Algebraic Semantics of Higher-Order Abstract Syntax and Second-Order Rewriting

Makoto Hamana

Department of Computer Science,
Gunma University, Japan

International School on Rewriting (ISR 2015),
August 10-14, 2015, Leipzig, Germany

This Lecture

I. Higher-Order Abstract Syntax

1. What is HOAS
2. Algebraic semantics
3. AIM: Meta-terms form a free $\boldsymbol{\Sigma}$-monoid

II. Second-Order Rewrite System

1. Definition of CRS
2. Algebraic interpretation
3. AIM: CRS rewriting forms a monotone $(\boldsymbol{\Sigma}, \boldsymbol{\mathcal{R}})$-algebra
4. Termination by interpretation

We begin with a review on TRS.

TRS: Review

Term Rewriting System (TRS) $\mathcal{R}$:

$$\begin{aligned} fact(0) &\to S(0) \\ fact(S(x)) &\to fact(x) * S(x) \end{aligned}$$

Terms are defined by

$$T_\Sigma(X) \ni \quad t \; ::= \; x \mid f(t_1, \ldots, t_m)$$

The rewrite relation

$$\to_\mathcal{R} \quad \subseteq \quad T_\Sigma(X) \times T_\Sigma(X)$$

is defined by

$$\frac{l \to r \in \mathcal{R}}{l\theta \to_\mathcal{R} r\theta} \qquad \frac{s \to_\mathcal{R} t}{f(\ldots, s, \ldots) \to_\mathcal{R} f(\ldots, t, \ldots)}$$

where θ is a substitution $\theta : X \to T_\Sigma(X)$

Problem of TRS

- ▷ TRS doesn't involve $\boldsymbol{\lambda}$-terms
- ▷ It is problematic in applying it to the theory of programming languages

Combinatory Reduction Systems CRS [Klop'80]

– A second-order extension of TRS

Example of CRS (1): Prenex normal forms

$$\mathrm{P} \wedge \forall(\boldsymbol{x}.\mathrm{Q}[\boldsymbol{x}]) \longrightarrow \forall(\boldsymbol{x}.\mathrm{P} \wedge \mathrm{Q}[\boldsymbol{x}]) \qquad \neg\forall(\boldsymbol{x}.\mathrm{Q}[\boldsymbol{x}]) \longrightarrow \exists(\boldsymbol{x}.\neg(\mathrm{Q}[\boldsymbol{x}]))$$

$$\forall(\boldsymbol{x}.\mathrm{Q}[\boldsymbol{x}]) \wedge \mathrm{P} \longrightarrow \forall(\boldsymbol{x}.\mathrm{Q}[\boldsymbol{x}] \wedge \mathrm{P}) \qquad \neg\exists(\boldsymbol{x}.\mathrm{Q}[\boldsymbol{x}]) \longrightarrow \forall(\boldsymbol{x}.\neg(\mathrm{Q}[\boldsymbol{x}]))$$

CRS (Combinatory Reduction System):

▷ Variable binders

▷ Metavariables

▷ Substitutions (Metavars, object vars)

Aim of This Lecture

Example of CRS (2): the λ-calculus

$$\mathbf{app}(\boldsymbol{\lambda}(\boldsymbol{x}.\mathrm{M}[\boldsymbol{x}]),\ \mathrm{N}) \longrightarrow \mathrm{M}[\mathrm{N}]$$

$$\boldsymbol{\lambda}(\boldsymbol{x}.\mathbf{app}(\mathrm{M},\ \boldsymbol{x})) \longrightarrow \mathrm{M}$$

▷ The syntax is similar to the usual meta-language in CS

▷ Aim: Algebraic semantics of CRS

Theory of TRS	⇝	Theory of CRS
algebraic models exist		what are algebraic models?

▷ Goals: CRS's

(1) Complete algebraic models

(2) Proof method of termination based on models

What is Higher-Order Abstract Syntax

▷ Abstract Syntax with variable binding

$$\lambda x.xx \qquad \forall x.\mathbf{bird}(x) \longrightarrow \mathbf{fly}(x)$$

▶ Not merely a context free grammar

▷ with Metavariables

$$\lambda x.\mathrm{M}[x] \qquad \forall x.\mathrm{P}[x] \longrightarrow \mathrm{Q}[x]$$

▷ Substitution for metavariables

$$(\lambda x.\mathrm{M}[x])\{\mathrm{M}^{(1)} \mapsto yy\} = \lambda x.xx$$

$$(\lambda x.\mathrm{M}[x])\{\mathrm{M}^{(1)} \mapsto \mathbf{fix}(y)\} = \lambda x.\mathbf{fix}(x)$$

- Meta-terms
  - Aczel. *A general Church-Rosser theorem*, unpublished, University of Manchester, 1978.
  - Aczel. *Frege structures* [Aczel'80]
  - Klop. *Combinatory Reduction Systems*, CWI, volume 127 of Mathematical Centre Tracts, 1980.

Higher-Order Abstract Syntax (without metavars)

▷ Formalisation – Haskell

```
data Lam :: * -> * where
  Var :: a -> Lam a
  App :: Lam a -> Lam a -> Lam a
  Abs :: Lam (Maybe a) -> Lam a
```

```
data Maybe a = Nothing
             | Just a
```

▷ Formalisation – Agda

```
data Lam : Nat -> Set where
  Var : Fin n -> Lam n
  App : Lam n -> Lam n -> Lam n
  Abs : Lam (S n) -> Lam n
```

```
data Fin : Nat -> Set where
  Zero : Fin (S n)
  S    : Fin n -> Fin (S n)
```

cf. Hamana, Fiore. *A Foundation for GADTs and Inductive Families: Dependent Polynomial Functor Approach*. ACM WGP'11.

