Intersection types and explicit substitution: an overview

Daniel L. Ventura[1] & Mauricio Ayala-Rincón[2] & Fairouz D. Kamareddine[3]

[1]Instituto de Informática - INF/UFG, Brazil
[2]Departamentos de Matemática e Ciência da Computação - GTC/UnB, Brazil
[3]ULTRA Group, MACS/Heriot-Watt University, Edinburgh, UK

International Workshop on 75 Years of Lambda-Calculus
University of St Andrews, Scotland 15th June, 2012

The λ-calculus and explicit substitution

Proposed by Church in 1932. [Church32]

$$\textit{Terms} \qquad M := x \mid (M\ M) \mid \lambda_x.M$$

Computations (reductions) are made by a unique rule:

$$(\lambda_x.M\ N) \longrightarrow M\{x := N\} \qquad (\beta)$$

Calculi with Explicit Substitutions extends it:

$$(\lambda_x.M\ N) \longrightarrow M\langle x := N\rangle \qquad (Beta)$$

▸ JumpES

There are different approaches of expliciting the substitutions.

Typing Systems

Simply typed λ-calculus proposed by Church.[Church40]

Classify objects (terms) in the formal system.

$\lambda_{x:int}.x : int \rightarrow int \quad \lambda_{x:bool}.x : bool \rightarrow bool \quad$ (*à la Church*)

$\lambda_{x}.x : int \rightarrow int \quad \lambda_{x}.x : bool \rightarrow bool \quad$ (*à la Curry*)

STLC is related to IPL: Curry-Howard(-de Bruijn) Isomorphism.

If $M : \langle \Gamma \vdash \tau \rangle$ then $\langle \Gamma \vdash \tau \rangle$ is called a typing of M.

Typing Systems Properties

- Subject Reduction (SR)

  If $M\!:\!\langle \Gamma \vdash \tau \rangle$ and $M \to_\beta N$, then $N\!:\!\langle \Gamma \vdash \tau \rangle$

- Subject Expansion (SE)

  If $N\!:\!\langle \Gamma \vdash \tau \rangle$ and $M \to_\beta N$, then $M\!:\!\langle \Gamma \vdash \tau \rangle$

- Strong or Weak Normalisation (SN and WN) for typable terms.

- Type Inference (M : ?)

- Principal Typing (PT): The most general typing.

- Inhabitation Problem (? : $\langle \Gamma \vdash \tau \rangle$)

SR and SN/WN

$$\frac{\lambda_x.M : \langle \Gamma \vdash \sigma \to \tau \rangle \qquad N : \langle \Gamma \vdash \sigma \rangle}{(\lambda_x.M\ N) : \langle \Gamma \vdash \tau \rangle} \quad \Rightarrow \quad \frac{M : \langle \Gamma, (x : \sigma) \vdash \tau \rangle \qquad N : \langle \Gamma \vdash \sigma \rangle}{M\{x := N\} : \langle \Gamma \vdash \tau \rangle}$$

$$\dfrac{\dfrac{\begin{matrix}[\sigma]^x \\ \mathcal{D} \\ \tau\end{matrix}}{\sigma \to \tau} \qquad \begin{matrix}\mathcal{D}' \\ \sigma\end{matrix}}{\tau} \quad \Rightarrow \quad \begin{matrix}\mathcal{D}' \\ [\sigma] \\ \mathcal{D} \\ \tau\end{matrix}$$

Principal Typing

Let $M\!:\!\langle\Gamma \vdash \tau\rangle$ be a type judgement in some type system $\mathcal{S}$

- $\Phi = \langle\Gamma \vdash_{\mathcal{S}} \tau\rangle$ is a typing of M in $\mathcal{S}$ ($M : \Phi$).
- Φ is a **principal typing** (PT) of M if $M : \Phi$ and Φ represents any other possible typing of M.
- PT property allows:
  - Separate Compilation [Jim96]
  - Compositional Software Analysis [Wells2002]

Principal Typing vs. Principal Type [Jim96]
Given term M and context Γ, τ is a **principal type** of M if it represents any other possible type of M in Γ.

Principal Type: $M : \langle \Gamma \vdash ? \rangle$

Example

Question: $\lambda_x.(y\ x) : \langle y{:}\alpha \to \alpha \vdash ? \rangle$

Answer: $\alpha \to \alpha$

Question: $\lambda_x.(y\ x) : \langle y{:}\alpha \to \beta \vdash ? \rangle$

Answer: $\alpha \to \beta$

Principal Typing vs. Principal Type

Principal Typing: $M : \langle ? \vdash ? \rangle$

Example

Question: $\lambda_x.(y\ x) : \langle ? \vdash ? \rangle$

Possible Typing: $\langle y{:}\alpha \to \beta \vdash \alpha \to \beta \rangle$

Possible Typing: $\langle y{:}\alpha \to \alpha \vdash \alpha \to \alpha \rangle$

Many many more

Principle Typing: $\langle y{:}\alpha \to \beta \vdash \alpha \to \beta \rangle$

Question: $\lambda_x.(y\ (y\ x)) : \langle ? \vdash ? \rangle$

Possible Typing: $\langle y{:}(\alpha \to \beta) \to \alpha \to \beta \vdash (\alpha \to \beta) \to \alpha \to \beta \rangle$

