TR-104

Towards Automated Synthetic Differential Geometry 1
—basic categorical construction—

Susumu Hayashi
(Kyoto University)

March. 1985

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

Towards Automated Synthetic Differential Geometry I

-basic categorical construction-

Susumu HAYASHI

Research Institute for Mathematical Science,

Kyoto University,

Sakyo, Kyoto, Japan

INTRODUCTION

This is the first report on SDG project, which consists a part of CAP project of ICOT WG5. The aim of CAP project is to create proof checkers for some specific fields of mathematics in order to investigate artificial intelligence for solving mathematics and the ideal man-machine interface in such activities [Furukawa & Yokoi 84]. SDG project aims to create a proof checker for a new field of mathematics, Synthetic differential geometry (SDG) initiated by F.W. Lawvere and A. Kock. SDG is a kind of non-standard geometry, in which there are *sufficiently many* nilpotent elements on the real line. Hence the universe of discourse of SDG *cannot* be the category of sets **SET**, but a kind of topos called a *well-adapted model* [Kock 81]. A remarkable feature of SDG is that large parts of proofs in SDG are algebraic or categorical computations. Hence it is probable that many theorems in SDG are easily proved by proof checkers furnished with powerful rewriters like **EKL** by [Ketonen & Weening 84]. (See Appendix A for a small experiment of SDG with **EKL**, in which a fairy short proofs of rules about differentiation are given.)

In these notes, we present a simple categorical construction, on which our language for SDG will be based. As noted above, the universe of SDG is other categories than **SET**, and axioms and concepts of SDG are stated in diagrammatic terms. But, diagrammatic terms like, pullback, commutative diagram etc., seem not to be so appropriate as a language for a proof checker. Diagrams are eminently useful to crystallize some ideas in SDG, but in general they are less intuitive at least for people do not care category theory. Besides, they are space consuming and we have not yet had enough experiences in using such graphic objects as a language for proof checkers. (No one would like to draw a square with labeled sides on a display than simply typing in as $f\circ g = g\circ f$). Fortunately, there is another way stating and proving facts in SDG by logical languages. The way, categorical logic, is a standard device of SDG, and we have decided to adopt it as the language for our SDG proof checker. Some statements in SDG are naturally expressed by *infinitary languages*. For example, the Axiom" of [Kock 81] in a topos **E** is stated as follows:

$$\mathbf{E} \models \ \bigwedge k \bigwedge n \forall f{:}D_k(n) \to R\, \exists! p \in P(k,n)[f \text{ is given by } p],$$

where k and n are variables run through *actual* natural numbers, $D_k(n)$ is a subobject of D^n and $R(k,n)$ is the object of the polynomials (with coefficients from R) in n variables and of total degree less than k. The sequence of objects $\{D_k(n)\}_{k,n\in N}$ should be constructed outside **E**. Hence the variables k and n live in **SET**. On the contrary, the variable f lives in **E**. Hence the above formula involves kinds of variables, i.e., variables living in **SET** and variables living in **E**. The standard viewpoint to this problem is to think **E** as a formal system. Namely f is considered as a variable of the formal system **E** and k, n are considered as *metavariables*. However, this viewpoint implies two levels of logic, which we do not like. In Part I of [Kock 81], those two kinds of variables seem to be treated without any distinctions. We hope to do things in our proof checker as Kock did. For this end, we will glue up the two universes of discorce **SET** and **E** into a topos called the *envelope* of **E** the logic of which our proof checker will be based on. The readers are assumed to be familiar with fundamentals of topos theory (cf. [Fourman 77], [Johnstone 77], [Makkai & Reyes 77]).

1. THE ENVELOPE OF A TOPOS

Let **E** be an arbitrary category. Then its envelope $\bar{\mathbf{E}}$ is defined as follows:

DEFINITION 1. *An object of* $\bar{\mathbf{E}}$ *is a pair*

$$\langle I, \{A_i\}_{i\in I}\rangle,$$

where I *is a set and* A_i *is an object of* **E** *for each* $i\in I$. *We will often write* $\{A_i\}_{i\in I}$ *or* $\{A_i\}_i$ *for* $\langle I, \{A_i\}_{i\in I}\rangle$.

A morphism h *from* $\{A_i\}_{i\in I}$ *to* $\{B_j\}_{j\in J}$ *is a pair*

$$h = \langle f: I\to J, \{\phi_i\}_{i\in I}\rangle,$$

when ϕ_i *is a morphism in* **E** *from* A_i *to* $B_{f(i)}$ *for each* $i\in I$.

The composition of two morphism $\langle f,\{\phi_i\}_i\rangle$, $\langle g,\{\psi_j\}_j\rangle$ *is defined by*

$$\langle g\circ f, \{\psi_{f(i)}\circ \phi_i\}_i\rangle.$$

The function f is called the *base part* of h and is denoted by $\nabla(h)$, and the family of morphisms $\{\phi_i\}_{i\in I}$ is called the *fiber part* of h and is denoted by $h^\natural$. Hence, for each $i\in\nabla(h)$, $h^\natural_i=\phi_i$. We will show this category $\bar{\mathbf{E}}$ is a topos, provided so is **E**. The details of the proof will be left for readers, for they are completely routine.