$\boldsymbol{\lambda}$-terms in de Bruijn Levels

- Different from de Bruijn indices
- Variables are numbers starting from 1, counted from the top
  Examples

$$\boldsymbol{\lambda x.\lambda y.yx} \qquad \boldsymbol{\lambda 1.\lambda 2.21}$$

$$\boldsymbol{(\lambda x.xx)(\lambda x.xx)} \qquad \boldsymbol{(\lambda 1.11)(\lambda 1.11)}$$

$$\boldsymbol{\lambda x.xy} \qquad \boldsymbol{\lambda 2.21}$$

- Terms in contexts
- Free variables are "external" binders

$\boldsymbol{\lambda}$-terms in de Bruijn Levels: Constructions

- Normal way: terms-in-contexts

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

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

- Construction rules in de Bruijn levels

$$\frac{}{1,\ldots,n \vdash i} \qquad \frac{1,\ldots,n \vdash t \quad 1,\ldots,n \vdash s}{1,\ldots,n \vdash t@s}$$

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

$\boldsymbol{\lambda}$-terms in de Bruijn Levels: Constructions

▷ We will simply write ($\boldsymbol{n} = \{\mathbf{1}, \ldots, \boldsymbol{n}\}$)

$$\boldsymbol{n} \vdash \boldsymbol{\lambda}(\boldsymbol{n}+\mathbf{1}.\boldsymbol{t})$$

▷ Construction rules in de Bruijn levels

$$\frac{}{\boldsymbol{n} \vdash \boldsymbol{i}} \qquad \frac{\boldsymbol{n} \vdash \boldsymbol{t} \quad \boldsymbol{n} \vdash \boldsymbol{s}}{\boldsymbol{n} \vdash \boldsymbol{t}@\boldsymbol{s}}$$

$$\frac{\boldsymbol{n}+\mathbf{1} \vdash \boldsymbol{t}}{\boldsymbol{n} \vdash \boldsymbol{\lambda}(\boldsymbol{n}+\mathbf{1}.\boldsymbol{t})}$$

Free Σ-monoids: HO Syntax with Metavariables [Fiore,Plotkin,Turi'99][Hamana APLAS'04]

- A Σ-monoid (A, α, ν, μ) is
  - a Σ-algebra (A, α) in the category $\mathbf{Set}^{\mathbb{F}}$ with
  - a monoid structure in $\mathbf{Set}^{\mathbb{F}}$ using ν, μ
  - both are compatible.
- Idea
  - Unit ν models the variable former
  - Multiplication μ models substitution for variables

★

Very Basic Category Theory

- Categories: definition, **Set**, $\mathbb{F}$
- Functors
- Natural transformations
- Functor category $\mathbf{Set}^{\mathbb{F}}$
  (Presheaf category)

Very Basic Category Theory (1)

▷ A category $\mathcal{C}$ consists of

– A set of objects $\mathbf{ob}\ \mathcal{C}$

. $A, B, C, \ldots$

– A set of arrows $\mathcal{C}(A, B) = \{f \mid f : A \longrightarrow B\}$

. for every $A, B \in \mathbf{ob}\ \mathcal{C}$

– Identities $\mathbf{id}_A : A \longrightarrow A$ for every $A \in \mathbf{ob}\ \mathcal{C}$

– Composition $g \circ f : A \longrightarrow C$

. for $f : A \longrightarrow B, \quad g : B \longrightarrow C$

satisfying

i) $f \circ \mathbf{id} = f$

ii) $\mathbf{id} \circ f = f$

iii) $f \circ (g \circ h) = (f \circ g) \circ h$.

Examples of Categories

Category **Set**

- ▷ Objects: (small) sets $\boldsymbol{S}$
- ▷ Arrows: functions $\boldsymbol{f : S \longrightarrow S'}$

Category $\mathbb{F} \cdots$ category of finite (skeltal) sets

- ▷ Objects: finite cardinals $\boldsymbol{n = \{1, \ldots, n\}}$
- ▷ Arrows: functions $\boldsymbol{\rho : m \longrightarrow n}$

Examples of Categories

Category **Mon** $\cdots$ category of monoids

- ▷ Objects: monoids $M = (M, \cdot, e)$
- ▷ Arrows: monoid homomorphisms $h : M \longrightarrow M'$

Category Σ**-alg** $\cdots$ category of Σ-algebras

- ▷ Objects: Σ-algebras $A = (A, \{f^A\}_{f \in \Sigma})$
- ▷ Arrows: homomorphisms $h : A \longrightarrow A'$

Very Basic Category Theory (2)

- Let $\mathcal{C}, \mathcal{D}$ be categories.
- A functor $F : \mathcal{C} \longrightarrow \mathcal{D}$ is given by (F_0, F_1)
  - i) the object part function $F_0 : \mathbf{ob}\ \mathcal{C} \longrightarrow \mathbf{ob}\ \mathcal{D}$
  - ii) the arrow part function $F_1 : \mathcal{C}(A, B) \longrightarrow \mathcal{D}(F_0(A), F_0(B))$

  satisfying

  - i) $F_1(\mathrm{id}_A) = \mathrm{id}_{F_0(A)}$
  - ii) $F_1(g \circ f) = F_1(g) \circ F_1(f)$

  Note: F_0, F_1 will be denoted simply by F

Important Examples of Categories

Category $\mathbf{Set}^{\mathbb{F}}$ (functor category, presheaf category)

- Object: functor $A : \mathbb{F} \longrightarrow \mathbf{Set}$
  (called presheaf, and written $A \in \mathbf{Set}^{\mathbb{F}}$)

- Arrow: natural transformation $f : A \longrightarrow B$

which is a family of functions (“components”)

$$\{\ f(n) : A(n) \longrightarrow B(n)\ \}_{n \in \mathbb{N}}$$

that satisfies the condition

$$\forall a \in A(m).\ B(\rho)(f(m)(a)) = f(n)(A(\rho)(a))$$

for all functions $\rho : m \longrightarrow n$.