Many many more

Principle Typing: $\langle y{:}\alpha \to \alpha \vdash \alpha \to \alpha \rangle$

Principal Typing vs. Principal Type

	Principal Type	Principal Typing
STLC	✓ [Hi97]	✓ [Wells2002]
Hindley/Milner	✓ [DM82]	X [Wells2002]
System F	X [Wells2002]	X [Wells2002]
System $\mathbb{I}$	✓ [KW04]	✓ [KW04]

Intersection type discipline

- Introduced by Coppo and Dezani-Ciancaglini. [CDC78, CDC80]
- Characterisation of the SN terms of the λ-calculus. [Pottinger80]
- It incorporates type polymorphism in a finitary way:

$$\lambda_x.x : (int \to int) \wedge (bool \to bool)$$

- PT has been verified in IT systems. [Bakel95, SM96a, KW04]
- Exists IT systems for explicit substitution (ES) calculi (e.g. $\lambda\mathrm{x}$ [LLDDvB2004], $\lambda\mathrm{ex}$ [Kesner09])
- There is not much work about IT systems for calculi used in implementations. (e.g. $\lambda\sigma$ and λs_e)

The System $\mathcal{E}$ [LLDDvB2004]

$$\text{(start)}\ \frac{}{x:\langle\Gamma \vdash \tau\rangle}\ ,(x:\tau)\in\Gamma \qquad \rightarrow_i\ \frac{M:\langle\Gamma,(x:\sigma)\vdash\tau\rangle}{\lambda x.M:\langle\Gamma\vdash\sigma\rightarrow\tau\rangle}$$

$$\text{(cut)}\ \frac{M:\langle\Gamma,(x:\sigma)\vdash\tau\rangle \qquad N:\langle\Gamma\vdash\sigma\rangle}{M\langle x:=N\rangle:\langle\Gamma\vdash\tau\rangle} \qquad \rightarrow_e\ \frac{M:\langle\Gamma\vdash\sigma\rightarrow\tau\rangle \qquad N:\langle\Gamma\vdash\sigma\rangle}{(M\ N):\langle\Gamma\vdash\tau\rangle}$$

$$\text{(drop)}\ \frac{M:\langle\Gamma,(x:\sigma)\vdash\tau\rangle \qquad N:\langle\Delta\vdash\sigma\rangle}{M\langle x:=N\rangle:\langle\Gamma\vdash\tau\rangle}\ ,x\notin av(M) \qquad \cap_i\ \frac{M:\langle\Gamma\vdash\sigma\rangle \qquad M:\langle\Gamma\vdash\tau\rangle}{M:\langle\Gamma\vdash\tau\cap\sigma\rangle}$$

$$\text{(K-cut)}\ \frac{M:\langle\Gamma,(x:\sigma)\vdash\tau\rangle \qquad N:\langle\Delta\vdash\sigma\rangle}{M\langle x:=N\rangle:\langle\Gamma\vdash\tau\rangle}\ ,x\notin\Gamma \qquad \cap_e\ \frac{M:\langle\Gamma\vdash\sigma_1\cap\sigma_2\rangle}{M:\langle\Gamma\vdash\sigma_i\rangle}\ ,i\in\{1,2\}$$

The **intersection types** are defined by:

$$\tau,\sigma\in\mathcal{T} ::= \mathcal{A} \mid \mathcal{T}\rightarrow\mathcal{T} \mid \mathcal{T}\cap\mathcal{T}$$

Properties [LLDDvB2004]

Typable terms are SN

- Available variables vs. Free variables:

$$av(M\langle x := N\rangle) = \begin{cases} av(M) \cup av(N), & \text{if } x \in av(M) \\ av(M), & \text{otherwise} \end{cases}$$

- $(\lambda_z.(\lambda_y.x\ (z\ z))\ \lambda_w.(w\ w))$ is not typable but $(\lambda_z.x\langle y := (z\ z)\rangle\ \lambda_w.(w\ w))$ has a typing in $\mathcal{E}$.

Weak SE:

If N is typable and $M \rightarrow_\beta N$ then M is typable.

The system $\mathcal{E}_\omega$, with rule (ω_i), has a characterisation of WN terms.

$$\omega_i \; \frac{}{M : \langle \Gamma \vdash \omega \rangle}$$

The λ-calculus with de Bruijn indices (λ_{dB})

Invented by N.G. de Bruijn[dB72].

$$\text{Terms} \quad M ::= \underline{n} \,|\, (M\ M) \,|\, \lambda.M \quad \text{for } n \in \mathbb{N}_* = \mathbb{N} \setminus \{0\}$$

Examples

$\lambda.(\lambda.(\underline{1}\ \underline{4}\ \underline{2})\ \underline{1})$

$\lambda.\underline{1} \simeq \lambda x.x \simeq \lambda y.y$

Definition (β-contraction)

$$(\lambda.M\ N) \rhd_\beta \{\underline{1}/N\}M$$

The λs_e-calculus [KR97]

