrow[ddr, "\iota \cdot \wedge \cdot F"'] & & \\ & \Delta C_1 \arrow[r] \arrow[d] \arrow[dr, phantom, "\mathrm{PB}"] & C_0^2 \arrow[d, "\wedge"] \\ & C_1 \arrow[r, "cod"'] & C_0 \end{tikzcd}$$

We define $Proj_j^F: DomF \longrightarrow C_1$ as $\pi_j \cdot e \cdot T (j = 0, 1)$.

The arrows $Proj_j^F$ correspond to the j-th projection in the sense of the following example $(j = 0, 1)$.

example 3.6 Let $\mathcal{E} = \mathbf{Set}$. Consider a small category $\mathbf{C}$ with finite products. We define its *internalization* $I(\mathbf{C}) = (I(\mathbf{C})_0, I(\mathbf{C})_1)$ as follows:

$$\begin{aligned} I(\mathbf{C})_0 &= \text{the set of objects of } \mathbf{C} \\ I(\mathbf{C})_1 &= \text{the set of morphisms of } \mathbf{C} \end{aligned}$$

The morphisms dom, cod, ι and $\bullet$ are defined in the usual sense. Its internal product operator is $a, b \mapsto a \times b$, with pairing operator $f, g \mapsto \langle f, g \rangle$. $Proj_j$ is identical to the function $a, b \mapsto \pi_j^{a,b}$ $(j = 0, 1)$.

More precisely, to two objects a and b in $\mathbf{C}$, there correspond morphisms $\tilde{a}, \tilde{b}: 1 \longrightarrow I(\mathbf{C})_0$. Consider a morphism $1 \xrightarrow{\langle \tilde{a}, \tilde{b} \rangle} I(\mathbf{C})_0$. $Proj_0^{\langle \tilde{a}, \tilde{b} \rangle}: 1 \longrightarrow I(\mathbf{C})_1$ corresponds to the first projection $\pi_0: a \times b \longrightarrow a$ in $\mathbf{C}$. In the same manner, $Proj_1^{\langle \tilde{a}, \tilde{b} \rangle}: 1 \longrightarrow I(\mathbf{C})_1$ corresponds to the second projection $\pi_1: a \times b \longrightarrow b$.

lemma 3.7 For every $F: DomF \rightarrow C_0^2$ and $G: DomG \rightarrow DomF$,
$Proj_j^{F \cdot G} = Proj_j^F \cdot G$ $(j = 0, 1)$.

Proof
$\langle Proj_0^{F \cdot G}, Proj_1^{F \cdot G} \rangle_{E_\Delta} = \langle F \cdot G, \iota \cdot \wedge \cdot F \cdot G \rangle_{D_\Delta} = \langle F, \iota \cdot \wedge \cdot F \rangle_{D_\Delta} \cdot G$. The last morphism is by definition $\langle Proj_0^F, Proj_1^F \rangle_{E_\Delta} \cdot G$. Therefore, $Proj_j^{F \cdot G} = Proj_j^F \cdot G$ $(j = 0, 1)$. ∎

We show some properties of $Proj$'s.

proposition 3.8 For every F with C_0^2 its codomain, $\langle\!\langle -,- \rangle\!\rangle \cdot \langle Proj_0^F, Proj_1^F \rangle = \iota \cdot \wedge \cdot F$. (This equality is the internal representation of the external equality $\langle \pi_0, \pi_1 \rangle = id$.)

Proof
Immediate from 3.5. ∎

proposition 3.9 For every F with C_0^2 its codomain,

$$dom \cdot Proj_j^F = \wedge \cdot F \tag{1}$$

$$cod \cdot Proj_j^F = \pi_j \cdot F \tag{2}$$

for $j = 0, 1$.

Proof
We may prove only the case $F = id_{C_0^2}$.
The proof of (1):

$$\begin{aligned}
\wedge &= dom \cdot \iota \cdot \wedge \\
&= dom \cdot \langle\!\langle -,- \rangle\!\rangle \langle Proj_0, Proj_1 \rangle \\
&= dom \cdot \pi_0 \cdot e \cdot \langle Proj_0, Proj_1 \rangle \\
&= dom \cdot Proj_0 = dom \cdot Proj_1
\end{aligned}$$