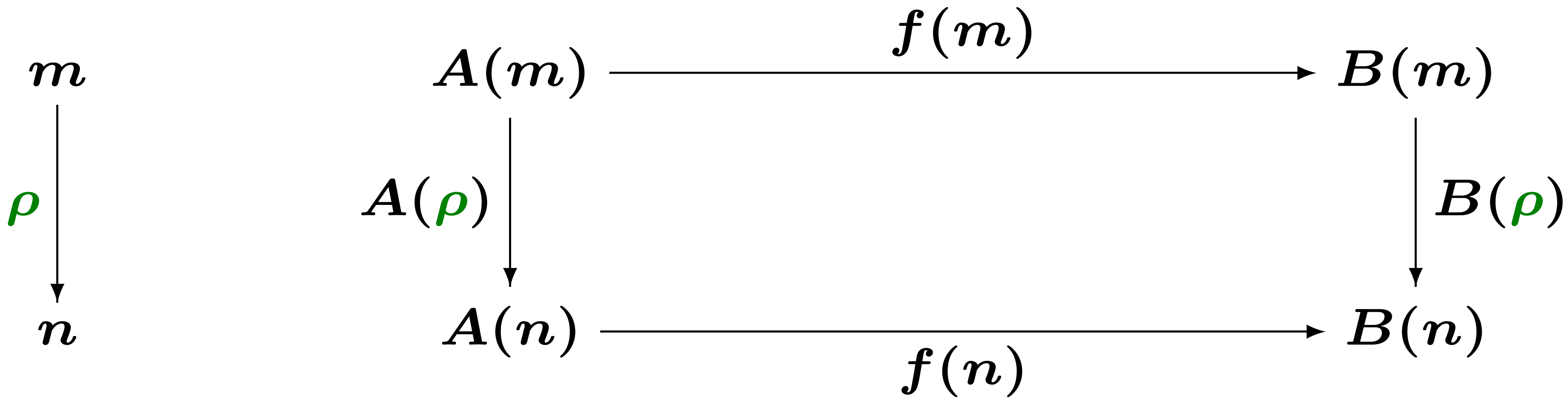
Idea: f is a polymorphic function w.r.t. n

An arrow of $\mathbf{Set}^{\mathbb{F}}$

This condition "naturality"

$$\forall a \in A(m).\ B(\rho)(f(m)(a)) = f(n)(A(\rho)(a))$$

diagrammatically means the commutativity of the diagram



Idea: f is a polymorphic function w.r.t. n

I. Abstract Syntax and Variable Binding

[Fiore,Plotkin,Turi LICS'99]

- Aim: To model syntax with variable binding, e.g.

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

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

- Syntax generated by 3 constructors
- λ is a special unary function symbol: it decreases the context

Abstract Syntax and Variable Binding

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

$$\frac{}{\boldsymbol{n} \vdash \boldsymbol{i}} \qquad \frac{\boldsymbol{n} \vdash \boldsymbol{t} \quad \boldsymbol{n} \vdash \boldsymbol{s}}{\boldsymbol{n} \vdash \boldsymbol{t@s}}$$

$$\frac{\boldsymbol{n+1} \vdash \boldsymbol{t}}{\boldsymbol{n} \vdash \boldsymbol{\lambda}(\boldsymbol{n+1.t})}$$

▷ Category $\mathbb{F}$ for variable contexts

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

arrows: all functions $\boldsymbol{n} \longrightarrow \boldsymbol{n'}$ (renamings)

▷ Context extension: $\boldsymbol{\delta A} \in \mathbf{Set}^{\mathbb{F}}$; $(\boldsymbol{\delta A})(\boldsymbol{n}) = \boldsymbol{A}(\boldsymbol{n}+\mathbf{1})$

The presheaf of variables: $\mathbf{V} \in \mathbf{Set}^{\mathbb{F}}$; $\mathbf{V}(\boldsymbol{n}) = \{\mathbf{1}, \ldots, \boldsymbol{n}\}$

*

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

Def. A binding signature Σ consists of a set Σ of function symbols with binding arities.

A function symbol with binding arity $(n_i \in \mathbb{N})$

$$f : \langle n_1, \ldots, n_l \rangle$$

has l arguments and binds n_i variables in the i-th argument .

Def. A Σ-algebra $A = (A, \{f^A\}_{f \in \Sigma})$ in $\mathbf{Set}^{\mathbb{F}}$ consists of

- a presheaf $A \in \mathbf{Set}^{\mathbb{F}}$ (carrier)
- an arrow of $\mathbf{Set}^{\mathbb{F}}$ (operation)

$$f^A : \delta^{n_1} A \times \ldots \times \delta^{n_l} A \longrightarrow A$$

corresponding to each function symbol $f : \langle n_1, \ldots, n_l \rangle \in \Sigma$.

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

Def. A binding signature Σ consists of a set Σ of function symbols with binding arities.

A function symbol with binding arity

$$f : \langle n_1, \ldots, n_l \rangle$$

has l arguments and binds n_i variables in the i-th argument .

Def. A $\mathbf{V}+\Sigma$-algebra $A = (A, \{f^A\}_{f \in \Sigma})$ consists of

- ▷ a presheaf $A \in \mathbf{Set}^{\mathbb{F}}$ (carrier)
- ▷ an arrow of $\mathbf{Set}^{\mathbb{F}}$ (operation)

$$f^A : \delta^{n_1} A \times \ldots \times \delta^{n_l} A \longrightarrow A$$

corresponding to each function symbol $f : \langle n_1, \ldots, n_l \rangle \in \Sigma$

- ▷ an arrow of $\mathbf{Set}^{\mathbb{F}}$ (unit)