PROPOSITION 1. *If* **E** *has all (finite) left limits, then* $\bar{\mathbf{E}}$ *has all (finite) left limits.*

Proof. The terminal object of $\bar{\mathbf{E}}$ is

$$\langle 1,\{1\}\rangle.$$

The product of $\{A_i\}_{i\in I}$ and $\{B_j\}_{j\in J}$ is given by

$$\langle I\times J, \{A_i\times B_j\}_{i\in I, j\in J}\rangle.$$

The projections are given by

$$\pi_1=\langle \pi_1: I\times J\to I, \{\pi_1: A_i\times B_j\to A_i\}_{i\in I, j\in J}\rangle,$$

$$\pi_2=\langle \pi_2: I\times J\to J, \{\pi_2: A_i\times B_j\to B_j\}_{i\in I, j\in J}\rangle.$$

The constructions of set-indexed products are the same as the above.

The equalizer of the parallel morphisms

$$\{A_i\}_i \underset{\langle g,\ \{\psi_i\}_i\rangle}{\overset{\langle f,\ \{\phi_i\}_i\rangle}{\rightrightarrows}} \{B_j\}_j,$$

is given by

$$\{E_i\}_{i\in K} \xrightarrow{\langle k,\ \{\epsilon_i\}_{i\in K}\rangle} \{A_i\}_{i\in I},$$

where

$$K=\{i\in I \mid f(i)=g(i)\},$$

k is the embedding of K into I,

and

$$E_i \xrightarrow{\ \epsilon_i\ } A_i \underset{\psi_i}{\overset{\phi_i}{\rightrightarrows}} B_{f(i)}$$

is an equalizer diagram for each $i\in K$. □

PROPOSITION 2. *If* $\mathbf{E}$ *is a cartesian closed category (c.c.c.) with all set-indexed products, then so is* $\bar{\mathbf{E}}$.

Proof. Let A be $\{A_i\}_{i\in I}$ and B be $\{B_j\}_{j\in J}$. Then B^A and its evaluation map $ev : B^A\times A\to B$ is given by

$$\langle J^I,\ \{\Pi_{i\in I}B_{f(i)}{}^{A_i}\}_{f\in J^I}\rangle,$$

$$\langle ev : J^I\times I\to J,\ \{ev_{f,i}\}_{f\in J^I, i\in I}\rangle,$$

where

$$\begin{array}{ccc} ev_{f,i}=(\Pi_{i\in I}B_{f(i)}{}^{A_i})\times A_i & \xrightarrow{\qquad\qquad\qquad} & B_{f(i)} \\ \Big\downarrow{\scriptstyle \pi_i\times id} & \circlearrowright & \nearrow{\scriptstyle ev} \\ B_{f(i)}{}^{A_i}\times A_i & & \end{array}$$

for each $f \in J^I$ and $i \in I$. Let C be $\{C_k\}_{k \in K}$ and let $\langle f, \{\phi_{k,i}\} \rangle$ be a morphism from $C \times A$ to B. Then the transpose of the morphism is given by

$$\langle f^{\wedge}, \{\langle \phi_{k,i}{}^{\wedge} \rangle_{i \in I}\}_{k \in K} \rangle : C \longrightarrow B^{A},$$

where $f^{\wedge}$ and $\phi_{k,i}{}^{\wedge}$ are the transposes of f and $\phi_{k,i}$, respectively. Note that we wrote (and will write) $\langle \psi_i \rangle_{i \in I}$ for the morphism from A to $\mathrm{P}_{i \in I} B_i$ such that $p_i \circ \langle \psi_i \rangle_{i \in I} = \psi_i$, for the family of morphisms $\{\psi_i : A \rightarrow B_i\}_{i \in I}$. □

PROPOSITION 3. *For any category* $\mathbf{E}$, $\bar{\mathbf{E}}$ *has all set-indexed coproducts. Furthermore, they are disjoint and universal* (cf. [Johnstone 77]).

Proof. Let $\{A_k\}_{k \in K}$ be a family of objects of $\bar{\mathbf{E}}$ indexed by the set K. Let A_k be $\{A^k{}_i\}_{i \in I}$ for each $k \in K$. Then the coproduct is given by

$$\coprod_{k \in K} A_k = \langle \coprod_{k \in K} I_k, \{A^k{}_i\}_i \rangle.$$

The inclusion map from A_k to $\coprod_{k \in K} A_k$ is given by

$$\langle \iota_k, \{A^k{}_i \rightarrow A^k{}_i\}_{i \in I} \rangle,$$

where ι_k is the inclusion map form I_k to $\coprod_{k \in K} I_k$. Disjointness and universality of the coproducts are trivial. □

By the above proposition, $\{A_i\}_{i \in I}$ is the coproduct of $\{\langle 1, \{A_i\} \rangle\}_{i \in I}$ in $\bar{\mathbf{E}}$. It looks as if $\langle 1, \{\coprod_i A_i\} \rangle$ were the coproduct of $\{\langle 1, \{A_i\} \rangle\}_{i \in I}$, but this is not generally true. But there is a canonical epimorphism from $\{A_i\}_{i \in I}$ to $\langle 1, \{\coprod_i A_i\} \rangle$ given by $\langle I \rightarrow 1, \{\iota_i\}_{i \in I} \rangle$ where ι_i is the inclusion map from A_i to $\coprod_i A_i$. If $\mathbf{E}$ has the initial object and I has two elements at least, then this epimorphism is never a monomorphism. We will give a logical characterization of $\langle 1, \{\coprod_i A_i\} \rangle$ in the next section.

