Invertible Terms Enumeration.

Carlos C. Martínez

Department of Mathematics and Computer Science,

Wesleyan University

cmartinez@wesleyan.edu

June, 2004

Intro:

- First appearance in the field of mathematical logic, as part of the functional foundation of mathematics approach to avoid paradox.
- A way to ensure the consistency of the operations in modern programming.
- What about functional programming?

About this talk:

Content = Formalisms + Algorithms

Formalisms = Usual definitions and Characterization results

Algorithms = Deciding efficiently type isomorphism

Types

Definition 1 *Let $\mathcal{T}_{\rightarrow}$ be a type system, a set of types expressions of $\Lambda_{\rightarrow}$ inductively defined as follows*

- *(Basic type)* $\iota \in \mathcal{T}_{\rightarrow}$
- *(Function type)* $\forall \tau, \tau' \in \mathcal{T}_{\rightarrow} \Rightarrow (\tau \rightarrow \tau') \in \mathcal{T}_{\rightarrow}$

Similarly, let $\mathcal{T}_{\rightarrow,\times}$ be the type system of $\Lambda_{\rightarrow,\times}$ inductively defined from the previous and the following extra scheme.

- *(Product type)* $\forall \tau, \tau' \in \mathcal{T}_{\rightarrow,\times} \Rightarrow (\tau \times \tau') \in \mathcal{T}_{\rightarrow,\times}$

λ-Calculus

Definition 2 ***($\Lambda_{\rightarrow}$ and $\Lambda_{\rightarrow,\times}$ Terms.)*** *The set $\Lambda_{\rightarrow}$ of lambda terms is inductively defined as the smallest set containing $\mathcal{A} = (\mathcal{V} \cup \Sigma)$ called atoms and closer under the rules.*

- *If $M^{\tau \rightarrow \tau'}$ and N^{τ} are terms, then the application $(MN)^{\tau'}$ is a term.*
- *If $M^{\tau'}$ is a term and $x \in \mathcal{V}_{\tau}$ then the lambda abstraction $(\lambda x.M)^{\tau \rightarrow \tau'}$ is a term.*

Similarly, we define inductively $\Lambda_{\rightarrow,\times}$

- $Pr_1^{\tau \times \tau' \rightarrow \tau}$ *and* $Pr_2^{\tau \times \tau' \rightarrow \tau'}$

and the new rule

- *If* M^{τ} *and* $N^{\tau'}$ *are terms, then the pairing* $\langle M, N \rangle^{\tau \times \tau'}$ *is a term.*

Computation Rules:

Definition 3 $\lambda_{\rightarrow}$ ***-Computation Rules:***

(α*-reduction) If* $y \notin FV(M) \cup BV(M)$*, then*
$\lambda x.M \rhd_{\alpha} \lambda y.M[x := y]$

(β*-reduction)* $(\lambda x.M)N \rhd_{\beta} M[x := N]$

(η*-expansion) If* $x \notin FV(M)$*, then* $M \rhd_{\eta} \lambda x.(Mx)$

$\lambda_{\rightarrow,\times}$ ***-Computation Rules:*** *the previous adding the following extras reductions, expansion*

(π_1*-reduction)* $Pr_1\langle M, N\rangle \rhd_{\pi_1} M$

(π_2*-reduction)* $Pr_2\langle M, N\rangle \rhd_{\pi_2} N$

($\times!$*-expansion)* $M \rhd_{\times!} \langle Pr_1(M), Pr_2(M)\rangle$

The Curry-Howard Isomorphism:

Isomorphism I

Definition 4 ***(Definable isomorphisms, invertible terms.)*** *Two types* τ *and* τ' *are definably isomorphic* ($\tau \cong_d \tau'$) *iff there exist functions (*λ*-terms)* $M^{\tau\to\tau'}$ *and* $N^{\tau'\to\tau}$ *such that* $M \circ N = I^{\tau'}$ *and* $N \circ M = I^{\tau}$*, where* I^{τ} *and* $I^{\tau'}$ *are* $\lambda x^{\tau}.x$ *and* $\lambda x^{\tau'}.x$*, the identities of type* A *and* B*. The terms* M *and* N *are called invertible. We also write* $M : \tau \cong_d \tau' : N$ *when we want to make the associated invertible terms explicit.*