. $\quad \nu : \mathbf{V} \longrightarrow A$

Σ-monoids – Algebras with Substitutions

Def. A Σ-monoid $A = (A, \nu, \mu)$ is a Σ-algebra A equipped with

- an arrow $\nu : \mathrm{V} \longrightarrow A$ of $\mathbf{Set}^{\mathbb{F}}$ (unit)
- a family of functions (multiplication),

$$\mu_n^{(m)} : A(m) \times A(n)^m \longrightarrow A(n)$$

which is (extra)natural and satisfies the Σ-monoid laws

$$\begin{array}{rll}
\text{(i)} & \mu_n(\nu_m(i); \overrightarrow{b}) & = b_i \\
\text{(ii)} & \mu_n^{(n)}(a; \nu_n(1), \ldots, \nu_n(n)) & = a \\
\text{(iii)} & \mu_n(a; \mu_n(b_1, \overrightarrow{c}), \ldots, \mu_n(b_l, \overrightarrow{c})) & = \mu_n(\mu_n(a; \overrightarrow{b}); \overrightarrow{c}) \\
\text{(vi)} & \mu_n(f_n^A(a_1, \ldots, a_l); \overrightarrow{b}) & = \\
f_n^A(& \mu_n(a_1; \overrightarrow{b}, (\nu_{m+k_i}(m+i))_{i=1,\ldots,k_1}), \ldots, & \\
 & \mu_n(a_l; \overrightarrow{b}, (\nu_{m+k_l}(m+i))_{i=1,\ldots,k_l})) &
\end{array}$$

for $f : \langle k_1, \ldots, k_l \rangle \in \Sigma$

Σ-monoids – Algebras with Substitutions

Def. A Σ-monoid $A = (A, \nu, \mu)$ is a Σ-algebra A equipped with

▷ an arrow $\nu : \mathbf{V} \longrightarrow A$ of $\mathbf{Set}^{\mathbb{F}}$ (unit)

▷ a family of functions (multiplication),

$$\mu_n^{(m)} : A(m) \times A(n)^m \longrightarrow A(n)$$

which is (extra)natural and satisfies the Σ-monoid laws

$$\begin{aligned}
&\text{(i)} && \mu_n(\nu_m(i); \overrightarrow{b}) && = b_i \\
&\text{(ii)} && \mu_n^{(n)}(a; \nu_n(1), \ldots, \nu_n(n)) && = a \\
&\text{(iii)} && \mu_n(a; \mu_n(b_1, \overrightarrow{c}), \ldots, \mu_n(b_l, \overrightarrow{c})) && = \mu_n(\mu_n(a; \overrightarrow{b}); \overrightarrow{c}) \\
&\text{(vi)} && \mu_n(f_n^A(a_1, \ldots, a_l); \overrightarrow{b}) && \\
& && \quad = f_n^A(\mu_n(a_1; \overrightarrow{b}), \ldots, \mu_n(a_l; \overrightarrow{b})) && \text{NB. roughly!}
\end{aligned}$$

– compatibility between Σ-algebra and multiplication

Example: $\boldsymbol{\lambda}$-terms

▷ Binding signature $\boldsymbol{\Sigma}_{\boldsymbol{\lambda}}$ for $\boldsymbol{\lambda}$-terms

$$\boldsymbol{\lambda} \;:\; \langle \mathbf{1} \rangle, \qquad\qquad @ \;:\; \langle \mathbf{0}, \mathbf{0} \rangle$$

▷ Carrier: the presheaf $\boldsymbol{\Lambda}$ of all $\boldsymbol{\lambda}$-terms

$$\boldsymbol{\Lambda}(\boldsymbol{n}) = \{\boldsymbol{t} \mid \boldsymbol{n} \vdash \boldsymbol{t}\}$$

$$\boldsymbol{\Lambda}(\boldsymbol{\rho}) : \boldsymbol{\Lambda}(\boldsymbol{m}) \longrightarrow \boldsymbol{\Lambda}(\boldsymbol{n}) \qquad \text{renaming on } \boldsymbol{\lambda}\text{-terms}$$

for $\boldsymbol{\rho} : \boldsymbol{m} \longrightarrow \boldsymbol{n}$ in $\mathbb{F}$.

▷ Forms a $\boldsymbol{\Sigma}_{\boldsymbol{\lambda}}$-algebra

$$\begin{array}{lcl} @^{\boldsymbol{\Lambda}} : \boldsymbol{\Lambda} \times \boldsymbol{\Lambda} & \longrightarrow & \boldsymbol{\Lambda} \\ @^{\boldsymbol{\Lambda}}_{\boldsymbol{n}}(\boldsymbol{s}, \boldsymbol{t}) & = & \boldsymbol{s} @ \boldsymbol{t} \end{array} \qquad \begin{array}{lcl} \boldsymbol{\lambda}^{\boldsymbol{\Lambda}} : \boldsymbol{\delta}\boldsymbol{\Lambda} & \longrightarrow & \boldsymbol{\Lambda} \\ \boldsymbol{\lambda}^{\boldsymbol{\Lambda}}_{\boldsymbol{n}}(\boldsymbol{t}) & = & \boldsymbol{\lambda}(\boldsymbol{n} + \mathbf{1}.\boldsymbol{t}) \end{array}$$

Example: λ-terms

- Binding signature Σ_λ for λ-terms

$$\lambda \; : \langle 1 \rangle, \qquad @ \; : \langle 0, 0 \rangle$$

- Carrier: the presheaf Λ of all λ-terms

$$\Lambda(n) = \{ t \mid n \vdash t \}$$
$$\Lambda(\rho) : \Lambda(m) \longrightarrow \Lambda(n) \qquad \text{renaming on } \lambda\text{-terms}$$

for $\rho : m \longrightarrow n$ in $\mathbb{F}$.

- Forms a $\mathbf{V} + \Sigma_\lambda$-algebra

$$\begin{array}{lll} \nu^\Lambda : \mathbf{V} \longrightarrow \Lambda & @^\Lambda : \Lambda \times \Lambda \longrightarrow \Lambda & \lambda^\Lambda : \delta\Lambda \longrightarrow \Lambda \\ \nu_n(i) = i & @^\Lambda_n(s, t) = s@t & \lambda^\Lambda_n(t) = \lambda(n + 1.t) \end{array}$$

Example: $\boldsymbol{\lambda}$-terms – Summary (1)