The condition a morphism is monomorphic (epimorphic) in $\bar{\mathbf{E}}$ is as follows.

LEMMA 1. *(i) If* $\mathbf{E}$ *has an initial object, then a morphism* $\langle f, \{\phi_i\} \rangle$ *in* $\bar{\mathbf{E}}$ *is monomorphic iff* f *is injective and each* ϕ_i *is monomorphic. (ii) For any category* $\mathbf{E}$, *a morphism* $\langle f, \{\phi_i\} \rangle$ *in* $\bar{\mathbf{E}}$ *is epimorphic iff* f *is surjective and each* ϕ_i *is epimorphic.*

By the above preparations, we can prove the main result.

THEOREM 1. *If* $\mathbf{E}$ *is a topos, then so is* $\bar{\mathbf{E}}$.

Proof. Obviously $\bar{\mathbf{E}}$ is locally small, if so is $\mathbf{E}$. Let $\{A_i\}_{i \in I}$ be a set of generators of $\mathbf{E}$. Then $\{\langle 1, \{A_i\} \rangle\}_{i \in I}$ is a set of generators of $\bar{\mathbf{E}}$. By the above propositions, $\bar{\mathbf{E}}$ has finite left limits, exponentials and set-indexed coproducts which are universal and disjoint. Hence it is enough tp prove $\bar{\mathbf{E}}$ has a subobject classifier. The subobject classifier is given by

$$\Omega=\langle \mathbf{bool}, \{\Omega_{tt}, 1_{ff}\}\rangle,$$

$$true : \langle 1, \{1\}\rangle \xrightarrow{\langle true\ :\ 1\to \mathbf{bool},\ \{true:1\to\Omega_{tt}\}\rangle} \Omega,$$

where

$$\Omega_{tt}= \text{the subobject classifier of } \mathbf{E},$$

$$1_{ff}= \text{the terminal object of } \mathbf{E},$$

$$\mathbf{bool}=\{tt, ff\} \quad (tt \text{ is } true \text{ and } ff \text{ is } false),$$

Let $\langle f,\{\phi_i\}_i\rangle$ be a monomorphism from $\{A_i\}_{i\in I}$ to $\{B_j\}_{j\in J}$ in $\bar{\mathbf{E}}$. By Lemma 1, we may assume I is a subset of J and ϕ_i is a monomorphism. Then its classifier is given by

$$\langle ch_f, \{\psi_j\}_{j\in J}\rangle : \{B_j\}_j \longrightarrow \Omega,$$

where ch_f is the characteristic function of f, i.e. the classifier of f in **SET**, and if $j\in I$ then

$$\psi_j : B_j \longrightarrow \Omega_{tt}$$

is the classifier of ϕ_j in $\mathbf{E}$, otherwise ψ_j is the uniquely determined morphism from B_j to 1_{ff}. □

Let $\phi=\langle f,\{\phi_i\}_{i\in I}\rangle$ be a predicate of type $A=\{A_i\}_{i\in I}$, i.e. a morphism from A to Ω. Then ϕ is uniquely determined by the set $S=\{i\in I \mid f(i)=tt\}$ and morphisms $\{\phi_i\}_{i\in S}$. So we sometimes write

$$\langle S, \{\phi_i\}_{i\in S}\rangle$$

for the predicate ϕ.

In the rest of the paper, **E** always stands a topos. We will show how to embed **SET** and **E** into $\bar{\mathbf{E}}$. The topos **SET** is embeddable into any topos **E** by the constant sheaf functor Δ:**E**→**SET**, but it is not always true that Δ is full and preserves exponentials. But the constant sheaf functor from **SET** to $\bar{\mathbf{E}}$ is a full-faithful embedding preserving exponentials. In our case, the constant sheaf functor preserves almost all things of **SET** except the subobject classifier. The preservation of logical operators and quantifiers will be examined in the next section. We will here define the embeddings and prove its fundamental properties.

DEFINITION 2. *The embedding* Δ: **SET**→**E** *is the constant sheaf functor, i.e. it is given by*

$$\Delta(I)=\langle I, \{1_i\}_i\rangle,$$

$$\Delta(f)=\langle f,\ \{1_i \to 1_{f(i)}\}_i\rangle.$$

The embedding $\#(-):\mathbf{E}\to\bar{\mathbf{E}}$ *is given by*

$$\#(A)=\langle 1,\ \{A\}\rangle,$$

$$\#(f)=\langle 1\to 1,\ \{f\}\rangle,$$

We may identify I *with* $\Delta(I)$ *and* A *with* $\#(A)$*, respectively. So we sometimes write* I (A) *instead of* $\Delta(I)$ $(\#(A))$.

