# Polymorphic Abstract Syntax via Grothendieck Construction

FoSSaCS'11, Saarbruecken, Germany

Gunma University, Japan **Makoto Hamana**

# Polymorphicな体系！

[Girard’71, Reynolds’74]

**System F**

**System Fω**

**λU**

[Girard’72]

**ML, Haskell**

[Milner’78]

**System Fc↑**

[Yorgey et al.’12]

**オブジェクト指向言語**

**Java, C++, Scala,etc.**

**Polymorphic Record Calculus**

[Ohori’95]

**Polymorphic π -calculus**

[Pierce, Sanngiorgi’00]

**Polymorphic Contracts**

[Guha, et al. ’07] [Belo et al.’11]

**Universe Polymorphism**

Agda

**Polymorphic Logic Programming**

[Shapiro’91][Hanus’91]

**Polymorphic XML**

[Hosoya et al.’09]

**⇒Polymorphic systemのための統一理論**

# My Research Programme: Algebraic Approach

$\Sigma$ $\rightsquigarrow$ $t : \tau$
signature

**Polymorphic Abstract Syntax**

e.g. System F syntax

$E \vdash s = t : \tau$
axioms

**Polymorphic Algebraic Theory**

e.g.
- System F axioms
- Haskell プログラム
- Polymorphic π

**Mechanized Formalization**

- Agda
- Coq
- Haskell

Presheaf category ≈ Dependent type theory

**General Notion of Models**

**Polymorphic Rewriting Theory**

**Operational Semantics**

# こ　の　発　表

- Grothendieck Construction とは何か
- どうして Polymorphic Syntax に役立つか

# I. Untyped Abstract Syntax with Binding

[Fiore,Plotkin,Turi. LICS99]

▷ Aim: model syntax with variable binding, e.g.

$$\frac{}{x_1,\ldots,x_n \vdash x_i} \qquad \frac{x_1,\ldots,x_n \vdash t \quad x_1,\ldots,x_n \vdash s}{x_1,\ldots,x_n \vdash t@s}$$

$$\frac{x_1,\ldots,x_n,x_{n+1} \vdash t}{x_1,\ldots,x_n \vdash \lambda(x_{n+1}.t)}$$

▷ Category $\mathbb{F}$ for contexts $\{x_1,\ldots,x_n\}$

objects: $n = \{1,\ldots,n\}$ (contexts)

arrows: all functions $n \to n'$ (renamings)

▷ Category $\mathbf{Set}^{\mathbb{F}} = \mathbb{F} \to \mathbf{Set}$ for modelling terms

$\Lambda \in \mathbf{Set}^{\mathbb{F}}$

$\Lambda(n) = \{\, t \mid n \vdash t \,\}$ $x_1,\ldots$

functor $F$ on $\mathbf{Set}^{\mathbb{F}}$ $\Rightarrow$ $\Lambda = \mu F \in \mathbf{Set}^{\mathbb{F}}$ is abstract syntax with binding

$$\boldsymbol{n} \qquad \vdash \quad \boldsymbol{t}$$

**context** **term**

- $\mathbf{Set}^{\mathbb{F}}$ suffices

# Problem

- ▷ Untyped [Fiore,Plotkin,Turi LICS 1999]

  $F \circlearrowright \mathbf{Set}^{\mathbb{F}} \quad \Rightarrow \quad \mu F \in \mathbf{Set}^{\mathbb{F}}$

  is abstract syntax with binding

- ▷ Polymorphic typed

  $F \circlearrowright \mathbf{Set}^{?} \quad \Rightarrow \quad \mu F \in \mathbf{Set}^{?}$

  is polymorphic abstract syntax

**⇒Find a suitable category for contexts**

## System F

*Types*

$$\tau ::= \alpha \mid b \mid \tau_1 \Rightarrow \tau_2 \mid \forall \alpha . \tau$$