- $\boldsymbol{V} + \boldsymbol{\Sigma}_{\boldsymbol{\lambda}}$-algebra $(\boldsymbol{\Lambda} \in \mathbf{Set}^{\mathbb{F}},\ \{\mathbf{var}, \mathbf{app}, \mathbf{abs}\})$

$$\begin{array}{llll} \mathbf{var} : \mathrm{V} \longrightarrow \boldsymbol{\Lambda} & \mathbf{app} : \boldsymbol{\Lambda} \times \boldsymbol{\Lambda} \longrightarrow \boldsymbol{\Lambda} & \mathbf{abs} & : \delta\boldsymbol{\Lambda} \longrightarrow \boldsymbol{\Lambda} \\ \quad\ \ \boldsymbol{i} \ \longmapsto \boldsymbol{i} & \qquad\quad \boldsymbol{s}\ ,\ \ \boldsymbol{t} \ \longmapsto \boldsymbol{s}@\boldsymbol{t} & \mathbf{abs}(\boldsymbol{n}) & : \boldsymbol{\Lambda}(\boldsymbol{n}+1) \longrightarrow \boldsymbol{\Lambda}(\boldsymbol{n}) \\ & & & \quad \boldsymbol{t} \qquad\qquad \longmapsto \boldsymbol{\lambda n}{+}\mathbf{1}.\boldsymbol{t} \end{array}$$

- Unit $=$ variables
- Multiplication $=$ substitution for variables
- Why presheaves? e.g. $\boldsymbol{\Lambda} \in \mathbf{Set}^{\mathbb{F}}$
- A judgment $\boldsymbol{n} \vdash \boldsymbol{t}$ is modelled as $\boldsymbol{t} \in \boldsymbol{\Lambda}(\boldsymbol{n})$
- Presheaf action $\boldsymbol{\Lambda}(\boldsymbol{\rho}) : \boldsymbol{\Lambda}(\boldsymbol{n}) \longrightarrow \boldsymbol{\Lambda}(\boldsymbol{n}')$
  is renaming of free variables $\boldsymbol{\rho} : \boldsymbol{n} \longrightarrow \boldsymbol{n}'$ in a $\boldsymbol{\lambda}$-term
- Thm. $\boldsymbol{\Lambda}$ is an initial $\mathbf{V} + \boldsymbol{\Sigma}_{\boldsymbol{\lambda}}$-algebra

Example: λ-terms – Summary (2)

Forms a Σ_λ-monoid (Λ, ν, μ)

▷ Multiplication $\mu_n^{(m)} : \Lambda(m) \times \Lambda(n)^m \longrightarrow \Lambda(n)$

. $\mu_n(t, \overrightarrow{s}) \overset{def}{=} t[1 := s_1, \ldots, n := s_m]$

(the substitution of λ-terms for de Bruijn variables) satisfying:

$$
\begin{array}{lll}
\text{(i)} & \mu_n(i;\ \overrightarrow{s}) = s_i & (i \in n) \\
\text{(ii)} & \mu_n^{(n)}(t; 1, \ldots, n) = t & \\
\text{(iii)} & \mu_n(t; \mu_n(s_1, \overrightarrow{u}), \ldots, \mu_n(s_l, \overrightarrow{u})) = \mu_n(\mu_n(t; \overrightarrow{b}); \overrightarrow{c}) & \\
\text{(vi-@)} & \mu_n(t_1 @\, t_2;\ \overrightarrow{s}) = \mu_n(t_1; \overrightarrow{s}) \;@\; \mu_n(t_2; \overrightarrow{s}) & \\
\text{(vi-}\lambda\text{)} & \mu_n(\lambda(n+1.\, t);\ \overrightarrow{s}) = \lambda(n+1.\, \mu_{n+1}(t;\ \overrightarrow{s}, n+1)) &
\end{array}
$$

▷ The above axioms of Σ_λ-monoid are satisfied, because μ is substitution.

▷ Thm. (Λ, ν, μ) is an initial Σ_λ-monoid

★

Example of $\mathbf{V}+\boldsymbol{\Sigma}$-algebra (1): Polynomial Interpretation

- $\forall(\boldsymbol{y}.\exists(\boldsymbol{x}.(\neg\mathbf{p}(\boldsymbol{x}) \vee \mathbf{q}(\boldsymbol{y})))).$
- Binding signature $\boldsymbol{\Sigma}$

$$\forall, \exists : \langle 1 \rangle \quad \wedge, \vee : \langle 0, 0 \rangle \quad \neg, \mathbf{p}, \mathbf{q} : \langle 0 \rangle$$

- $\mathbf{V}+\boldsymbol{\Sigma}$-algebra $(\boldsymbol{K}_{\mathbb{N}}, >_{\boldsymbol{K}_{\mathbb{N}}})$ $\quad \boldsymbol{K}_{\mathbb{N}} \in \mathbf{Set}^{\mathbb{F}}$
  - carrier $\boldsymbol{K}_{\mathbb{N}}(\boldsymbol{n}) = \mathbb{N}$ $\quad$ for "the sizes of formulae"
  - operations $\quad \boldsymbol{\nu}_{\boldsymbol{n}}^{\boldsymbol{K}}(\boldsymbol{x}) = \mathbf{1}$

$$\wedge_{\boldsymbol{n}}^{\boldsymbol{K}_{\mathbb{N}}}(\boldsymbol{x}, \boldsymbol{y}) = \vee_{\boldsymbol{n}}^{\boldsymbol{K}_{\mathbb{N}}}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x} + \boldsymbol{y} \quad \forall_{\boldsymbol{n}}^{\boldsymbol{K}_{\mathbb{N}}}(\boldsymbol{x}) = \exists_{\boldsymbol{n}}^{\boldsymbol{K}_{\mathbb{N}}}(\boldsymbol{x}) = \boldsymbol{x} + \mathbf{1}$$

$$\neg_{\boldsymbol{n}}^{\boldsymbol{K}_{\mathbb{N}}}(\boldsymbol{x}) = \mathbf{p}_{\boldsymbol{n}}^{\boldsymbol{K}_{\mathbb{N}}} = \mathbf{q}_{\boldsymbol{n}}^{\boldsymbol{K}_{\mathbb{N}}} = \boldsymbol{x} + \mathbf{1}$$

- Difficult to be a $\boldsymbol{\Sigma}$-monoid (what is multiplication?)
- Basically first-order algebra