PROPOSITION 4. *(i)* Δ *and* $\#$ *are full-faithful. (ii)* Δ *and* $\#$ *preserves exponentials. (iii)* Δ *has a left adjoint* ∇ *and a right adjoint* Γ*. Hence it preserves all limits and colimits. (iv)* $\#$ *has a left adjoint* Π*. Hence it preserves all limits. But* $\#$ *does not preserve* $1 \amalg 1$*, so it never have right adjoint. Hence we have the following adjoint pairs*

$$\mathbf{SET}\ \underset{\Gamma}{\overset{\nabla}{\leftleftarrows}}\!\!\overset{\Delta}{\to}\ \bar{\mathbf{E}}\ \underset{\#}{\overset{\Pi}{\rightleftarrows}}\ \mathbf{E}$$

Proof. (i) Trivial. (ii) By Proposition 2.
(iii) For any topos, the global section functor Γ is the right adjoint of Δ. The left adjoint of Δ is the base part functor ∇, i.e.

$$\nabla(\langle I,\ \{A_i\}_i\rangle)=I,$$

$$\nabla(\langle f,\ \{\phi_i\}_i\rangle)=f.$$

(iv) We define a functor Π from $\bar{\mathbf{E}}$ to $\mathbf{E}$ by

$$\Pi(\langle I,\ \{A_i\}_i\rangle)=\Pi_{i\in I}A_i.$$

$$\begin{array}{ccc}
\Pi(\langle I,\ \{A_i\}_i\rangle) & \xrightarrow{\ \pi_i\ } & A_i \\
\Big\downarrow{\scriptstyle \Pi(\langle f,\ \{\phi_i\}_i\rangle)} & & \Big\downarrow{\scriptstyle \phi_i} \\
\Pi(\langle J,\ \{B_j\}_j\rangle) & \xrightarrow[\ \pi_j\ ]{} & B_{f(i)}
\end{array}$$

Then Π is the left adjoint of $\#$. On the contrary, $\#$ does not preserve $1 \amalg 1$, since

$$\#(1) \sqcup \#(1) = \{\{*, **\}, \{1_*, 1_{**}\}\},$$
$$\neq \{\{*\}, \{1 \sqcup 1\}\}$$
$$= \#(1 \sqcup 1).$$

Hence # has no right adjoint. □

DEFINITION 3. *An object A of* $\bar{\mathbf{E}}$ *is said discreate iff* $A \in \Delta(\mathbf{SET})$. *An object A of* $\bar{\mathbf{E}}$ *is said smooth iff* $A \in \#(\mathbf{E})$.

We will consider **E** and **SET** as full subcategories of $\bar{\mathbf{E}}$. These two subcategories are reflective and **SET** is also coreflective in the sense of [MacLane 71]. They do not share any objects except the terminal object. The natural number object (NNO) of $\bar{\mathbf{E}}$ is the set of actual natural numbers, i.e.

$$1 \xrightarrow{\Delta(0)} \Delta(N) \xrightarrow{\Delta(s)} \Delta(N).$$

Neither Δ nor # preserve the subobject classifier. For the clarity, we will write Ω_{II} for $\#(\Omega_{\mathrm{E}})$ and **bool** as $\Delta(\Omega_{\mathrm{SET}})$. The power object is not preserved, since the subobject classifier is not preserved. By Proposition 2 and Theorem 1, $P(\{A_i\}_{i \in I})$ is

$$\langle P(I), \{\Pi_{i \in p} P(A_i)\}_p \rangle,$$

so

$$P(\#(A)) = \langle \mathbf{bool}, \{P(A)_{\mathrm{tt}}, 1_{\mathrm{ff}}\} \rangle,$$

$$P(\Delta(I)) = \langle P(I), \{\Pi_{i \in p} \Omega_i\}_p \rangle,$$

where Ω_i is the subobject classifier of **E**. Since Δ and # preserves exponentials, we see

$$\Omega_{\mathrm{II}}{}^{A} \simeq \#(P(A)) \longrightarrow P(\#(A))$$

$$\mathbf{bool}^{I} \simeq \Delta(P(I)) \longrightarrow P(\Delta(I))$$

We will give a logical characterization of $\Omega_{\mathrm{II}}{}^{A}$ and $\mathbf{bool}^{I}$ in the next section.

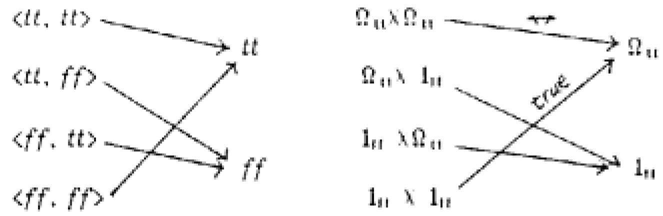
2. THE LOGIC OF ENVELOPE

In this section, we will examine the logic of $\bar{\mathbf{E}}$ and give a characterization of some objects of **E** and **SET** by the aid of the logic. Since $\bar{\mathbf{E}}$ is a topos, **E** obeys Heyting logic. We will give a characterization of logical symbols in $\bar{\mathbf{E}}$ by the aid of those in **SET** and **E**. We will follow [Fourman 77] to develop logic in a topos.