(2) is trivial from the definition of $Proj$. ∎

proposition 3.10 For every F with C_0^2 its codomain,

$$\begin{aligned}
\langle\!\langle -,- \rangle\!\rangle \cdot \langle \bullet(Proj_0^F \times id), \bullet(Proj_1^F \times id) \rangle &= \pi_1^{\wedge,cod} \\
\bullet(Proj_j^F \times \langle\!\langle -,- \rangle\!\rangle) &= \pi_j \cdot \pi_1^{\wedge,cod}
\end{aligned}$$

for $j = 0, 1$. In the external sense, these equalities correspond to the following ones:

$$\begin{aligned}
\langle \pi_0 \cdot h, \pi_1 \cdot h \rangle &= h \\
\pi_j \cdot \langle h_0, h_1 \rangle &= h_j \ (j = 0, 1)
\end{aligned}$$

for relevant h, h_0 and h_1.

Proof
We prove the case $F = id_{C_0^2}$. As for the first one,

$$\begin{aligned}
\langle\!\langle -,- \rangle\!\rangle \cdot \langle Proj_0 \bullet id, Proj_1 \bullet id \rangle &= \bullet \cdot (\langle\!\langle -,- \rangle\!\rangle \times id) \cdot (\langle Proj_0, Proj_1 \rangle \times id) \\
&= \bullet \cdot (\iota \times id) = \pi_1^{\wedge,cod}
\end{aligned}$$

As for the second equation, consider the following commutative diagram:

$$\begin{tikzcd}
A \arrow[dr, "T"] \arrow[drr, bend left, "\pi_0"] \arrow[ddr, bend right, "\langle\!\langle -,- \rangle\!\rangle \cdot \iota \cdot \pi_1"'] & & \\
& \Delta C_1 \arrow[r] \arrow[d] \arrow[dr, phantom, "\mathrm{PB}"] & C_0^2 \arrow[d, "\wedge"] \\
& C_1 \arrow[r, "cod"'] & C_0
\end{tikzcd}$$

,where A is defined as the pullback of:

$$\begin{tikzcd}
A \arrow[r, "\pi_1"] \arrow[d, "\pi_0"'] \arrow[dr, phantom, "\mathrm{PB}"] & C_0^2 \arrow[d, "id"] \\
\Delta C_1 \arrow[r, "\langle cod \cdot \pi_0 \cdot e, cod \cdot \pi_1 \cdot e \rangle"'] & C_0^2
\end{tikzcd}$$

In the above diagram, $T = \langle \bullet(Proj_0 \times \langle\!\langle -,- \rangle\!\rangle), \bullet(Proj_1 \times \langle\!\langle -,- \rangle\!\rangle) \rangle \cdot \pi_1$ because

$$\langle\!\langle -,- \rangle\!\rangle \cdot \langle \bullet(Proj_0 \times \langle\!\langle -,- \rangle\!\rangle), \bullet(Proj_1 \times \langle\!\langle -,- \rangle\!\rangle) \rangle \cdot \pi_1 = \langle\!\langle -,- \rangle\!\rangle$$

and,

$$\begin{aligned}
\langle cod \cdot \pi_0 \cdot e, cod \cdot \pi_1 \cdot e \rangle \langle \bullet(Proj_0 \times \langle\!\langle -,- \rangle\!\rangle), \bullet(Proj_1 \times \langle\!\langle -,- \rangle\!\rangle) \rangle \cdot \pi_1 &= \langle cod \cdot Proj_0 \cdot \pi_0, cod \cdot Proj_1 \cdot \pi_0 \rangle \\
&= \langle \pi_0 \cdot \pi_0, \pi_1 \cdot \pi_0 \rangle = \pi_0
\end{aligned}$$

On the other hand, $\langle cod \cdot \pi_0 \cdot e, cod \cdot \pi_1 \cdot e \rangle \cdot \pi_1 = \pi_0$ by the very definition. Therefore $T = \pi_1$, hence $\pi_1 = \langle \bullet(Proj_0 \times \langle\!\langle -,- \rangle\!\rangle), \bullet(Proj_1 \times \langle\!\langle -,- \rangle\!\rangle) \rangle \cdot \pi_1$. This implies that $\bullet(Proj_0 \times \langle\!\langle -,- \rangle\!\rangle) = \pi_j \cdot \pi_1$ for $j = 0, 1$.