Terms $M ::= \underline{n} \,|\, (M\ M) \,|\, \lambda.M \,|\, M\sigma^i M \,|\, \varphi^j_k M,$

where $n, i, j \in \mathbb{N}^*$ and $k \in \mathbb{N}$

- Extension of λs [KR95], allowing composition.
- Natural extension for the λ_{dB}-calculus
- One-sorted calculus: Terms.
- Introduces operators to realise substitutions and updatings.
- The β-substitution is simulated by the term $M\sigma^i N$

$$(\lambda.M\ N) \longrightarrow M\sigma^1 N \qquad (\sigma\text{-generation})$$

λs rewriting rules [KR95]

$(\lambda.M\ N)$	$\longrightarrow$	$M\sigma^1 N$	(σ-generation)
$(\lambda.M)\,\sigma^i N$	$\longrightarrow$	$\lambda.(M\sigma^{i+1}N)$	(σ-λ-transition)
$(M_1\ M_2)\sigma^i N$	$\longrightarrow$	$((M_1\sigma^i N)\ (M_2\sigma^i N))$	(σ-app-transition)
$\underline{n}\,\sigma^i N$	$\longrightarrow$	$\begin{cases} \underline{n-1} & \text{if } n > i \\ \varphi^i_0 N & \text{if } n = i \\ \underline{n} & \text{if } n < i \end{cases}$	(σ-destruction)
$\varphi^i_k(\lambda.M)$	$\longrightarrow$	$\lambda.(\varphi^i_{k+1}M)$	(φ-λ-transition)
$\varphi^i_k(M_1\ M_2)$	$\longrightarrow$	$((\varphi^i_k M_1)\ (\varphi^i_k M_2))$	(φ-app-transition)
$\varphi^i_k\ \underline{n}$	$\longrightarrow$	$\begin{cases} \underline{n+i-1} & \text{if } n > k \\ \underline{n} & \text{if } n \leq k \end{cases}$	(φ-destruction)

λs_e rewriting rules [KR97]

$(M_1\sigma^i M_2)\sigma^j N$	$\longrightarrow$	$(M_1\sigma^{j+1}N)\sigma^i(M_2\sigma^{j-i+1}N)$ if $i \leq j$	(σ-σ-transition)
$(\varphi^i_k M)\sigma^j N$	$\longrightarrow$	$\varphi^{i-1}_k M$ if $k < j < k+i$	(σ-φ-transition 1)
$(\varphi^i_k M)\,\sigma^j N$	$\longrightarrow$	$\varphi^i_k(M\sigma^{j-i+1}N)$ if $k+i \leq j$	(σ-φ-transition 2)
$\varphi^i_k(M\sigma^j N)$	$\longrightarrow$	$(\varphi^i_{k+1}M)\sigma^j(\varphi^i_{k+1-j}N)$ if $j \leq k+1$	(φ-σ-transition)
$\varphi^i_k(\varphi^j_l M)$	$\longrightarrow$	$\varphi^j_l(\varphi^i_{k+1-j}M)$ if $l+j \leq k$	(φ-φ-transition 1)
$\varphi^i_k(\varphi^j_l M)$	$\longrightarrow$	$\varphi^{j+i-1}_l M$ if $l \leq k < l+j$	(φ-φ-transition 2)

The $\lambda\sigma$-calculus [ACCL91]

$$\text{Terms} \quad M ::= \underline{1} \mid (M\ M) \mid \lambda.M \mid M[S]$$

$$\text{Substitutions} \quad S ::= id \mid \uparrow \mid M.S \mid S \circ S$$

- Two-sorted calculus: Terms and Substitutions.
- Defined with n-ary substitutions.
- Allows *composition* (unrestricted composition).
- Uses de Bruijn indices: $\underline{n+1} \cong \underline{1}[\uparrow^n]$.
- The β-reduction is simulated by the term $M[S]$ (*closure*)

$$(\lambda.M\ N) \longrightarrow M[N.id] \qquad (Beta)$$

$\lambda\sigma$ rewriting rules

$(\lambda.M\ N)$	$\longrightarrow$	$M[N.id]$	$(Beta)$
$(M\ N)[S]$	$\longrightarrow$	$(M[S]\ N[S])$	(App)
$\underline{1}\,[M.S]$	$\longrightarrow$	M	$(VarCons)$
$M[id]$	$\longrightarrow$	M	(Id)
$(\lambda.M)[S]$	$\longrightarrow$	$\lambda.(M[\underline{1}.(S\circ\uparrow)])$	(Abs)
$(M[S])[S']$	$\longrightarrow$	$M[S \circ S']$	$(Clos)$
$id \circ S$	$\longrightarrow$	S	(IdL)
$\uparrow\circ\,(M.S)$	$\longrightarrow$	S	$(ShiftCons)$
$(S_1 \circ S_2) \circ S_3$	$\longrightarrow$	$S_1 \circ (S_2 \circ S_3)$	$(AssEnv)$
$(M.S) \circ S'$	$\longrightarrow$	$M[S'].(S \circ S')$	$(MapEnv)$
$S \circ id$	$\longrightarrow$	S	(IdR)
$\underline{1}\,.\uparrow$	$\longrightarrow$	id	$(VarShift)$
$\underline{1}\,[S].(\uparrow\circ\, S)$	$\longrightarrow$	S	$(Scons)$

Simple types for λ_{dB}

Definition (Simple Types and Contexts)

Types $\sigma, \tau \in \mathcal{S} ::= \mathcal{A} \mid \mathcal{S} \to \mathcal{S}$ **Contexts** $\Gamma ::= nil \mid \sigma.\Gamma$