The equivalence $\longleftrightarrow$ is a morphism from $\Omega\times\Omega$ to Ω which is the classifier of the diagonal map

$$\Delta : \Omega \xrightarrow{\langle id,id\rangle} \Omega\times\Omega .$$

By the previous section, it is presented by

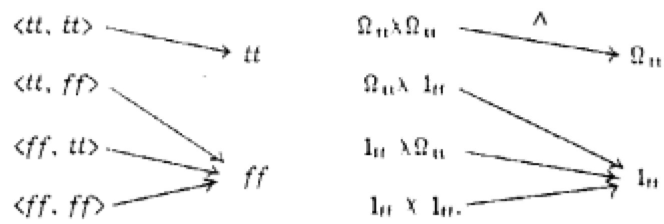


The left hand side of the above figure shows the base part $\nabla(\longleftrightarrow)$ and its arrows show the correspondence of elements, e.g. $\langle tt, tt\rangle$ is mapped to tt by $\nabla(\longleftrightarrow)$. On the other hand, the right hand side shows the fiber part $\longleftrightarrow^{*}$ and its arrows are morphisms in **E**, e.g. the top arrow is the logical equivalence in **E**.

Similarly, the conjunction $\wedge : \Omega\times\Omega \to \Omega$ which is the classifier of

$$1 \xrightarrow{\langle true,\ true\rangle} \Omega\times\Omega$$

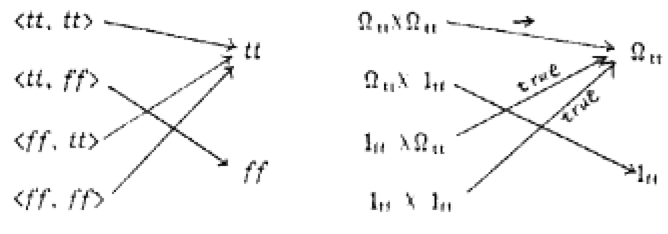
is presented by



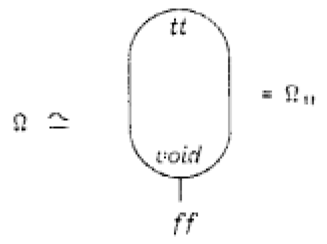
The implication is defined by

$$p \to q = (p \wedge q) \longleftrightarrow p,$$

so it is presented by



The internal poset Ω ordered by $p \leq q = p \to q$, looks as follows:



where *void* is the bottom element, i.e. *false* of Ω_{11}.

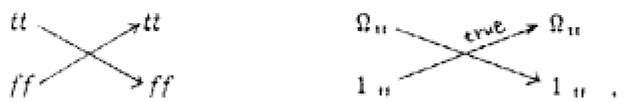
Define

$$\neg p = p \to ff,$$

$$\sim p = p \to void,$$

then

$\neg : \Omega \to \Omega$



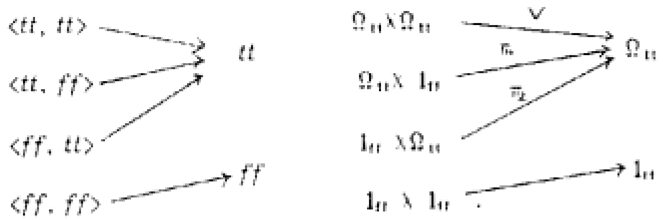
$\sim : \Omega \to \Omega$

Note that $\neg$ is the negation in $\bar{\mathbf{E}}$, on the contrary, $\sim$ is the negation in $\mathbf{E}$.

The disjunction $\vee$ is defined by

$$p \vee q = \forall r((p \to \wedge q \to r) \to r),$$

is presented by



Although we do not give the proof of this fact, it will turn out an easy exercise after the characterization of $\forall$ presented below.

Let $A=\{A_i\}_{i\in I}$ and $B=\{B_j\}_{j\in J}$ and

$$\phi : A\times B \xrightarrow{\langle f,\{\phi_{i,j}\}_{i,j}\rangle} \Omega.$$

The morphism ϕ may be thought as a binary predicate $\phi(a,b)$. Then a predicate $\forall b{:}B.\phi(a,b)$ is defined by

$$= \circ \langle \phi^{\wedge}, true_{A\times B}{}^{\wedge} : A \longrightarrow \Omega\rangle,$$

where $=$ is the equality of the type $P(B)$. By the previous section,

$$\forall b{:}B.\phi = \langle\{i\in I \mid \forall j.f(i,j)\},\ \{ = \circ \langle \pi_{j\in J}\phi_{i,j}{}^{\wedge},\ \Pi_{j\in J}true_{A_i\times B_j}{}^{\wedge}\rangle\}_i\rangle,$$

where $=$ is the equality of the type $\Pi_{j\in J}P(B_j)$. For any two families of morphisms $\{\phi_i{:}A\to B_i\}_{i\in I}$, $\{\psi_i{:}A\to B_i\}_{i\in I}$

$$= \circ \langle\langle\phi_i\rangle_{i\in I},\ \langle\psi_i\rangle_{i\in I}\rangle = \Lambda_I \circ \langle =_i \circ \langle\phi_i,\ \psi_i\rangle\rangle_{i\in I}$$

holds, where Λ_I is the classifier of

$$1 \xrightarrow{\langle true_i\rangle_{i\in I}} \Pi_{i\in I}\Omega_i.$$

We will write $\Lambda_{i\in I}\phi_i$ for $\Lambda_I \circ \langle\phi_i\rangle_{i\in I}$. Then we see

$$\forall b{:}B.\phi = \langle\{i \mid \forall j.f(i,j)\},\ \{\Lambda_{j\in J}\forall b{:}B_j.\phi_{i,j}(a,b)\}_i\rangle.$$