Exponentiation

In the same way as product, *exponentiation* can be defined as:

definition 3.11 The internal category $\mathbf{C} = (C_0, C_1)$ is said to have the *internal exponential* if there exist *internal exponentiation operator* $C_0^2 \xrightarrow{\Rightarrow} C_0$ together with the currying operator $*C_1 \xrightarrow{(-)^*} C_1$ which satisfy:

1. $*C_1$ is the pullback of the diagram below.

$$\begin{tikzcd}
*C_1 \arrow[r] \arrow[d] \arrow[dr, phantom, "\mathrm{PB}"] & C_0^2 \arrow[d, "\wedge"] \\
C_1 \arrow[r, "dom"'] & C_0
\end{tikzcd} \qquad (E_\bullet)$$

2.

$$\begin{tikzcd}
*C_1 \arrow[r, "(-)^*"] \arrow[dr, "\pi_0 \cdot \pi_0^{E_\bullet}"'] & C_1 \arrow[d, "dom"] \\
& C_0
\end{tikzcd}
\qquad
\begin{tikzcd}
*C_1 \arrow[r, "\langle \pi_1 \cdot \pi_0^{E_\bullet}, cod \cdot \pi_1^{E_\bullet} \rangle"] \arrow[d, "(-)^*"'] \arrow[dr, phantom, "\mathrm{PB}"] & C_0^2 \arrow[d, "\Rightarrow"] \\
C_1 \arrow[r, "cod"'] & C_0
\end{tikzcd} \qquad (D_\bullet)$$

3. The following diagram commutes, where $t = \langle \wedge \cdot dom\pi_1, \pi_0 \bullet \langle\!\langle -,- \rangle\!\rangle \cdot \langle \pi_1 \bullet Proj_0, Proj_1 \rangle \rangle$.

$$\begin{tikzcd}
*C_1 \times C_1 \arrow[r, "t"] \arrow[d, "(-)^* \times id"'] & *C_1 \arrow[d, "(-)^*"] \\
C_2 \arrow[r, "\bullet"] & C_1
\end{tikzcd}$$

definition 3.12 (Eval) Consider a morphism $F: DomF \longrightarrow C_0^2$. We define $Eval^F$ as $\langle F, \iota\cdot \Rightarrow \cdot F \rangle_{D_\bullet}$.

$$\begin{tikzcd}
DomF \arrow[dr, "Eval^F"] \arrow[drr, "F"] \arrow[ddr, "\iota\cdot \Rightarrow \cdot F"'] & & \\
& *C_1 \arrow[r] \arrow[d] \arrow[dr, phantom, "\mathrm{PB}"] & C_0^2 \arrow[d, "\Rightarrow"] \\
& C_1 \arrow[r, "cod"'] & C_0
\end{tikzcd}$$

lemma 3.13 $Eval^{F \cdot G} = Eval^F \cdot G$
for every $F: DomF \rightarrow C_0^2$ and $G: DomG \rightarrow DomF$.

Proof
In the same manner as 3.7. ∎

proposition 3.14 The following diagrams commute.

$$\begin{tikzcd}
& & C_0 \\
Dom I \arrow[urr, "\pi_1 \cdot F"] \arrow[r, "Eval^F"] \arrow[drr, "\wedge \cdot \langle \Rightarrow, \pi_0 \rangle \cdot F"'] & *C_1 \arrow[r, "\pi_1^{E_\bullet}"] & C_1 \arrow[u, "cod"'] \arrow[d, "dom"] \\
& & C_0
\end{tikzcd}$$

Proof
We have only to prove the case $F = id$. As for the upper triangle, $cod \cdot \pi_1^{E_\bullet} \cdot Eval^F$ is trivially equal to π_1 from 3.11(2). As for the lower one, we first show $\pi_0 \cdot Eval = \langle \Rightarrow, \pi_0 \rangle$.

$$\begin{aligned}
\pi_1 \cdot \pi_0 \cdot Eval &= \pi_0 \\
\pi_0 \cdot \pi_0 \cdot Eval &= dom \cdot (-)^* \cdot Eval \\
&= dom \cdot \iota \cdot \Rightarrow \\
&= \Rightarrow
\end{aligned}$$

Therefor, $\pi_0 \cdot Eval = \langle \Rightarrow, \pi_0 \rangle$. Then from $dom \cdot \pi_1^{\wedge, dom} = \wedge \cdot \pi_0^{\wedge, dom}$, we obtain $dom \cdot \pi_1 \cdot Eval = \wedge \cdot \pi_0 \cdot Eval = \wedge \cdot \langle \Rightarrow, \pi_0 \rangle$.
∎