System $\lambda_{dB}^{\to}$

$$(\text{var}) \frac{}{\underline{1} : \langle \tau.\Gamma \vdash \tau \rangle} \qquad \to_i \frac{M : \langle \sigma.\Gamma \vdash \tau \rangle}{\lambda.M : \langle \Gamma \vdash \sigma \to \tau \rangle}$$

$$(\text{varn}) \frac{\underline{n} : \langle \Gamma \vdash \tau \rangle}{\underline{n+1} : \langle \sigma.\Gamma \vdash \tau \rangle} \qquad \to_e \frac{M : \langle \Gamma \vdash \sigma \to \tau \rangle \qquad N : \langle \Gamma \vdash \sigma \rangle}{(M\ N) : \langle \Gamma \vdash \tau \rangle}$$

Simple types for λs_e
The System $\lambda s_e{}^{\rightarrow}$

$$(\mathrm{var})\ \frac{}{\underline{1} : \langle \tau.\Gamma \vdash \tau \rangle} \qquad \rightarrow_i\ \frac{M : \langle \sigma.\Gamma \vdash \tau \rangle}{\lambda.M : \langle \Gamma \vdash \sigma \rightarrow \tau \rangle}$$

$$(\mathrm{varn})\ \frac{\underline{n} : \langle \Gamma \vdash \tau \rangle}{\underline{n+1} : \langle \sigma.\Gamma \vdash \tau \rangle} \qquad \rightarrow_e\ \frac{M : \langle \Gamma \vdash \sigma \rightarrow \tau \rangle \qquad N : \langle \Gamma \vdash \sigma \rangle}{(M\ N) : \langle \Gamma \vdash \tau \rangle}$$

$$(\varphi)\ \frac{M : \langle \Gamma_{\leq k}.\Gamma_{\geq k+i} \vdash \tau \rangle}{\varphi^i_k M : \langle \Gamma \vdash \tau \rangle}\ ,\ \text{where } |\Gamma| \geq k + i - 1$$

$$(\sigma)\ \frac{N : \langle \Gamma_{\geq i} \vdash \rho \rangle \qquad M : \langle \Gamma_{<i}.\rho.\Gamma_{\geq i} \vdash \tau \rangle}{M \sigma^i N : \langle \Gamma \vdash \tau \rangle}\ ,\ \text{where } |\Gamma| \geq i - 1$$

Simple types for $\lambda\sigma$

The System $\lambda\sigma^{\rightarrow}$

Terms

$$(\text{var})\ \frac{}{\underline{1} : \langle \tau.\Gamma \vdash \tau \rangle} \qquad \rightarrow_e\ \frac{M_1 : \langle \Gamma \vdash \sigma \rightarrow \tau \rangle \qquad M_2 : \langle \Gamma \vdash \sigma \rangle}{(M_1\ M_2) : \langle \Gamma \vdash \tau \rangle}$$

$$\rightarrow_i\ \frac{M : \langle \sigma.\Gamma \vdash \tau \rangle}{\lambda.M : \langle \Gamma \vdash \sigma \rightarrow \tau \rangle} \qquad (\text{clos})\ \frac{S : \langle \Gamma \rhd \Gamma' \rangle \qquad M : \langle \Gamma' \vdash \tau \rangle}{M[S] : \langle \Gamma \vdash \tau \rangle}$$

Substitutions

$$(\text{id})\ \frac{}{id : \langle \Gamma \rhd \Gamma \rangle} \qquad (\text{cons})\ \frac{M : \langle \Gamma \vdash \tau \rangle \qquad S : \langle \Gamma \rhd \Gamma' \rangle}{M.S : \langle \Gamma \rhd \tau.\Gamma' \rangle}$$

$$(\text{shift})\ \frac{}{\uparrow : \langle \tau.\Gamma \rhd \Gamma \rangle} \qquad (\text{comp})\ \frac{S : \langle \Gamma \rhd \Gamma' \rangle \qquad S' : \langle \Gamma' \rhd \Gamma'' \rangle}{S' \circ S : \langle \Gamma \rhd \Gamma'' \rangle}$$

Simple type systems and ES
Both $\lambda\sigma^{\rightarrow}$ and $\lambda s_e{}^{\rightarrow}$:

- Have application on HOU [DHK95, AK01].
- Typable terms are WN. [Gou98, AKR07]
- Have PT. [VAK08]
  (equivalence with Wells' PT)
- Are not PSN

  "If M is SN in λ then M is SN in $\lambda\sigma/\lambda s_e$"

  $\lambda\sigma^{\rightarrow}$: Melliès' example [Mellies95]
  $\lambda s_e{}^{\rightarrow}$: Guillaume's example [Guillaume2000]
- Unrestricted composition is the one to blame.[Ritter99]

IT systems and ES calculi [VAK10b]

- We studied a couple of IT system for the λ_{dB}. [VAK09, VAK10]
- The systems λ_{dB}^{SM} introduced in [VAK10] is a de Bruijn version of the system in [SM96a].
- Two properties were sought after in the IT systems' development:
  - SR property
  - Relevance
- The IT systems introduced for both $\lambda\sigma$ and λs_e were based on the system $\lambda_{dB}^{\wedge}$, a variation of the system λ_{dB}^{SM}.
- The systems $\lambda\sigma^{\wedge}$ and $\lambda s_e{}^{\wedge}$ have the SR property.