A predicate $\phi{:}A\to\Omega$ is said *smooth* if $\nabla(f)$ is the constant function tt. Let $X=\Delta(I)$, $Y=\Delta(J)$, $Z=\#(C)$ and a morphism

$$\phi{:}X\times Y\times Z \longrightarrow \Omega$$

be *smooth*. Then ϕ can be identified with the family of predicates

$$\phi^{\sim} = \{\phi_{i,j} : C \longrightarrow \Omega_{ij}\}_{i\in I, j\in J}.$$

Then the predicate

$$\forall x{:}X.\phi(x,y,z) : Y \times Z \longrightarrow \Omega$$

can be identified with

$$\{\bigwedge_{i\in I}\phi_{i,j} : C \longrightarrow \Omega_{ij}\}_{j\in J}.$$

Since $\bigwedge_{i\in I}$ is the infinitary conjunction of $\mathbf{E}$ in the sense of [Makkai & Reyes 77], this means the quantifier $\forall x{:}\Delta(I)$ of $\bar{\mathbf{E}}$ coincides with the infinitary conjunction in the usal categorical logic, provided that the predicate quantified by it is smooth. On the contrary, if $X=\#(A)$, $Y=\Delta(J)$ and $Z=\#(C)$, then the a smooth predicate $\phi{:}X\times Y\times Z \longrightarrow \Omega$ can be identified with

$$\{\phi_j : A\times C \longrightarrow \Omega_{ij}\}_{j\in J}.$$

Then $\forall x{:}X.\phi(x,y,z)$ can be identified with

$$\{\forall a{:}A.\phi_j : C \longrightarrow \Omega_{ij}\}_{j\in J}.$$

So the quantifier $\forall x{:}\#(A)$ of $\bar{\mathbf{E}}$ coincides with the quantifier $\forall a{:}A$ of $\mathbf{E}$, provided that the predicate quantified by it is smooth.

The existential quantifier defined by

$$\exists b{:}B.\phi \;=\; \forall p{:}\Omega(\forall(\phi(b) \rightarrow p) \rightarrow p)$$

is characterized by

$$\langle\{i\in I \mid \exists j.f(i,j)\},\ \{\bigvee_{j\in J}\exists b{:}B_j.\phi_{i,j}(a,b)\}_i\rangle$$

by the previous results. Similar to the universal quantifier, $\exists x{:}\Delta(I)$ of $\bar{\mathbf{E}}$ coincides with the infinitary disjunction $\bigvee_{i\in I}$ and $\exists x{:}\#(A)$ of $\bar{\mathbf{E}}$ coincides with $\exists x{:}A$ of $\mathbf{E}$, provided that the quantified predicate is smooth.

By the above results, we see the following preservation results.

PROPOSITION 6. *(i)* Δ *preserves*

$$\vee,\ \wedge,\ \longleftrightarrow,\ \rightarrow,\ \bigwedge,\ \bigvee,\ \mathit{true},\ \mathit{false},$$

(ii) $\#$ *preserves*

$$\vee,\ \wedge,\ \longleftrightarrow,\ \rightarrow,\ \bigwedge,\ \bigvee,\ \mathit{true}.$$

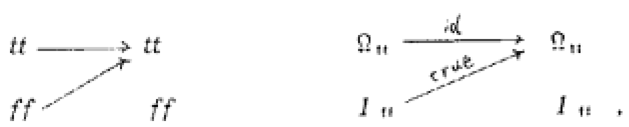
Note that # does not preserve *false* : $1 \to \Omega$, as $\#(false) \models void$. So $\#(\neg p)$ is $\sim\#(p)$. The following definition is useful to axiomatize the logic of **E**.

DEFINITION 4. *Let p be a variable of the type* Ω. *We define*

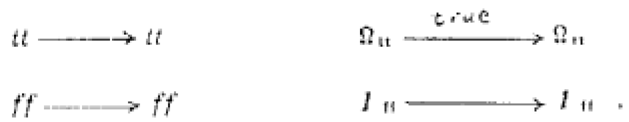
$$open(p) = void \to p, \quad bool(p) = p \vee \neg p$$

These predicates are characterized by the following figures:

bool : $\Omega \to \Omega$



open : $\Omega \to \Omega$



PROPOSTION 7. *In the envelope* **E**, *the following axioms hold*

(Axiom 1) $false \to p$,

(Axiom 2) $open(p) \vee \neg p$,

(Axiom 3) $\neg(open(p) \wedge \neg p)$.

From these axioms and the intuitionistic logic, we can derive the followings:

$$bool(true),\ bool(false),\ \sim false,\ open(p) \wedge bool(p) \to p,$$

$$\neg\neg void,\ open(void),\ open(p) \to (void \to p),$$

The following gives characterizations of some concepts in **E** by the aid of the above logic.