In the same manner as products, we obtain the following propositions.

proposition 3.15 For every F whose codomain is C_0^2,
$(-)^* \cdot \langle \langle dom \cdot \pi_1, \pi_0 \cdot F \cdot \pi_0^{\Rightarrow \cdot F, cod} \rangle, \pi_1^{E_\bullet} \cdot Eval^{F \cdot \pi_0}, \langle\!\langle -,- \rangle\!\rangle \langle \bullet \langle \pi_1, Proj_0^{\langle \Rightarrow \cdot F \cdot \pi_0, \pi_0 \cdot F \cdot \pi_0 \rangle} \rangle, Proj_1^{\langle \Rightarrow \cdot F \cdot \pi_0, \pi_0 \cdot F \cdot \pi_0 \rangle} \rangle \rangle$
$= \pi_1^{\Rightarrow \cdot F, cod}: DomF \times' C_1 \longrightarrow C_1$.

$$\begin{tikzcd}
DomF \times' C_1 \arrow[r] \arrow[d] \arrow[dr, phantom, "\mathrm{PB}"] & DomF \arrow[d, "\Rightarrow \cdot F"] \\
C_1 \arrow[r, "cod"'] & C_0
\end{tikzcd}$$

where $DomF \times' C_1$ denotes the pullback of the above diagram.
This equality represents the following (external) equality.

$$(Eval \cdot \langle (-) \cdot \pi_0, \pi_1 \rangle)^* = (-).$$

proposition 3.16 For every F whose codomain is C_0^2,

$$\bullet \langle \pi_1^{E_\bullet} \cdot Eval^{\pi_0^{E_\bullet}}, \langle\!\langle -,- \rangle\!\rangle \langle \bullet \langle \pi_1, Proj_0^{\langle cod \cdot \pi_1, \pi_0 \cdot \pi_0 \rangle} \rangle, Proj_1^{\langle cod \cdot \pi_1, \pi_0 \cdot \pi_0 \rangle} \rangle \cdot \langle \pi_0^{E_\bullet}, (-)^* \rangle \rangle = \pi_1^{E_\bullet}: *C_1 \longrightarrow C_1.$$

Externally, this corresponds to the equality:

$$Eval \cdot \langle (h)^* \cdot \pi_0, \pi_1 \rangle = h.$$

Terminal

definition 3.17 The internal category $\mathbf{C} = (C_0, C_1)$ is said to have the *internal terminal operator* $1 \xrightarrow{1} C_0$ and the !-operator $C_0 \xrightarrow{\dagger} C_1$ if they satisfy the following conditions:

1. .

$$\begin{tikzcd} C_0 \arrow[r, "\dagger"] \arrow[dr, "id"'] & C_1 \arrow[d, "dom"] \\ & C_0 \end{tikzcd} \qquad \begin{tikzcd} C_0 \arrow[r] \arrow[d, "\dagger"'] \arrow[dr, phantom, "\mathrm{PB}"] & 1 \arrow[d, "1"] \\ C_1 \arrow[r, "cod"'] & C_0 \end{tikzcd}$$

2. $\dagger \cdot dom = \bullet\langle \dagger \cdot dom, id\rangle$.

proposition 3.18

$$\begin{aligned} dom \cdot \dagger &= id \\ 1 \cdot ! &= cod \cdot \dagger \\ \bullet\langle \dagger \cdot dom, id\rangle &= \dagger \cdot dom \end{aligned}$$

Proof
Trivial from the definition. ∎

proposition 3.19 For every f such that $cod \cdot f = dom \cdot f = 1 \cdot !$, $f = \dagger \cdot 1 \cdot !$. In particular, $\dagger \cdot 1 = \iota \cdot 1$.

Proof
Since $cod \cdot f = 1 \cdot !$, f factors through $\dagger$ of the second diagram of 3.17-1. Let $f = \dagger \cdot h$. $h = dom \cdot \dagger \cdot h = dom \cdot f = 1 \cdot !$. Therefore $f = \dagger \cdot 1 \cdot !$. ∎

Internal ccc

definition 3.20 An internal category $\mathbf{C}$ is called an *internal ccc* if it has the internal terminal, the internal product and the internal exponential.