Definition (Restricted intersection types and contexts)

1. The **restricted intersection types** are defined by:

$$\tau, \sigma \in \mathcal{T} ::= \mathcal{A} \,|\, \mathcal{U} \to \mathcal{T}$$
$$u, v \in \mathcal{U} ::= \omega \,|\, \mathcal{U} \wedge \mathcal{U} \,|\, \mathcal{T}$$

$\wedge$ is commutative, associative and has ω as neutral element.

2. **Contexts**: $\Gamma ::= nil \,|\, u.\Gamma$ s.t. $u \in \mathcal{U}$

$$nil \wedge \Gamma = \Gamma \wedge nil = \Gamma \quad (u_1.\Gamma) \wedge (u_2.\Delta) = (u_1 \wedge u_2).(\Gamma \wedge \Delta)$$

The system $\lambda^{\wedge}_{dB}$

System $\lambda^{\wedge}_{dB}$

$$\text{var}\ \frac{}{\underline{1} : \langle \tau.nil \vdash \tau \rangle}\ ,\ \tau \in \mathcal{T} \qquad \to'_i\ \frac{M : \langle nil \vdash \tau \rangle}{\lambda.M : \langle nil \vdash \omega \to \tau \rangle}$$

$$\text{varn}\ \frac{\underline{n} : \langle \Gamma \vdash \tau \rangle}{\underline{n+1} : \langle \omega.\Gamma \vdash \tau \rangle} \qquad \to_i\ \frac{M : \langle u.\Gamma \vdash \tau \rangle}{\lambda.M : \langle \Gamma \vdash u \to \tau \rangle}$$

$$\to^{\omega}_{e}\ \frac{M_1 : \langle \Gamma \vdash \omega \to \tau \rangle}{(M_1\ M_2) : \langle \Gamma \vdash \tau \rangle}$$

$$\to_e\ \frac{M_1 : \langle \Gamma \vdash \wedge^{n}_{i=1} \sigma_i \to \tau \rangle \qquad M_2 : \langle \Delta^1 \vdash \sigma_1 \rangle\ \ldots\ M_2 : \langle \Delta^n \vdash \sigma_n \rangle}{(M_1\ M_2) : \langle \Gamma \wedge \Delta^1 \wedge \cdots \wedge \Delta^n \vdash \tau \rangle}$$

Intersection types for λs_e

System $\lambda s_e{}^{\wedge}$

$$(nil\text{-}\sigma)\ \frac{M:\langle\Gamma\vdash\tau\rangle}{M\sigma^i N:\langle\Gamma\vdash\tau\rangle}\ ,\ |\Gamma|<i \qquad (\omega\text{-}\sigma)\ \frac{M:\langle\Gamma\vdash\tau\rangle}{M\sigma^i N:\langle\Gamma_{<i}.\Gamma_{>i}\vdash\tau\rangle}\ ,\ \Gamma_i=\omega$$

$$(\wedge\text{-}nil\text{-}\sigma)\ \frac{N:\langle nil\vdash\sigma_1\rangle\ \ldots\ N:\langle nil\vdash\sigma_m\rangle \qquad M:\langle\omega^{\underline{i-1}}.\wedge_{j=1}^m\sigma_j.nil\vdash\tau\rangle}{M\sigma^i N:\langle nil\vdash\tau\rangle}$$

$$(\wedge\text{-}\omega\text{-}\sigma)\ \frac{N:\langle nil\vdash\sigma_1\rangle\ \ldots\ N:\langle nil\vdash\sigma_m\rangle \qquad M:\langle\Gamma\vdash\tau\rangle}{M\sigma^i N:\langle\Gamma_{<(i-k)}.nil\vdash\tau\rangle}\ ,\ \Gamma_i=\wedge_{j=1}^m\sigma_j\ (*)$$

$$(\wedge\text{-}\sigma)\ \frac{N:\langle\Delta^1\vdash\sigma_1\rangle\ \ldots\ N:\langle\Delta^m\vdash\sigma_m\rangle \qquad M:\langle\Gamma\vdash\tau\rangle}{M\sigma^i N:\langle(\Gamma_{<i}.\Gamma_{>i})\wedge\omega^{\underline{i-1}}.(\Delta^1\wedge\cdots\wedge\Delta^m)\vdash\tau\rangle}\ ,\ \Gamma_i=\wedge_{j=1}^m\sigma_j\ (**)$$

$$(\omega\text{-}\varphi)\ \frac{M:\langle\Gamma\vdash\tau\rangle}{\varphi_k^i M:\langle\Gamma_{\leq k}.\omega^{\underline{i-1}}.\Gamma_{>k}\vdash\tau\rangle}\ ,\ |\Gamma|>k \qquad (nil\text{-}\varphi)\ \frac{M:\langle\Gamma\vdash\tau\rangle}{\varphi_k^i M:\langle\Gamma\vdash\tau\rangle}\ ,\ |\Gamma|\leq k$$

(*) $\Gamma=\Gamma_{<(i-k)}.\omega^{\underline{k}}.\wedge_{j=1}^m\sigma_j.nil$ and $\Gamma_{(i-k-1)}\neq\omega$