$\alpha \cdots$ type variables $\quad b \cdots$ base types

*Well-formed types*

$$\frac{1 \leq i \leq n}{\alpha_1, \ldots, \alpha_n \vdash \alpha_i} \qquad \frac{}{\alpha_1, \ldots, \alpha_n \vdash b}$$

$$\frac{\alpha_1, \ldots, \alpha_n \vdash \sigma \quad \alpha_1, \ldots, \alpha_n \vdash \tau}{\alpha_1, \ldots, \alpha_n \vdash \sigma \Rightarrow \tau} \qquad \frac{\alpha_1, \ldots, \alpha_n, \alpha_{n+1} \vdash \tau}{\alpha_1, \ldots, \alpha_n \vdash \forall \alpha_{n+1} . \tau}$$

*Well-typed terms*

$$\frac{\Xi, \alpha \mid \Gamma \vdash t : \tau}{\Xi \mid \Gamma \vdash \Lambda \alpha . t : \forall \alpha . \tau} \qquad \frac{\Xi \mid \Gamma \vdash t : \forall \alpha . \tau \quad \Xi \vdash \sigma}{\Xi \mid \Gamma \vdash t\,\sigma : \tau[\alpha := \sigma]}$$

$\ldots$

Notes

- ▷ $\Xi = \alpha_1, \ldots, \alpha_n$ is a *type context*
- ▷ $\Gamma = x_1 : \tau_1, \ldots, x_k : \tau_k$ is a *term context*

$$n \mid \Gamma \vdash t : \tau$$

type context | term context | term | type

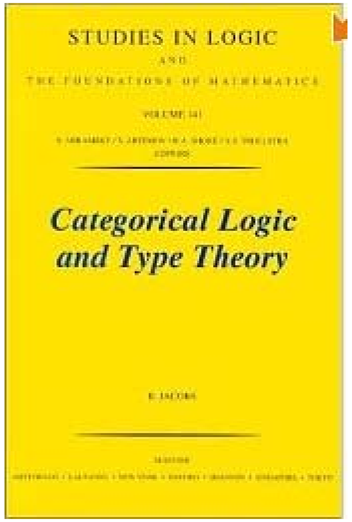
$$? \longrightarrow \mathbf{Set}$$

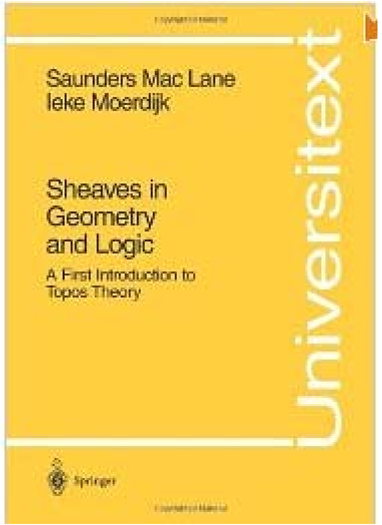
$$(n \mid \Gamma \vdash \tau) \longmapsto \{\, t \mid n \mid \Gamma \vdash t : \tau \,\}$$

⇒**Collect all possible (dependent) triples**

# The Grothendieck Construction

- プログラム意味論、型理論では比較的よく表れる




*Alexander Grothendieck*, 1928-

# The Grothendieck Construction

▷ A construction from

an Indexed category to a Category

$$\mathcal{A} : \mathcal{C} \to \mathbf{Cat} \quad \longmapsto \quad \int \mathcal{A} \quad \in \mathbf{Cat}$$

# Idea

- ▷ $A_1, A_2$ sets

  $A_1 \uplus A_2 = \{ (1, a) \mid a \in A_1 \} \cup \{ (2, a) \mid a \in A_2 \}$

- ▷ $A_1, A_2, \ldots \in \mathbf{Set} \quad \Leftrightarrow \quad A : \mathbb{N} \to \mathbf{Set}$

  $\biguplus_{i \in \mathbb{N}} A_i = \{ (i, a) \mid i \in \mathbb{N},\ a \in A_i \}$