We show some typical examples of internal ccc's.

example 3.21 In Set,

1. Let $\mathbf{C}$ be a small ccc. Then its internalization $I(\mathbf{C})$ is an internal ccc. Its internal product operator, pairing operator is given in the example 3.6. Its exponential operator is $a, b \mapsto a^b$ with currying operator the function $f \mapsto f^*$. Its terminal operator is $() \mapsto 1$. This internalization accounts for the name, "internal ccc."
2. $\mathbf{1} = (1, 1)$ is also an internal ccc In $\mathbf{1}$, terminal, products and exponentials are degenerated ones: $1 \to 1$.
3. Let Ω be the set of propositions constructed by
   (a) atomic formulas including $\top$ and
   (b) $\wedge$ and $\supset$ as logical connectives and
   (c) their related axioms

   We assume that two propositions are identical if they are equivalent.
   Let Ω_1 be the equalizer of the following diagram:

   $$\begin{tikzcd} \Omega_1 \arrow[r, tail] & \Omega^2 \arrow[r, shift left, "\pi_1"] \arrow[r, shift right, "\wedge"'] & \Omega \end{tikzcd}$$

   $\mathbf{\Omega} = (\Omega, \Omega_1)$ is an internal ccc. Its internal product operator is $\wedge$ together with the internal exponential operator $\supset$ and the terminal operator $\top$. Other operators are trivially defined.
4. Let $\mathbf{\Omega}^H$ be $\mathbf{\Omega}$ added with $\perp$, the absurdity as an atomic formula. $\mathbf{\Omega}^H = (\Omega^H, \Omega_1^H)$ is the heyting algebra on Ω^H in which Ω_1^H defines its order relation.

Cases (3) and (4) are important. In topos theory or PL-category [14], they are used as the representing object: the property that $\mathbf{\Omega}$ has the internal ccc structure is intensively used to classify the structure of the category in question. Furthermore, Ω_1 can be viewed as defining a partial order on Ω, such that $\mathbf{\Omega}$ is a $\wedge$-semi-lattice. Moreover, in case (4), $\mathbf{\Omega}^H$ defines a heyting algebra.

definition 3.22 In the internal category $\mathbf{C}$, a morphism $t_n: C_0^n \longrightarrow C_1$ is called an *internal operator* of arity n ($n = 0, 1, 2, \cdots$).
Furthermore, if $dom \cdot t_n = 1 \cdot !$, it is also called an *internal element* of *type* $cod \cdot t_n$.

example 3.23 1. $eval: C_0^2 \to C_1$ is an internal operator of arity 2.
2. For $\Delta: C_0 \xrightarrow{\langle id, id\rangle} C_0^2$, $Proj_j^\Delta: C_0 \to C_1$ is an internal operator of arity 1.

3. In the notation of 3.6, $Proj_j^{\langle \bar{a}, \bar{b}\rangle}: 1 \longrightarrow C_1$ is also an internal operator of arity 0.

4. The term $(-)^* \cdot \langle\langle 1 \cdot !, id\rangle, Proj_j^{\langle 1\cdot !, id\rangle}\rangle$ is an internal element of type $\Rightarrow \cdot \langle 1 \cdot !, id\rangle$.

Proof

We can easily show that every component of the diagram below commutes.

$$\begin{tikzcd}
C_0 \arrow[rr, "1 \cdot !"] \arrow[d, equal] & & C_0 \\
C_0^2 \arrow[r, "{\langle\langle 1 \cdot !, \wedge\rangle, Proj_j^{\langle 1\cdot !, \wedge\rangle}\rangle}"] \arrow[d, equal] & {*C_1} \arrow[r, "(-)^*"] \arrow[ur, "\pi_0 \cdot \pi_0"] \arrow[dr, "\Rightarrow \cdot \pi_0"'] & C_1 \arrow[u, "dom"'] \arrow[d, "cod"] \\
C_0^2 \arrow[rr, "\Rightarrow \langle 1 \cdot !, \wedge\rangle"] & & C_0
\end{tikzcd}$$

∎

In general, this construction corresponds to "naming" in the (external)ccc theory. Through this naming, every operator is translated to an element. (for example, see [9]).

definition 3.24 (naming) We call the morphism
$\lceil - \rceil = (-)^* \cdot \langle\langle 1 \cdot !, dom\rangle, \bullet\langle id, Proj_1^{\langle 1\cdot !, dom\rangle}\rangle\rangle : C_1 \to *C_1 \to C_1$ the *naming operator*.