(**) $\Delta^k\neq nil$, for some $1\leq k\leq m$, or $\Gamma_{>i}\neq nil$

System $\lambda s_e{}^{\wedge}$ properties

Theorem (SR for $\lambda s_e{}^{\wedge}$)

If $M : \langle \Gamma \vdash \tau \rangle$ *and* $M \rightarrow_{\lambda s_e} M'$*, then* $M' : \langle \Gamma \vdash \tau \rangle$*.*

Corollary (SR for $\lambda^{\wedge}_{dB}$)

If $M : \langle \Gamma \vdash_{\lambda^{\wedge}_{dB}} \tau \rangle$ *and* $M \rightarrow_{\beta} M'$*, then* $M' : \langle \Gamma \vdash \tau \rangle$*.*

The system $\lambda\sigma^{\wedge}$

Terms

$$\text{var}\ \frac{}{\underline{1} : \langle \tau.nil \vdash \tau \rangle} \qquad \to_i\ \frac{M : \langle u.\Gamma \vdash \tau \rangle}{\lambda.M : \langle \Gamma \vdash u \to \tau \rangle}$$

$$\to_e^{\omega}\ \frac{M_1 : \langle \Gamma \vdash \omega \to \tau \rangle}{(M_1\ M_2) : \langle \Gamma \vdash \tau \rangle} \qquad \to_i'\ \frac{M : \langle nil \vdash \tau \rangle}{\lambda.M : \langle nil \vdash \omega \to \tau \rangle}$$

$$\to_e\ \frac{M_1 : \langle \Gamma \vdash \wedge_{i=1}^{n} \sigma_i \to \tau \rangle \qquad M_2 : \langle \Delta^1 \vdash \sigma_1 \rangle\ \ldots\ M_2 : \langle \Delta^n \vdash \sigma_n \rangle}{(M_1\ M_2) : \langle \Gamma \wedge \Delta^1 \wedge \cdots \wedge \Delta^n \vdash \tau \rangle}$$

$$(\text{clos})\ \frac{S : \langle \Gamma \rhd \Gamma' \rangle \qquad M : \langle \Gamma' \vdash \tau \rangle}{M[S] : \langle \Gamma \vdash \tau \rangle}$$

The system $\lambda\sigma^{\wedge}$
Substitutions

$$(\wedge\text{-cons})\ \frac{M:\langle\Delta^1\vdash\sigma_1\rangle\ \ldots\ M:\langle\Delta^n\vdash\sigma_n\rangle \qquad S:\langle\Delta\triangleright\Delta'\rangle}{M.S:\langle\Delta^1\wedge\cdots\wedge\Delta^n\wedge\Delta\triangleright(\wedge_{i=1}^n\sigma_i).\Delta'\rangle}$$

$$(\mathrm{id})\ \frac{}{id:\langle\Gamma\triangleright\Gamma\rangle}\ ,\ \Gamma\neq\Delta.\omega^{\underline{m}}$$

$$(\mathrm{comp})\ \frac{S:\langle\Gamma\triangleright\Gamma''\rangle \qquad S':\langle\Gamma''\triangleright\Gamma'\rangle}{S'\circ S:\langle\Gamma\triangleright\Gamma'\rangle}$$

$$(nil\text{-shift})\ \frac{}{\uparrow:\langle nil\triangleright nil\rangle}$$

$$(nil\text{-cons})\ \frac{S:\langle\Delta\triangleright nil\rangle}{M.S:\langle\Delta\triangleright nil\rangle}$$

$$(\mathrm{shift})\ \frac{}{\uparrow:\langle\omega.\Gamma\triangleright\Gamma\rangle}\ ,\ \Gamma\neq\Delta.\omega^{\underline{n}}$$

$$(\omega\text{-cons})\ \frac{S:\langle\Delta\triangleright\Delta'\rangle}{M.S:\langle\Delta\triangleright\omega.\Delta'\rangle}\ ,\ \Delta'\neq\omega^{\underline{n}}$$

System $\lambda\sigma^{\wedge}$ properties

Theorem (SR for $\lambda\sigma^{\wedge}$)

If $M : \langle \Gamma \vdash \tau \rangle$ *and* $M \longrightarrow_{\lambda\sigma} M'$ *then* $M' : \langle \Gamma \vdash \tau \rangle$. *Particularly, if* $S : \langle \Gamma \rhd \Gamma' \rangle$ *and* $S \longrightarrow_{\lambda\sigma} S'$ *then* $S' : \langle \Gamma \rhd \Gamma' \rangle$.

Towards the characterisation of termination

1. Subject Expansion (SE)

$$\text{If } M' : \langle \Gamma \vdash \tau \rangle \text{ and } M \rightarrow_\beta M', \text{ then } M : \langle \Gamma \vdash \tau \rangle$$

   Important to prove WN characterisations in IT systems:

   "Typability for normal forms $\Longrightarrow$ typability for any WN term."

2. Typability implies WN.

$$\text{If } M : \langle \Gamma \vdash \tau \rangle \text{ then } M \text{ is WN.}$$

Towards the characterisation of termination

1. Subject Expansion (SE)

$$\text{If } M' : \langle \Gamma \vdash \tau \rangle \text{ and } M \rightarrow_{\beta} M', \text{ then } M : \langle \Gamma \vdash \tau \rangle$$

   Important to prove WN characterisations in IT systems:

   "Typability for normal forms $\Longrightarrow$ typability for any WN term."