Example (2): Second-Order Polynomial Interpretation

$$\mathsf{map}(x.\mathrm{F}[x], \mathsf{nil}) \longrightarrow \mathsf{nil}$$

$$\mathsf{map}(x.\mathrm{F}[x], \mathsf{cons}(\mathrm{X}, \mathrm{Y})) \longrightarrow \mathsf{cons}(\mathrm{F}[\mathrm{X}], \mathsf{map}(\mathrm{F}, \mathrm{Y}))$$

- Binding signature
  $\Sigma_{\mathsf{map}} = \{\mathsf{nil} : \langle\rangle \quad \mathsf{cons} : \langle 0, 0\rangle \quad \mathsf{map} : \langle 1, 0\rangle\}$,
  metavariables $Z = \{\mathrm{F}^{(1)}, \mathrm{X}^{(0)}, \mathrm{Y}^{(0)}\}$

- carrier: presheaf of clones $\mathbf{H} \in \mathbf{Set}^{\mathbb{F}}$

$$\mathbf{H}(0) = \mathbb{N}$$

$$\mathbf{H}(n) = (\mathbb{N}^n \longrightarrow \mathbb{N}) \quad (\text{for } n > 0)$$

$$\mathbf{H}(\rho)(f) = f \circ \langle \pi_{\rho 1}, \ldots, \pi_{\rho m}\rangle$$

operations (at n): $\mathsf{nil}^{\mathbf{H}}_n = \mathbf{K1} : \mathbb{N}^n \longrightarrow \mathbb{N}$

$$\mathsf{cons}^{\mathbf{H}}_n(x, y) = (+) \circ \langle x, y, \mathbf{K2}\rangle : \mathbb{N}^n \longrightarrow \mathbb{N}$$

$$\mathsf{map}^{\mathbf{H}}_n(f, a) = f \circ \langle \mathrm{id}, \mathbf{K3}\rangle + (\times) \circ \langle a, f \circ \langle \mathrm{id}, a\rangle\rangle$$

Example (2): Second-Order Polynomial Interpretation

operations (at 0):

$$\mathbf{nil}_0^{\mathbf{H}} = \mathbf{1} \qquad : \mathbb{N}$$

$$\mathbf{cons}_0^{\mathbf{H}}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x} + \boldsymbol{y} + \mathbf{2} \qquad : \mathbb{N}$$

$$\mathbf{map}_0^{\mathbf{H}}(\boldsymbol{f}, \boldsymbol{a}) = \boldsymbol{f}(\mathbf{3}) + \boldsymbol{a} \times \boldsymbol{f}(\boldsymbol{a}) \qquad : \mathbb{N}$$

some calculation:

Homomorphisms

Structure preserving maps for algebras

Def. A homomorphism $\boldsymbol{h}$ between $\boldsymbol{\Sigma}$-algebras $(\boldsymbol{A}, \{\boldsymbol{f}^{\boldsymbol{A}}\}_{\boldsymbol{f}\in\boldsymbol{\Sigma}})$ to $(\boldsymbol{B}, \{\boldsymbol{f}^{\boldsymbol{B}}\}_{\boldsymbol{f}\in\boldsymbol{\Sigma}})$ is an arrow $\boldsymbol{h} : \boldsymbol{A} \longrightarrow \boldsymbol{B}$ of $\mathbf{Set}^{\mathbb{F}}$ such that

$$\boldsymbol{h_n(f_n^A(a_1, \ldots, a_l)) = f_n^B(h_{n+i_1}(a_1), \ldots, h_{n+i_l}(a_l))}$$

Def. A morphism of $\boldsymbol{\Sigma}$-monoids $\boldsymbol{h} : (\boldsymbol{M}, \boldsymbol{\nu}^{\boldsymbol{A}}, \boldsymbol{\mu}^{\boldsymbol{A}}) \longrightarrow (\boldsymbol{M'}, \boldsymbol{\nu}^{\boldsymbol{B}}, \boldsymbol{\mu}^{\boldsymbol{B}})$ is a $\boldsymbol{\Sigma}$-algebra homomorphism that is also a monoid morphism, i.e., it satisfies moreover

$$\boldsymbol{h_n(\nu_n^A(x)) = \nu_n^B(x)}$$

$$\boldsymbol{h_n(\mu_n^{(m)\,A}(a; \overrightarrow{b})) = \mu_n^{(m)\,B}(h_m(a); h_n(b_1), \ldots, h_n(b_m))}$$

Homomorphisms

Initial $\mathbf{V}+\boldsymbol{\Sigma}$-algebra

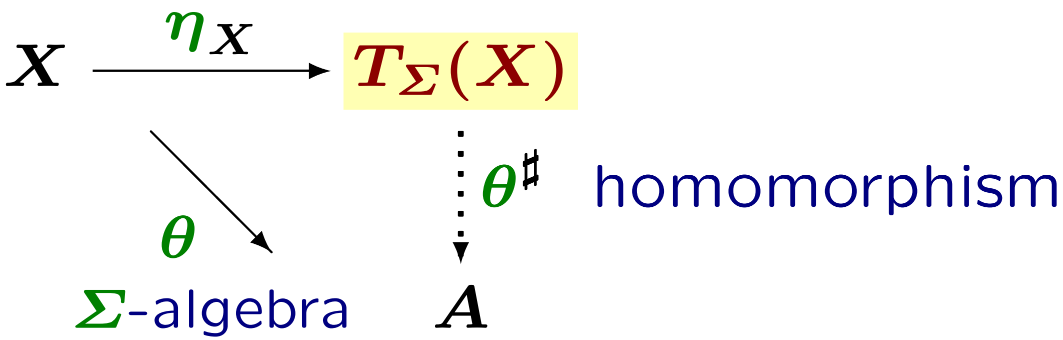
- ▷ Carrier: $\mathbf{T_{\Sigma}V}(\boldsymbol{n}) = \{$all terms using $\boldsymbol{\Sigma}$ with free vars in $\boldsymbol{n}\}$
- ▷ $\exists$ unique homomorphism: $\mathbf{T_{\Sigma}V} \longrightarrow \boldsymbol{A}$ for any $\mathbf{V}+\boldsymbol{\Sigma}$-algebra $\boldsymbol{A}$

