2006-6-14

individual names
- variables
- constants
  - parameters
  - proper constants $\underbrace{\qquad\qquad}_{\Sigma}$ ($\Sigma \leftarrow \Omega$, $\Sigma \leftarrow A$)

variable $\hat{=}$ argument variable

free variable $\hat{=}$ dangling argument pointer

parameter $\hat{=}$ global variable

Examples

Specification: BA
Constants: $0, 1, -, +, \cdot$
Variables: x, y, z
parameters: a, b, c

$x + 0 = x$

$\lambda x.\, x+0 = \lambda x.x$ ← x occurs in the description but not in the term

$BA,\ a = b \vdash a = a \cdot b \cdot b \cdot a$

$BA,\ x = y \vdash 0 = 1$

$BA,\ a = b \not\vdash 0 = 1$

Notational Conveniences

$e_1, e_2 \vDash\!\dashv e_3 \rightsquigarrow \{e_1, e_2\} \vDash\!\dashv \{e_3\}$

$\ominus A \rightsquigarrow \mathcal{G} \ominus A$

$\ominus e \rightsquigarrow \mathcal{G} \ominus e$

$\ominus s \rightsquigarrow \mathcal{G} \ominus s$

$\mathcal{M} \models A \rightsquigarrow \forall e \in A: \mathcal{M} \models e$

Closed Specifications

A closed $:\Longleftrightarrow \mathcal{N}A \cap Var = \emptyset$

Extensionality $\boxed{s = t \vDash\!\dashv \lambda x.s = \lambda x.t}$

$\Rightarrow$ variables $\hat{=}$ dangling argument references

$\Rightarrow$ every specification is equivalent to a closed specification

$\Rightarrow$ open specifications are a notational convenience

Closed specs have nice properties

Let A be closed. Then:

1) $A \models e \iff \forall \mathcal{M}: \mathcal{M} \models A \Rightarrow \mathcal{M} \models e$ — semantic entailment
2) $A \models e \Rightarrow \Theta A \models \Theta e$
3) $A \vdash e \Rightarrow \Theta A \vdash \Theta e$

$\left.\begin{array}{l} 2) \\ 3) \end{array}\right\}$ stability

$\Rightarrow$ Generativity is a consequence of stability

$\Rightarrow$ Can replace vars with params and vice versa in $A \models e$ and $A \vdash e$

$\Rightarrow$ Variables are not essential for $A \models e$, $A \vdash e$

$\Rightarrow$ Generativity and stability for open specs are consequences of stability for closed specs

Formal proofs

- Gödel-style proofs (derivations)
- conversion proofs

compile

Recall

$A \vdash e :\iff \exists$ Gödel-style proof of e from A

Soundness: $A \vdash e \Rightarrow A \models e$

Example of Gödel-style proof

$fx = a \vdash fy = a$ (f, a constants, x variable)

1	$fx = a$		
2	$\lambda x.fx = \lambda x.a$	ξ	1
3	$(\lambda x.fx)y = (\lambda x.a)y$	CR	2
4	$(\lambda x.fx)y = fy$	β	
5	$(\lambda x.a)y = a$	β	
6	$fy = (\lambda x.fx)y$	Sym	4
7	$fy = (\lambda x.a)y$	Trans	6, 3
8	$fy = a$	Trans	7, 5

Def Gödel-style proof

A Gödel-style proof of e from A is a tuple $(e_1, \dots, e_n)$ such that

1) $e_n = e$
2) $\forall i \in \{1, \dots, n\}$:
   - $e_i \in A$ or
   - exists instance (E, e_i) of a deduction rule such that $E \subseteq \{e_1, \dots, e_{i-1}\}$

This definition will work for every set of inference rules

Example of conversion proof

$BA \vdash x = x \cdot x$

	x			
$=$	$x \cdot 1$	Id	$x = x \cdot 1$	
$=$	$x\,(x + \bar{x})$	Compl	$x + \bar{x} = 1$	Sym, CR
$=$	$xx + x\bar{x}$	Dist	$x(y+z) = xy + xz$	Genericity
$=$	$xx + 0$	Compl	$x\bar{x} = 0$	CR
$=$	xx	Id	$x + 0 = x$	Genericity

Definition of conversion proof

A conversion proof of e from A is a tuple $(s_1, \dots, s_n)$ such that

1) $e = (s_1, s_n)$
2) $\forall i \in \{1, \dots, n-1\}$:

   (s_i, s_{i+1}) is a λ-conversion step or an A-conversion step

Trans, Ref

Def of conversion steps

A λ-conversion step is an equation $s = t$ such that $s = t$ or $t = s$ is a λ-reduction step

An A-conversion step is an equation $s = t$ such that $s = t$ or $t = s$ is an A-reduction step

Sym

Definition of λ-reduction steps

- A β-reduction step is an extension of an equation $(\lambda x.s)\,t = s[x := t]$
- An η-reduction step is an extension of an equation $\lambda x.\, sx = s$ where $x \notin \mathcal{N}x$
- A λ-reduction step is a β- or an η-reduction step

β, η

Definition of A-reduction steps

An A-reduction step is
an extension of an instance of an equation in A

Extensions and instances

- An extension of e is an equation that can be derived from e with CL, CR, ξ

$$x = a$$
$$fx = fa$$
$$\lambda x.fx = \lambda x.fa$$
$$(\lambda x.fx)a = (\lambda x.fa)a$$

- An instance of e is an equation θe where $Ker\, \theta \subseteq Var$

$$fx = y$$
$$fa = y$$
$$(\lambda x.x)x = y$$
$$(\lambda x.x)a = fy$$

$(\beta, \xi, CL, CR, Sym, Trans)$ Generativity

- Conversion proof — Trans, Ref
  - conversion step — Sym
    - λ-reduction step
      - extension
        - β
        - η
    - A-reduction step
      - extension — CL, CR, ξ
        - instance
          - A

$$\boxed{\begin{array}{l} \exists \text{ Gödel-style proof of } e \text{ from } A \\ \Longleftrightarrow \exists \text{ conversion proof of } e \text{ from } A \end{array}}$$

Proof

"$\Longleftarrow$" easy

"$\Longrightarrow$" Show for every instance (E, e') of a deduction rule:

$\forall e'' \in E: (\exists \text{ conversion proof of } e'' \text{ from } A)$
$\Rightarrow (\exists \text{ conversion proof of } e' \text{ from } A)$