2. Typability implies WN.

$$\text{If } M : \langle \Gamma \vdash \tau \rangle \text{ then } M \text{ is WN.}$$

Subject Expansion property

- Structural analysis proofs for both $\lambda s_e{}^{\wedge}$ and $\lambda\sigma^{\wedge}$ can be made: Generation Lemmas

Example

(Generation for operators in $\lambda s_e{}^{\wedge}$)

1. If $M\sigma^i N : \langle nil \vdash_{\lambda s_e{}^{\wedge}} \tau\rangle$, then either:
   (a) $M : \langle nil \vdash_{\lambda s_e{}^{\wedge}} \tau\rangle$
   (b) $M : \langle \omega\, \underline{i-1}.\, \wedge_{j=1}^{m} \sigma_j.nil \vdash_{\lambda s_e{}^{\wedge}} \tau\rangle$ where $\forall 1 \leq j \leq m$, $N : \langle nil \vdash_{\lambda s_e{}^{\wedge}} \sigma_j\rangle$

Subject Expansion for $\lambda s_e{}^{\wedge}$

- (σ-generation): Let $(\lambda.M\ N) \to M\sigma^1 N$ and $M\sigma^1 N : \langle \Gamma \vdash \tau \rangle$.

  If $\Gamma = nil$ then either:

(a) $\dfrac{M : \langle nil \vdash \tau \rangle}{M\sigma^1 N : \langle nil \vdash \tau \rangle}$ then

$$\dfrac{\dfrac{M : \langle nil \vdash \tau \rangle}{\lambda.M : \langle nil \vdash \omega \to \tau \rangle}}{(\lambda.M\ N) : \langle nil \vdash \tau \rangle}$$

(b) $\dfrac{M : \langle \wedge_{j=1}^{m} \sigma_j . nil \vdash \tau \rangle \qquad N : \langle nil \vdash \sigma_1 \rangle \cdots N : \langle nil \vdash \sigma_m \rangle}{M\sigma^1 N : \langle nil \vdash \tau \rangle}$ then

$$\dfrac{\dfrac{M : \langle \wedge_{j=1}^{m} \sigma_j . nil \vdash \tau \rangle}{\lambda.M : \langle nil \vdash \wedge_{j=1}^{m} \sigma_j \to \tau \rangle} \qquad N : \langle nil \vdash \sigma_1 \rangle \cdots N : \langle nil \vdash \sigma_m \rangle}{(\lambda.M\ N) : \langle nil \vdash \tau \rangle}$$

Subject Expansion for $\lambda s_e{}^{\wedge}$

- (σ-generation): Let $(\lambda.M\ N) \to M\sigma^1 N$ and $M\sigma^1 N : \langle \Gamma \vdash \tau \rangle$.

  If $\Gamma = nil$ then either:

  (a) $\dfrac{M : \langle nil \vdash \tau \rangle}{M\sigma^1 N : \langle nil \vdash \tau \rangle}$ then

$$\dfrac{\dfrac{M : \langle nil \vdash \tau \rangle}{\lambda.M : \langle nil \vdash \omega \to \tau \rangle}}{(\lambda.M\ N) : \langle nil \vdash \tau \rangle}$$

  (b) $\dfrac{M : \langle \wedge_{j=1}^m \sigma_j . nil \vdash \tau \rangle \qquad N : \langle nil \vdash \sigma_1 \rangle \cdots N : \langle nil \vdash \sigma_m \rangle}{M\sigma^1 N : \langle nil \vdash \tau \rangle}$ then

$$\dfrac{\dfrac{M : \langle \wedge_{j=1}^m \sigma_j . nil \vdash \tau \rangle}{\lambda.M : \langle nil \vdash \wedge_{j=1}^m \sigma_j \to \tau \rangle} \qquad N : \langle nil \vdash \sigma_1 \rangle \cdots N : \langle nil \vdash \sigma_m \rangle}{(\lambda.M\ N) : \langle nil \vdash \tau \rangle}$$

Subject Expansion for $\lambda s_e{}^{\wedge}$

- (σ-generation): Let $(\lambda.M\ N) \to M\sigma^1 N$ and $M\sigma^1 N : \langle \Gamma \vdash \tau \rangle$.

  If $\Gamma = nil$ then either:

  (a) $\dfrac{M : \langle nil \vdash \tau \rangle}{M\sigma^1 N : \langle nil \vdash \tau \rangle}$ then

$$\dfrac{\dfrac{M : \langle nil \vdash \tau \rangle}{\lambda.M : \langle nil \vdash \omega \to \tau \rangle}}{(\lambda.M\ N) : \langle nil \vdash \tau \rangle}$$

  (b) $\dfrac{M : \langle \wedge_{j=1}^{m} \sigma_j . nil \vdash \tau \rangle \qquad N : \langle nil \vdash \sigma_1 \rangle \cdots N : \langle nil \vdash \sigma_m \rangle}{M\sigma^1 N : \langle nil \vdash \tau \rangle}$ then

$$\dfrac{\dfrac{M : \langle \wedge_{j=1}^{m} \sigma_j . nil \vdash \tau \rangle}{\lambda.M : \langle nil \vdash \wedge_{j=1}^{m} \sigma_j \to \tau \rangle} \qquad N : \langle nil \vdash \sigma_1 \rangle \cdots N : \langle nil \vdash \sigma_m \rangle}{(\lambda.M\ N) : \langle nil \vdash \tau \rangle}$$