Excersise

- ▷ Describe an initial $\boldsymbol{V} + \boldsymbol{\Sigma}_{\mathbf{map}}$-algebra
- ▷ Describe an initial $\boldsymbol{\Sigma}_{\mathbf{map}}$-monoid

TRS: An Important Idea — Freeness of Terms

$T_\Sigma(X)$, the set of all terms forms a free Σ-algebra over X in **Set**



Unique homomorphism = compositional interpretation

$$\theta^\sharp(f(t_1, t_2)) = f_A(\theta^\sharp(t_1), \theta^\sharp(t_2)) \qquad \text{(homomorphic)}$$

Remember the principle of semantics:

$$[\![f(t_1, t_2)]\!] = [\![f]\!]([\![t_1]\!], [\![t_2]\!]) \qquad \text{(compositional)}$$

▶ CRS's case: Does the set of all meta-terms have freeness?

Open problem [Klop'80][Klop, van Oostrom, van Raamsdonk TCS'93]

TRS: Special Case – Initiality of Ground Terms

$T_{\Sigma}(\varnothing)$, the set of all ground terms forms an initial Σ-algebra in **Set**

$$T_{\Sigma}(\varnothing)$$

$$\Big\downarrow \, !_{A} \quad \text{exists unique homomorphism}$$

Σ-algebra A

Unique homomorphism = compositional interpretation

Intuition:

- ▷ Initial algebras = syntax
- ▷ Free algebras over X = syntax generated by X

$\mathbf{Set}^{\mathbb{F}}$: Two "generic" examples of $\boldsymbol{\Sigma}$-monoids

"Generic" means defined for a signature $\boldsymbol{\Sigma}$

(1) The initial $\boldsymbol{\Sigma}$-monoid $\mathbf{T}_{\boldsymbol{\Sigma}}\mathbf{V}$

(= $\boldsymbol{\Sigma}$-monoid over $\mathbf{0}$ = initial $\mathbf{V}+\boldsymbol{\Sigma}$-algebra)

higher-order abstract syntax (without metavariables)

▶ CRS's terms

(2) The free $\boldsymbol{\Sigma}$-monoid $\mathbf{M}_{\boldsymbol{\Sigma}}\boldsymbol{Z}$ over $\underline{\boldsymbol{Z}}$

higher-order abstract syntax with metavariables $\boldsymbol{Z}$

▶ CRS's meta-terms

Note: Terms are special cases of meta-terms.

CRS: Meta-terms (1)

- A binding signature Σ
- Z is an $\mathbb{N}$-indexed set of metavariables parameterised by arities:

$$Z(l) \stackrel{def}{=} \{\mathrm{z} \mid \mathrm{z}^{l}, \text{ where } l \in \mathbb{N}\}.$$

- Raw meta-terms generated by Z:

$$t ::= x \mid f(x_1 \cdots x_{i_1}.t_1, \ldots, x_1 \cdots x_{i_l}.t_l) \mid \mathrm{z}[t_1, \ldots, t_l]$$

- A meta-term t is a raw meta-term derived from:

$$\frac{x \in n}{n \vdash x} \qquad \frac{f : \langle i_1, \ldots, i_l \rangle \in \Sigma \quad n{+}i_1 \vdash t_1 \ \cdots \ n{+}i_l \vdash t_l}{n \vdash f(\, n{+}1 \ldots n{+}i_1.t_1, \ \ldots, \ n{+}1 \ldots n{+}i_l.t_l\,)}$$

$$\frac{\mathrm{z} \in Z(l) \quad n \vdash t_1 \ \cdots \ n \vdash t_l}{n \vdash \mathrm{z}[t_1, \ldots, t_l]}$$

- Every meta-term follows the method of de Bruijn levels
  – some examples

CRS: Meta-terms (2)

▷ Presheaf $\mathrm{M}_{\Sigma} Z \in \mathbf{Set}^{\mathbb{F}}$

$$\mathrm{M}_{\Sigma} Z(n) = \{t \mid n \vdash t\}$$

with the presheaf action $\cdots$ "renaming on variables"

$$\mathrm{M}_{\Sigma} Z(\rho)(x) = \rho(x) \qquad (x \in n)$$

$$\mathrm{M}_{\Sigma} Z(\rho)(f(n+\overrightarrow{i_1}.t_1, \ldots, n+\overrightarrow{i_l}.t_l))$$

$$= f(n+\overrightarrow{i_1}.\mathrm{M}_{\Sigma} Z(\rho)(t_1), \ldots, n+\overrightarrow{i_l}.\mathrm{M}_{\Sigma} Z(\rho)(t_l))$$

$$\mathrm{M}_{\Sigma} Z(\rho)(\mathrm{z}[t_1, \ldots, t_l]) = \mathrm{z}[\mathrm{M}_{\Sigma} Z(\rho)(t_1), \ldots, \mathrm{M}_{\Sigma} Z(\rho)(t_l)]$$

for $\rho : m \to n$ in $\mathbb{F}$.

CRS: Meta-terms (3)

- $\mathbf{V}+\boldsymbol{\Sigma}$-algebra $(\mathrm{M}_{\Sigma}\boldsymbol{Z}, \{\boldsymbol{\nu}, \boldsymbol{f_T}\}_{f\in\Sigma})$

$$\boldsymbol{\nu}(n) : \mathrm{V}(n) \longrightarrow \mathrm{M}_{\Sigma}Z(n),$$
$$x \longmapsto x$$
$$f_T : \delta^{i_1}\mathrm{M}_{\Sigma}Z \times \cdots \times \delta^{i_l}\mathrm{M}_{\Sigma}Z \longrightarrow \mathrm{M}_{\Sigma}Z$$
$$(t_1, \ldots, t_l) \longmapsto f(n+\overrightarrow{i_1}.t_1, \ldots, n+\overrightarrow{i_l}.t_l).$$

- Multiplication $\boldsymbol{\mu}\cdots$ “substitution for variables”

$$\mu_n(\mathsf{var}(i); \overrightarrow{s}) = s_i$$
$$\mu_n(f(t_1, \ldots, t_l); \overrightarrow{s}) = f(\mu_{n+i_1}(t_1; \overrightarrow{s}), \ldots, \mu_{n+i_l}(t_l; \overrightarrow{s}))$$
$$\mu_n(\mathrm{Z}[t_1, \ldots, t_l]; \overrightarrow{s}) = \mathrm{Z}[\mu_n(t_1, \overrightarrow{s}), \ldots, \mu_n(t_l, \overrightarrow{s})]$$

- $\mathrm{M}_{\Sigma}\boldsymbol{Z}$ is a free $\boldsymbol{\Sigma}$-monoid over $\boldsymbol{Z}$.