PROPOSITON 8. *(i) A predicate* $P(a): A \to \Omega$ *is smooth iff the following holds*

$$\forall a{:}A.open(P(a)).$$

A predicate $P(a):A\to\Omega$ is discreate, i.e. $P^{\#}$ is the family of morphisms which are constantly true, iff the following holds

$$\forall a{:}A.bool(P(a)).$$

Especially, the following hold

$$\Omega_{11} = \{p\in\Omega \mid open(p)\},$$

$$\mathbf{bool} = \{p\in\Omega \mid bool(p)\}.$$

(ii) Let $\{A_i\}_{i\in I}$ be a family of smooth objects. Then the smooth object $B=\amalg_i A_i$ and with canonical injection $f_i : A_i\to B$ in **E** *is characterized by the following (cf.* [Makkai & Reyes 77]):

$$f_i(a_i)=f_i(a'_i)\to a_i=a'_i,$$

$$AA_i(b)=AA_j(b)\to void \ for \ i\neq j,\ i,\ j\in I,$$

$$b=b\to \mathbf{V}_{i\in I}AA_i(b),$$

where $AA_i(b)$ stands for $\exists a_i.f_i(a_i)=b$.

(iii) Let A be a smooth object and I be a discreate object. Then we see

$$\Omega_{11}{}^{A} = \{p\in\Omega^{A} \mid \forall a{:}A.open(p(a))\},$$

$$\mathbf{bool}^{A} = \{p\in\Omega^{I} \mid \forall i{:}I.bool(p(i))\}.$$

Note that

$$\forall a(P(a)\vee\neg P(a))\to(Qa.P(a)\vee\neg Qa.P(a)),$$

holds, where Q is $\forall$ or $\exists$. This is not always true in a topos, even if the type of a is a constant sheaf. Similarly, in a topos, $P(a) : 1 \amalg 1\to\Omega$ may not be discreate in our sense, even if $P(a)$ is decidable. Namely, (i) of the above proposition is not always true for a topos.

We have not given a logical characterization of discreateness or smoothness of an object. It will need type destructors corresponding to $\nabla(-)$ and $(-)^{\flat}$.

We will show how to express Axiom" in the introduction by the aid of the logic of **E**. Set

$$D_N(N) = \langle N \times N,\ \{R^{D_k(n)}\}_{k \in N, n \in N} \rangle,$$

$$deg : D_N(N) \xrightarrow{\langle \pi_1,\ \{1_{k,n}\}_{k,n} \rangle} N,$$

$$varn : D_N(N) \xrightarrow{\langle \pi_2,\ \{1_{k,n}\}_{k,n} \rangle} N.$$

Then Axiom" is stated as

$$\forall k \forall n \forall f \in D_N(N)[deg(f)=k \wedge varn(f)=n \rightarrow \exists! p \in R(k,n)[f \text{ is given by } p]],$$

where k and n are variables of type N. Note that we have to construct a family of types of $\mathbf{E}$ $\{R^{D(n)}\}_{k \in N, n \in N}$, from which $D_N(N)$ is constructed. This would be performed in our proof checker by a type formation rule like the following

$$\frac{\begin{array}{c}[i \in I,\ I \in \mathbf{SET}] \\ A(i) \in \mathbf{E}\end{array}}{\coprod_{i \in I} A(i) \in \mathbf{E}.}$$

Besides, we need the introduction and elimination rules as [Martin-Löf 82]. This is closely related to the type destructors corresponding to $\nabla(-)$ and $(-)^{\wedge}$ mentioned above. These will be discussed in another paper.

REFERENCES

Ballantyne, A.M. & Bledsoe, W.W. [1977]: Automatic Proofs of Theorems in Analysis Using Non-standard Techniques, *JACM* **24**, 353-374

Furukawa, K. & Yokoi, T. [1984]: Basic Software System, in *Proceedings of the International Conference on Fifth Generation Computer Systems 1984*, 37-57, OHM North-Holland

Fourman, M.P. [1977]: The logic of topoi, in *Handbook of Mathematical Logic* (editor J. Barwise), 1053-1090, North-Holland, Amsterdam

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

Ketonen, J. & Weening, J.S. [1984]: *EKL - An Interactive Proof Checker User's Manual*, Stanford University

Ketonen, J. & Weening, J.S. [1984a]: *The Language of an Interactive Proof Checker*, Stanford University

MacLane, S. [1971]: *Categories for the Working Mathematicans*, Springer-Verlag, New York

Makkai, M. & Reyes, G.E. [1977]: *First Order Categorical Logic*, Lecture Notes in Mathematics 611, Springer-Verlag, Berlin

Martin-Löf, P. [1982]: Constructive mathematics and computer programming, in *Logic, methodology, and Philosophy of Science IV* (editor L.J. Cohen editors L.J. Cohen, J. Los, H. Pfeiffer and K.P. Podewski), 153-175, North-Holland, Amsterdam