For a morphism $Domf \xrightarrow{f} C_1$, we denote the composition $\lceil - \rceil \cdot f$ by $\lceil f \rceil$. As in the (external) ccc-theory, the internal composition of internal operators can be represented in terms of *internal elements*.

proposition 3.25 Given operators $C_0^n \xrightarrow{f} C_1$ and $C_0^n \xrightarrow{a} C_1$ such that $cod \cdot a = dom \cdot f$, $\lceil f \rceil$ is an internal element which satisfies:

$$\bullet\langle \pi_1^{E_\bullet} \cdot Eval, \langle\langle -, - \rangle\rangle \langle \bullet\langle \lceil f \rceil, \dagger \cdot dom \cdot a\rangle, a\rangle\rangle = \bullet\langle f, a\rangle.$$

Proof

By routine calculations, $dom \cdot \lceil - \rceil = 1 \cdot !$. Therefore, $dom \cdot \lceil f \rceil = 1 \cdot !$, an internal element. The equality can be proved by easy calculations. ∎

theorem 3.26 (equational presentation of internal ccc) Equalities 3.7, 3.8, 3.9, 3.10, 3.13, 3.15, 3.16, 3.18 give the equational presentation of the internal ccc.

Precisely, given the following data:

$$\iota, dom, cod, \bullet, \wedge, \langle\langle -, - \rangle\rangle, \Rightarrow, (-)^*, 1, \dagger$$

and if for each morphism F whose codomain is C_0^2, there are given $Proj_0^F, Proj_1^F, eval^F$, and equalities listed as above, we have an internal ccc structure.

Proof

This theorem is the translation of the equational presentation of (external) ccc to that of the internal ccc. The proof is completely analogous to that of the case of external ccc. (see, for example, [9]) ∎

Using the equational presentation of internal ccc, we can translate theorems on external ccc into those on internal ccc. Among them, we present here the functional completeness of internal ccc.

definition 3.27 Let X be given in the diagram below. Given an internal ccc $\mathbf{C} = (C_0, C_1)$, its polynomial $\mathbf{C}[X]$ on a *variable* X is the internal ccc which is freely generated from $\mathbf{C}$ and X.

$$\begin{tikzcd}
 & C_0 \\
C_0 \arrow[ur, "1 \cdot !"] \arrow[r, "X"] \arrow[dr, "id"'] & C_1 \arrow[u, "dom"'] \arrow[d, "cod"] \\
 & C_0
\end{tikzcd}$$

Given a morphism $F: DomF \to C_0$, a variable X^F of type F is defined as $X \cdot F$. a variable of type id is simply written as X.

The following theorem is the straightforward translation of the corresponding theorem on the external ccc.

proposition 3.28 (functional completeness of internal ccc) Given an n-ary operator $C_0^n \xrightarrow{\varphi[X^F]} C_1$ of $\mathbf{C}[X]$ $(n = 0, 1, \cdots)$, there is a unique operator $C_0^n \xrightarrow{f} C_1$ of $\mathbf{C}$ such that $\varphi[X^F] = \bullet\langle f, X^F \cdot dom \cdot f\rangle$.

notation 3.29 By $(X^F).\varphi[X^F]$, we denote the operator f in the above proposition.

example 3.30 $(X).X = \iota$ since $X = \bullet\langle \iota, X\rangle$.
$(X^A).X^A = \iota \cdot A$.

The following definition is also the translation of the corresponding definition on the external ccc.

definition 3.31 An internal ccc $\mathbf{C} = (C_0, C_1)$ is *degenerated* if there exists an operator $C_0 \xrightarrow{\text{¡}} C_1$ such that $\bullet\langle \text{¡}, \dagger \rangle = \iota$. Note that this equality implies that ¡ is an internal element of type *id*.

$$\begin{tikzcd} & C_0 \\ C_0 \arrow[ur, "1 \cdot !"] \arrow[r, "\text{¡}"] \arrow[dr, "id"'] & C_1 \arrow[u, "dom"'] \arrow[d, "cod"] \\ & C_0 \end{tikzcd}$$

Externally, this means that every $a \xrightarrow{!} 1$ has its inverse, that is, that every object is isomorphic to 1.

For example, $\mathbf{1} = (1, 1)$ is degenerated.

Suppose that an (external) ccc $\mathbf{C}$ has the initial object 0. It is well known that it is degenerated when there exists a morphism $1 \rightarrow 0$. The similar situation also appears in the relation between an inconsistent **ML**-theory and a degenerated internal ccc.