CRS: Terms

- A term is a meta-term using only the first two rules
- A meta-term t is a raw meta-term derived from:

$$\frac{x \in n}{n \vdash x} \qquad \frac{f : \langle i_1, \ldots, i_l \rangle \in \Sigma \quad n{+}i_1 \vdash t_1 \ \cdots \ n{+}i_l \vdash t_l}{n \vdash f(\, n{+}1 \ldots n{+}i_1 . t_1 ,\ \ldots ,\ n{+}1 \ldots n{+}i_l . t_l \,)}$$

- Presheaf $\mathbf{M}_\Sigma Z \in \mathbf{Set}^{\mathbb{F}}$

$$\mathbf{T}_\Sigma\mathbf{V}(n) = \{t \mid n \vdash t\}$$

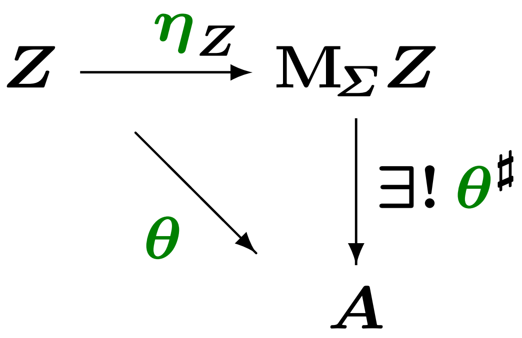
- $\mathbf{T}_\Sigma\mathbf{V} = \mathbf{M}_\Sigma\mathbf{0}$ is a free Σ-monoid over $\mathbf{0}$.
- $\mathbf{T}_\Sigma\mathbf{V} = \mathbf{M}_\Sigma\mathbf{0}$ is an initial $\mathbf{V}+\Sigma$-algebra

. $!_A : \mathbf{T}_\Sigma\mathbf{V} \longrightarrow A$

Free Σ-monoids: HO Syntax with Metavariables

Thm. $(\mathrm{M}_{\Sigma} Z, \nu, \mu)$ forms a free Σ-monoid over Z

▷ Freeness of $\mathrm{M}_{\Sigma} Z$: given assignment θ



Σ-monoid morphism

▷ The unique Σ-monoid morphism $\theta^{\sharp}$ that extends θ

$$\theta^{\sharp}_n : \mathrm{M}_{\Sigma} Z(n) \longrightarrow A(n) \qquad (A, \nu, \mu)\ \Sigma\text{-monoid}$$

$$\theta^{\sharp}_n(i) = \nu_n(i)$$

$$\theta^{\sharp}_n(f(t_1, \ldots, t_l)) = f^A_n(\theta^{\sharp}_{n+i_1}(t_1), \ldots, \theta^{\sharp}_{n+i_l}(t_l))$$

$$\theta^{\sharp}_n(\mathrm{z}[t_1, \ldots, t_l]) = \mu_n(\, \theta_l(\mathrm{z});\ \theta^{\sharp}_n(t_1), \ldots \theta^{\sharp}_n(t_l)\,)$$

Proof. Check: presheaf, Σ-algebra, Σ-monoid, freeness. ▶ (Ex.)

Instance: Valuation

- ▷ Case $\boldsymbol{A} = \mathbf{T_\Sigma V}$

$$\begin{array}{ccc} \boldsymbol{Z} & \xrightarrow{\ \eta_Z\ } & \mathbf{M_\Sigma}\boldsymbol{Z} \\ & {\scriptstyle\theta}\searrow & \big\downarrow \exists!\ \theta^\sharp \qquad \boldsymbol{\Sigma}\text{-monoid morphism} \\ & & \mathbf{T_\Sigma V} \end{array}$$

- ▷ Substitution for metavariables

$$\theta^\sharp_n : \mathbf{M_\Sigma}\boldsymbol{Z}(n) \longrightarrow \mathbf{T_\Sigma V}(n)$$

$$\theta^\sharp_n(i) = i$$

$$\theta^\sharp_n(f(t_1, \ldots, t_l)) = f(\theta^\sharp_{n+i_1}(t_1), \ldots, \theta^\sharp_{n+i_l}(t_l))$$

$$\theta^\sharp_n(\mathrm{z}[t_1, \ldots, t_l]) = \mu_n(\,\theta_l(\mathrm{z});\ \theta^\sharp_n(t_1), \ldots \theta^\sharp_n(t_l)\,)$$

- ▷ This is called a valuation in CRS.

- ▷ Case $\boldsymbol{A} = \mathbf{M_\Sigma}\boldsymbol{Z}$ is also possible (“meta-valuation”).

- ▷ E.g. $\mathbf{M}_{\Sigma_\lambda} Z \longrightarrow \mathbf{T}_{\Sigma_\lambda}\mathbf{V}$

$$\theta^\sharp(\ \lambda(x.\mathrm{z}[x]@y)\ ) = \lambda(\ x.(x@x)@y\ )$$

where $\theta(\mathrm{z}^{(1)}) = \mathbf{1}@\mathbf{1}$

- ▷ Other examples of Σ-monoids: clone $\mathcal{C}(D^{(-)}, D)$ (for Λ), terms with additional a free variable $\delta\mathbf{T}_{\Sigma}\mathbf{V}$.