APPENDIX

EKL is not appropriate as a proof checker of SDG, for its logic is classical but the principle of excluded middle leads a contradiction in SDG. However, this obstacle does not affect this experiments, for there is no essential use of logical operators in the following derivations.

```
(proof real)

(decl plus (type |(ground.ground.ground*):ground|)
      (syntype constant) (infixname +)
      (bindingpower 930) (associativity both))

(decl times (type |(ground.ground.ground*):ground|)
      (syntype constant) (infixname *)
      (bindingpower 935) (associativity both))

(decl (zero unit) (syntype constant) (type |ground|))

(decl (aa bb cc)  (type |ground|))

(axiom |all aa.( aa + zero = aa &  zero + aa = aa)|)
(label simpinfo)

(axiom |all aa.( aa * unit = aa &  unit * aa = aa)|)
(label simpinfo)

(decl (ff gg hh) (type |ground:ground|))

;simpfacts

(axiom |zero != unit|)
(label simpinfo)

;multiplication

(axiom |all aa. aa * zero = zero & zero * aa = zero|)
(label simpinfo)

(axiom |all aa. aa * unit = aa & unit * aa = aa|)
(label simpinfo)
```

```
(axiom |all aa bb. aa * bb = bb * aa|)
(label product_commute)(label commute)


;addition

(axiom |all aa. aa + zero = aa|)
(label simpinfo)

(axiom |all aa bb. aa + bb = bb + aa|)
(label plus_commute)(label commute)

(axiom |all aa bb cc.aa + bb = aa + cc iff bb = cc|)
(label simpinfo)(label plus_cancel)

(axiom |all aa bb cc.bb + aa = cc + aa iff bb = cc|)
(label simpinfo)(label plus_cancel)

;extensionality

(axiom |all f g.(all x.f(x) = g(x)) iff f = g|)
(label ext)


;distributivity
(axiom |all aa bb cc. aa * (bb + cc) = aa * bb + aa * cc|)
(label simpinfo)(label distfacts)

(axiom |all aa bb cc. (aa + bb) * cc = aa * cc + bb * cc|)
(label simpinfo)(label distfacts)

(decl nilpotent (type |ground:truthval|))

(decl (d d1 d2 d3) (type |ground|) (sort |nilpotent|))

(decl derivation
 (type |(ground:ground):(ground:ground)|) (postfixname |'|)
 (bindingpower 990))

(axiom |all f aa d. f(aa + d) = f(aa) + d * (f')(aa)|)
(label taylor)

(decl func_plus
       (type |((ground:ground),(ground:ground)):(ground:ground)|)
       (infixname ++) (bindingpower 930))
```

```
(define func_plus "all f1 f2. f1 ++ f2 = lambda aa. f1(aa) + f2(aa)")
(label func_plusdef)

(decl func_product
      (type :((ground:ground),(ground:ground)):(ground:ground):)
      (infixname **) (bindingpower 935))

(define func_product
        "all f1 f2. f1 ** f2 = lambda aa. f1(aa) * f2(aa)")
(label func_productdef)

(axiom "all d. d * d = zero")
(label simpinfo)

(axiom :all aa bb.(all d.d * aa = d * bb) iff aa = bb:)
(label simpinfo)(label taylor_unique)

(axiom :all aa bb.(all d.aa * d = bb * d) iff aa = bb:)
(label simpinfo)(label taylor_unique)

(proof diff_product)
::::::(f ** g)' = g ** f' ++ f ** g'

(trw :all d.(f ** g)(x+d) = f(x+d)*g(x+d): (open func_product))
:all d.(f ** g)(x+d) = f(x+d)*g(x+d)

(rw *  (use taylor mode exact) (use distfacts mode always)
       (use product_commute))
:all d.(f ** g)(x)+d*((f ** g)')(x) =
:       f(x)*g(x)+d*g(x)*(f')(x)+d*f(x)*(g')(x)

(rw * (part 1 (part 1 (part 1 (open func_product)))))
:all d.d*((f ** g)')(x) = d*g(x)*(f')(x)+d*f(x)*(g')(x)

(trw :all d.d*g(x)*(f')(x)+d*f(x)*(g')(x)
                               =  d*((g ** f' ++ f ** g')(x)):
  (open func_product) (open func_plus) (use distfacts))
:all d.d*g(x)*(f')(x)+d*f(x)*(g')(x) = d*(g ** f' ++ f ** g')(x)

(rw "-2" (use * mode exact))
:((f ** g)')(x) = (g ** f' ++ f ** g')(x)

(derive :all x.((f ** g)')(x) = (g ** f' ++ f ** g')(x): *)
```

```
(rw * (use ext))
;(f ** g)' = g ** f' ++ f ** g'

(proof diff_plus)
;;;;; (f ++ g)' = f' ++ g'

(trw :all d.(f ++ g)(x+d) = f(x+d)+g(x+d): (open func_plus))
:all d.(f ++ g)(x+d) = f(x+d)+g(x+d)

(rw * (use taylor mode exact) (use plus_commute))
:all d.(f ++ g)(x)+d*((f ++ g)')(x) = f(x)+g(x)+d*(f')(x)+d*(g')(x)

(rw * (part 1 (part 1 (part 1 (open func_plus)))))
:all d.d*((f ++ g)')(x) = d*(f')(x)+d*(g')(x)
(label tmp1)

(trw :all x d.d*(f')(x)+d*(g')(x) = d*(f' ++ g')(x):
      ((open func_plus) (use distfacts mode exact)))
:all x d.d*(f')(x)+d*(g')(x) = d*(f' ++ g')(x)
(label tmp2)

(rw tmp1 (use tmp2))
:((f ++ g)')(x) = (f' ++ g')(x)

(derive :all x.((f ++ g)')(x) = (f' ++ g')(x): *)

(rw * (use ext))
:(f ++ g)' = f' ++ g'
```