4 Relation between internal ccc and ML

In this section, we investigate the relation between internal ccc and **ML**.

theorem 4.1 From an internal ccc $\mathbf{C} = (C_0, C_1)$, We can construct an **ML**-theory $\mathcal{L}(\mathbf{C})$.

construction

Types of $\mathcal{L}(\mathbf{C})$ are morphisms $C_0^n \rightarrow C_0 (n = 0, 1, 2, \cdots)$ with

1. Primitive types of $\mathcal{L}(\mathbf{C})$ are those morphisms $1 \longrightarrow C_0$ $(n = 0, 1, 2, \cdots)$.
2. The j-th projection $C_0^n \xrightarrow{\pi_j} C_0$ $(0 \leq j < n)$ is translated to the j-th variable.
3. As for type constructors,
   $\times$ is interpreted as $C_0^2 \xrightarrow{\wedge} C_0$,
   $\rightarrow$ is interpreted as $C_0^2 \xrightarrow{\Rightarrow} C_0$,
   and () is interpreted as $1 \xrightarrow{1} C_0$.

Terms of $\mathcal{L}(\mathbf{C})$ are internal elements of the polynomial $\mathbf{C}[X]$ with X a variable. In particular,

1. Primitive terms are those morphisms $1 \longrightarrow C_1$.
2. A variable is $X : C_0 \longrightarrow C_1$.
3. $*$ is $1 \xrightarrow{1} C_0 \xrightarrow{\dagger} C_1 = \iota \cdot 1$.
4. Given two internal elements a, b of arity n, $\langle a, b \rangle$ is $C_0^n \xrightarrow{\langle a,b \rangle} \Delta C_1 \xrightarrow{\langle\!\langle -,- \rangle\!\rangle} C_1$.
5. Given types $A, B : C_0^n \longrightarrow C_0$ and an internal element c of type $A \times B = \wedge \cdot \langle A, B \rangle$, $proj_j^{A,B} c$ is interpreted as $\bullet\langle Proj_j^{\langle A,B \rangle}, c \rangle$ for $j = 0, 1$.
6. Given an internal element $C_0^n \xrightarrow{b[X^A]} C_1$, $\lambda X : A.\ b[X^A]$ is interpreted as $\lceil (X^A).b[X^A] \rceil$.
7. Given types $A, B : C_0^n \longrightarrow C_0$ and internal elements f with type $\Rightarrow \cdot \langle A, B \rangle$ and a with type A, $apply^{A,B}(f, a)$ is interpreted as $\bullet\langle \pi_1^{E_*} \cdot Eval^{\langle A,B \rangle}, \langle\!\langle -,- \rangle\!\rangle \cdot \langle f, a \rangle \rangle$.
8. Given a type $T : C_0^n \longrightarrow C_0$ and an internal element $C_0^m \xrightarrow{f} C_1$, if α denotes the j-th type variable, $f\{T/\alpha\}$ is $f \cdot \tilde{T}$, where $\tilde{T}$ satisfies that $\pi_j \cdot \tilde{T} = T$ and $\pi_{j'} \cdot \tilde{T} = \pi_{j'}$ for $j' \neq j$. [1]

The type of a given term $f : C_0^n \longrightarrow C_1$ is defined as $cod \cdot f$.

This construction is completely the internal version of the translation from ccc to the typed λ-calculus. Therefore, the above construction clearly satisfies the axioms of **ML** except AXIOM 6 of the definition 2.1. Explicitly, AXIOM 1 is the immediate result of 3.24, AXIOM 2 and 3 are also the immediate results of 3.13. AXIOM 4 and 5 can be proved with routine calculations using 3.15, 3.16 and 3.28.

AXIOM 6 is easily proved by the induction on the construction of terms using 3.7 and 3.13.

definition 4.2 Given an **ML**-theory **L**, its associated internal category $\mathcal{C}(\mathbf{L})$ in **Set** is given as:
$\mathcal{C}(\mathbf{L})_0 =$ the set of types which do not depend on type variables.
$\mathcal{C}(\mathbf{L})_1 = \{(X : A, b[X] : B) |\ A$ and B do not depend on type variables. $\}$
dom is the function $(X : A, b[X] : B) \mapsto A$, together with *cod* the function $(X : A, b[X] : B) \mapsto B$, ι the function $A \mapsto (X : A, X : A)$ etc.

proposition 4.3 $\mathcal{C}(\mathbf{L})$ is an internal ccc in **Set**.