Where $\circ$ is infix notation for the usual function composition $\lambda x.\lambda f.\lambda g.f(gx)$

Isomorphism II

Definition 5 ***(Semantic isomorphisms.)*** *Two types τ and τ' are isomorphic in a specific model $\mathcal{M}$ if their interpretations are isomorphic in $\mathcal{M}$ in the following sense, there are in the model invertible functions f and g between, then we write $\mathcal{M} \models \tau \cong \tau'$. Two types will be Semantically isomorphic, noted $\tau \cong \tau'$, if $\mathcal{M} \models \tau \cong \tau'$ holds for all $\mathcal{M}$ of the calculus.*

Theorem 1 *Let τ and τ' be types then $\tau \cong_{\mathrm{d}} \tau' \Leftrightarrow \tau \cong \tau'$.*

Isomorphism III

Definition 6 ***(Soundness, completeness.)*** *We say that an equational theory Th is a sound theory of isomorphism for a calculus if*

$$\forall \tau, \tau' \ \epsilon \ \mathcal{T} : Th \vdash \tau = \tau' \Rightarrow \tau \cong \tau'$$

Respectively, an equational theory Th is a complete theory of isomorphism for a calculus

$$\forall \tau, \tau' \ \epsilon \ \mathcal{T} : Th \vdash \tau \cong \tau' \Rightarrow \tau = \tau'$$

Where $\mathcal{T}$ is an arbitrary type system.

Theory: $Th^1{}_{\rightarrow}$

Definition 7 *$Th^1{}_{\rightarrow}$ be the theory of equality on $\mathcal{T}_{\rightarrow}$ plus the following axiom scheme.*

$$\forall \tau, \tau', \tau'' \in \mathcal{T}_{\rightarrow} : \tau \rightarrow (\tau' \rightarrow \tau'') = \tau' \rightarrow (\tau \rightarrow \tau'') \qquad \textbf{(Swap)}$$

Finite Hereditary Permutation

Definition 8 ***(f.h.p.)*** *An untyped λ-term M is a f.h.p if and only if*

- $M = \lambda z.z$,
- $M = \lambda z.\lambda x_1 \ldots \lambda x_n.z(Q_1 x_{\pi(1)}) \ldots (Q_n x_{\pi(n)})$ *where* $\pi : n \to n$ *is a permutation and* Q_i *is a f.h.p. for all* $1 \leq i \leq n$

Theorem 2 *(Dezani-Ciancaglini 1976) Let M be an untyped term possessing $\beta\eta$-normal form then M is invertible if and only if M is a finite hereditary permutation.*

Building $Th^1_{\rightarrow}$ invertible terms.

Algorithm 1 *:*

BIT1: Types $\times$ *Types* $\longrightarrow$ *(Terms)Set*

Instance: BIT1$(\tau, \tau') \longrightarrow \{M \mid M : \tau \cong_d \tau' : M^{-1}\}$

if length$(\tau) \neq$ *length*(τ') *then* $\emptyset$
else
$n :=$ *length*(τ)*;*
BIT1(ι, ι) *:=* $\lambda z^{\iota}.z$*;*
BIT1(τ, τ') *:=* $\emptyset$*;*
For $\pi \in$ *permutation*(n) *do*
While BIT1$(\tau'_{\pi(i)}, \tau_i) \neq \emptyset$ *do*
$Inv_\pi = \{\lambda z^{\tau}.\lambda x_1{}^{\tau'_1} \ldots \lambda x_n{}^{\tau'_n}.z(Q_1 x_{\pi(1)}) \ldots (Q_n x_{\pi(n)}) \mid Q_i \in BIT1(\tau'_{\pi(i)}, \tau_i)\}$
BIT1(τ, τ') *:= BIT1*$(\tau, \tau') \cup Inv_\pi$
Return$(BIT1(\tau, \tau'))$*;*

Where input := $\{\tau = [\tau_1, \tau_2, \ldots, \tau_n, \iota], \tau' = [\tau'_1, \tau'_2, \ldots, \tau'_m, \iota]\}$