- ▷ Given $\mathcal{A} : \mathcal{C} \to \mathbf{Cat}$

  **The Grothendieck Construction**

$$\int^{i \in \mathcal{C}} \mathcal{A}(i) \quad \in \ \mathbf{Cat}$$

  - ▶ objects $= \biguplus_{i \in \mathcal{C}} \mathcal{A}(i) = \{ (i, a) \mid i \in \mathcal{C},\ a \in \mathcal{A}(i) \}$
  - ▶ arrow: $(u, \gamma) : (i, a) \to (j, b)$ such that
    $u : i \to j$ and $\gamma : \mathcal{A}(u)(a) \to b$ in $\mathcal{A}(j)$

# The Category of Contexts with Result types

- ▷ To use the **Grothendieck construction**


$$\stackrel{def}{=} \int^{n \in \mathbb{F}} \mathbf{G}(n)$$

- ▷ where $\mathbf{G} : \mathbb{F} \longrightarrow \mathbf{Cat}$ is a functor

$$n \mid \Gamma \vdash t : \tau$$

$$\mathbf{G}(n) \stackrel{def}{=} \mathbf{Ctx}(n) \times \mathbb{T}(n)$$

$$n \mid \underset{\Gamma}{\uplus} \vdash \underset{\tau}{\uplus}$$

# Arrows in $\int G$ give a Right Notion of Renaming

$$
\begin{array}{l|ccccc}
 & n & \mid \; \Gamma & \vdash & t & : \; \tau \\
\hline
\text{renaming } \rho \curvearrowright & m & \mid \; \rho(\Gamma) & \vdash & \rho(t) & : \; \rho(\tau) \\
 & & \text{renaming} \downarrow \pi & & \downarrow & \quad \| \\
\hline
 & m & \mid \; \Delta & \vdash & \pi(\rho(t)) & : \; \sigma
\end{array}
$$

- ▷ Category $\int^{n \in \mathbb{F}} \mathbf{G}(n)$

  objects: $(n \mid \Gamma \vdash \tau)$

  arrows: $(\rho, \pi) : (n \mid \Gamma \vdash \tau) \to (m \mid \Delta \vdash \sigma)$

  . s.t. $\rho : n \to m \qquad \pi : \rho\Gamma \to \Delta$

  & $\rho(\tau) = \sigma$

# Summary

- ▷ Untyped [Fiore,Plotkin,Turi LICS 1999]

$$F \circlearrowright \mathbf{Set}^{\mathbb{F}} \quad \Rightarrow \quad \mu F \;\in\; \mathbf{Set}^{\mathbb{F}}$$

is abstract syntax with binding

- ▷ Polymorphic typed [H. FoSSaCS 2011]

$$F \circlearrowright \mathbf{Set}^{\int \mathbf{G}} \quad \Rightarrow \quad \mu F \;\in\; \mathbf{Set}^{\int \mathbf{G}}$$

is polymorphic abstract syntax

  - ▶ where $\mathbf{G}(n) = \quad \mathbf{Ctx}(n) \times \mathbb{T}(n)$.

# Summary

▷ Untyped [Fiore,Plotkin,Turi LICS 1999]

$$\boldsymbol{F} \circlearrowright \mathbf{Set}^{\mathbb{F}} \quad \Rightarrow \quad \mu \boldsymbol{F} \in \mathbf{Set}^{\mathbb{F}}$$

is abstract syntax with binding

▷ Polymorphic typed [H. FoSSaCS 2011]

$$\boldsymbol{F} \circlearrowright \mathbf{Set}^{\int \mathbf{G}} \quad \Rightarrow \quad \mu \boldsymbol{F} \in \mathbf{Set}^{\int \mathbf{G}}$$

is polymorphic abstract syntax

- where $\mathbf{G}(\boldsymbol{n}) = \mathbf{Ctx}(\boldsymbol{n}) \times \mathbb{T}(\boldsymbol{n})$.
- This has not been known for **12 years!**
- The Grothendieck construction $\int$ is the key.

# My Research Programme: Algebraic Approach