Proof
Immediate ▮
The following proposition is an immediate consequence of the definitions of $\mathcal{C}$ and $\mathcal{L}$.

[1] The problem of the linearlization of type variables arises here. We do not discuss its details.

proposition 4.4 Assume that an ML-theory L satisfies that any primitive term does not depend on type variables. In this case, $\mathcal{L}(\mathcal{C}(\mathbf{L}))$ has the one-to-one correspondence to **L**.

The above results imply that the universe of types that an ML-theory handles is the cartesian closure of primtive types under $\rightarrow$ and $\times$. In this sense, an ML-theory can be said *predicative*.

remark 4.5 Note that the assumption in 4.4 is not essential at least when we work in **Set**. Given a constant term p_α which depends on a type variable α, there is a corresponding function $\tilde{p}: \mathcal{C}(\mathrm{L})_0 \longrightarrow \mathcal{C}(\mathrm{L})_1$ such that $\tilde{p}(A) = (*, p\{A/\alpha\})$.

The following corollary is an immediate result from the construction of $\mathcal{C}$ and $\mathcal{L}$.

corollary 4.6 ML is conservative over typed λ-calculus.

The stronger form is shown in [1]. Our $\mathcal{C}$ and $\mathcal{L}$ give a natural proof to the corollary.

We clarify the relation between degenerated internal ccc and inconsistent ML in the following proposition.

proposition 4.7 If an internal ccc **C** is degenerated, then $\mathcal{L}(\mathbf{C})$ is inconsistent.

Proof

The term ¡ has certainly type *id*. This means $\perp : \alpha$ if $\perp$ is interpreted as ¡. ∎

We are mainly concerned with consistent ML theory. Also in the coherent semantics of $\mathcal{F}$ [6], the type $\Lambda\alpha.\alpha$ is interpreted as $\emptyset$. This corresponds to our usual manner that we deal with only non-degenerated ccc's.

Acknoledgements.

The author is grateful to Prof. Yoneda Nobuo for his helpful comments and encouragement.

References

[1] V. Breazu-Tannen, A. Meyer: Polymorphism IS Conservative over Simple Types, *Proc. 2nd Ann. Symp. on Logic in Computer Science*, 1987,pp. 7-17.

[2] Th. Coquand, G. Huet: The Calculus of Constructions, *Information and Constrol*, vol. 76, 1988 pp. 95-120.

[3] Th. Coquand: An Analysis of Girard's Paradox, *Proc. 1st Ann. Symp. on Logic in Computer Science*, 1986.

[4] Th. Ehrhard: A Categorical Semantics of Constructions, *Proc. 3rd Ann. Symp. on Logic in Computer Science*, 1988, pp. 264-273.

[5] J. Y. Girard: Une extension de l'interprétation de Gödel à l'analyse et son application à l'elimination des coupures dans l'analyse et la thêorie des types, *Proc. 2nd Scandinavian Logic Symposium*, eds. J.E. Fenstatd, North-Holland, 1971, pp. 63-92.

[6] J. Y. Girard: The system F of variable types, fifteen years later, *Theoretical Computer Science*, vol. 45, no. 2, 1986, pp. 159-192.

[7] J.C. Mitchell, R. Harper: The essence of ML, *Proc. of 15th POPL*, 1988, pp. 28-46

[8] P.T. Johnstone: *Topos Theory*, Academic Press, 1977.

[9] J. Lambek and P. Scott: *Introduction to Higher Order Categorical Logic*, Cambridge University Press, 1986.

[10] S. MacLane: *Categories for Working Mathematician*, Springer-Verlag, 1971.

[11] D. MacQueen, G. Plotkin, R. Sethi: An ideal model of recursive polymorphis types, *Information and Control*, vol 71, 1986, pp. 95-130.

[12] J. C. Reynolds: Toward a Thoery of Type Structure, *Proc. Colloque sur la Programmation* (LNCS 19), 1974, pp. 408-425.

[13] R. Seely: Locally Cartesian Closed Categories and Type Theory, *Math. Proc. Camb. Phil. Soc*, vol. 95, 1984.

[14] R. Seely: Categorical Semantis for Higher Order Polymorphic Lambda Calculus, *J. Symbolic Logic*, vol. 52, no. 4, 1987, pp. 969-988.

[15] Standard ML Reference Manual(PRELIMINARY),AT&T, 1988.