Theory: $Th^1{}_{\rightarrow,\times}$

Definition 9 *$Th^1{}_{\rightarrow,\times}$ be the theory of equality on $\mathcal{T}_{\rightarrow,\times}$ plus the following axiom schemes.*

$\forall \tau, \tau', \tau'' \in \mathcal{T}_{\rightarrow,\times}$:

$$\tau \times \tau' = \tau' \times \tau \qquad [\boldsymbol{C}_{\times}]$$

$$\tau \times (\tau' \times \tau'') = (\tau \times \tau') \times \tau'' \qquad [\boldsymbol{A}_{\times}]$$

$$(\tau \times \tau') \rightarrow \tau'' = \tau \rightarrow (\tau' \rightarrow \tau'') \qquad [\boldsymbol{Curry}]$$

$$\tau \rightarrow (\tau' \times \tau'') = (\tau \rightarrow \tau') \times (\tau \rightarrow \tau'') \qquad [\boldsymbol{Split}]$$

Rewriting Types

Definition 10 *Let $\mathcal{R}_{\rightarrow,\times}$ be the transitive and substitutive reduction relation generated by*

1. $\tau \times (\tau' \times \tau'') \rhd (\tau \times \tau') \times \tau''$
2. $(\tau \times \tau') \rightarrow \tau'' \rhd \tau \rightarrow (\tau' \rightarrow \tau'')$
3. $\tau \rightarrow (\tau' \times \tau'') \rhd (\tau \rightarrow \tau') \times (\tau \rightarrow \tau'')$

Proposition 1 *Each type has a unique type normal form in $\mathcal{R}_{\rightarrow,\times}$*

Background

Proposition 2 *Let* $I_{\tau,\tau'} := \{M \mid M : \tau \cong_d \tau' : M^{-1}\}$ *then there is a 1-1 correspondence between* $I_{\tau,\tau'}$ *and* $I_{nf(\tau),nf(\tau')}$

$\times$-Finite Hereditary Permutation

Definition 11 ***(INV$_{\rightarrow,\times}$ Invertible Terms.)*** *Let M^{τ} be a typed $\lambda_{\rightarrow,\times}$-term such that τ is $\mathcal{R}_{\rightarrow,\times}$-type normal form, it is a invertible term if and only if*

- $M_{\tau} = \lambda z^{\tau}.z$,
- $M_{\tau} = \lambda\langle z_1,\ldots,z_n\rangle^{\tau}.\langle P_1 z_{\pi(1)},\ldots,P_n z_{\pi(n)}\rangle$ *where $\pi : n \to n$ is a permutation, such that P_i is a f.h.p. for all $1 \leq i \leq n$ and*

Building $Th^1_{\rightarrow,\times}$-invertible terms.

Algorithm 2 :

BIT1$_\times$: *Types* $\times$ *Types* $\longrightarrow$ *(Terms)Set*

Instance: $BIT1_\times(\tau,\tau') \longrightarrow \{M \mid M : \tau \cong_d \tau' : M^{-1}\}$

if $length_\times(\tau) \neq length_\times(\tau')$ *then* $\emptyset$
else
$n := length_\times(\tau)$;
$BIT1_\times(\iota,\iota) := \lambda z^\iota . z$;
$BIT1_\times(\tau,\tau') := \emptyset$;
For $\pi \in permutation(n)$ *do*
While $(BIT1(\tau_{\pi(i)},\tau'_i) \neq \emptyset, \forall i \leq n)$ *do*
$Inv_\pi = \{\lambda\langle z_1,\ldots z_n\rangle^\tau . \langle P_1 z_{\pi(1)},\ldots,P_n z_{\pi(n)}\rangle \mid P_i \in BIT1(\tau_{\pi(i)},\tau'_i)\}$;
$BIT1_\times(\tau,\tau') := BIT1_\times(\tau,\tau') \bigcup Inv_\pi$
Return$(BIT1_\times(\tau,\tau'))$;

Where input := $\{\ \tau = (\tau_1 \times \tau_2,\ldots,\times\tau_n), \tau' = (\tau'_1 \times \tau'_2,\ldots,\times\tau'_m)\ \}$