Subject Expansion property

- As for SR, we can derive SE for $\lambda^{\wedge}_{dB}$ from the property for $\lambda s_e{}^{\wedge}$:

  If $M' : \langle \Gamma \vdash_{\lambda^{\wedge}_{dB}} \tau \rangle$ and $M \to_\beta M'$ then (simulation property [KR97]):

  $$M \to^{+}_{\lambda s_e} M'$$

  hence

  $$M : \langle \Gamma \vdash_{\lambda^{\wedge}_{dB}} \tau \rangle$$

- Besides WN characterisation, allows further investigations about PT for all those systems.

WN for the simply typed ES calculi

- The WN of the simply typed λs_e was proved in [AKR07].
- The proof was inspired in the proof of the property for the simply typed $\lambda\sigma$ in [Gou98].
- The proof was adapted for the simply typed $\lambda\omega_e$, in order to prove the property for the simply typed λs_e.
- The general idea behind the proofs are translations for simply typed expressions(terms/substitutions) in each calculus into functions:

$$[\![M]\!](T_1; \cdots ; T_m) \in \Lambda^{\rightarrow}$$

A quasi-order is then defined:

$$M \geq N \quad \text{if} \quad \forall \overline{T},\ [\![M]\!](\overline{T}) \rightarrow^*_\beta [\![N]\!](\overline{T})$$
$$M > N \quad \text{if} \quad \forall \overline{T},\ [\![M]\!](\overline{T}) \rightarrow^+_\beta [\![N]\!](\overline{T})$$

WN for the IT systems

- A Church-style version was used in both simply typed cases, in order to have a unique type for some typable term.
- We can translate typing derivations for IT systems instead.
- The translation shall be working with typing derivations in the CDV type system [CDV81]
- Verify whether is possible to prove the property directly for the system $\lambda s_e{}^{\wedge}$.

IT and Principal Typings

IT systems such as [CDV80, RV84, Rocca88, Bakel95] have PT.

The expansion is one of the central syntactical notions related with PT for IT:

$$\frac{x:\alpha\to\beta \qquad y:\alpha}{(x\ y):\beta} \quad\Rightarrow\quad \left\{ \begin{array}{c} \dfrac{\dfrac{x:\alpha_1\to\beta_1 \qquad y:\alpha_1}{(x\ y):\beta_1} \qquad \dfrac{x:\alpha_2\to\beta_2 \qquad y:\alpha_2}{(x\ y):\beta_2}}{(x\ y):\beta_1\cap\beta_2} \\ \\ \dfrac{x:\alpha_1\cap\alpha_2\to\beta \qquad \dfrac{y:\alpha_1 \qquad y:\alpha_2}{y:\alpha_1\cap\alpha_2}}{(x\ y):\beta} \end{array} \right.$$

The System $\mathbb{I}$ [KW04] has a typing inference based on substitution (expansion variables).

Conclusions

- Explicit substitution calculi are important for implementations based on the λ-calculus

- Intersection type systems treat polymorphism in a "machine friendly" way

- We've been investigating IT system for two ES calculi: $\lambda\sigma$ and λs_e.

- The IT systems proposed may give the first characterisation of termination in each calculus.

- Further investigation about the notion for PT on $\lambda^{\wedge}_{dB}$, $\lambda\sigma^{\wedge}$ and $\lambda s_e{}^{\wedge}$ is needed.

- There is no PT notion defined for other ES calculi with IT systems such as $\lambda\mathrm{x}$ and $\lambda\mathrm{e}x$.

	$\lambda_{dB}^{\rightarrow}$	λ_{dB}^{SM}	$\lambda_{dB}^{\wedge}$
SR	yes	yes* [VAK10b]	yes
SE	no	?	yes
PT	yes [VAK08]	yes (β-nfs)[VAK10]	?
SN	yes	?	?
WN	yes	?	?

	$\lambda s^{\rightarrow}$	λs^{SM}	$\lambda s_e{}^{\rightarrow}$	$\lambda s_e{}^{\wedge}$
SR	yes	yes* [VAK10b]	yes	yes [VAK10b]
SE	no	?	no	yes
PT	yes [VAK08]	?	yes [VAK08]	?
SN	yes	?	no [Guillaume2000]	no
WN	yes	?	yes [AKR07]	?

	$\lambda\sigma^{\rightarrow}$	$\lambda\sigma^{\wedge}$	$\lambda\mathrm{x}$	λex
SR	yes	yes [VAK10b]	yes [LLDDvB2004]	?
SE	no	yes	yes [LLDDvB2004]	?
PT	yes [VAK08]	?	?	?
SN	no [Mellies95]	no	yes [LLDDvB2004]	yes [Kesner09]
WN	yes [Gou98]	?	yes [LLDDvB2004]	?

References

M. Abadi, L. Cardelli, P.-L. Curien, and J.-J. Lévy.
Explicit Substitutions.
J. of Func. Programming, 1(4):375–416, 1991.

M. Ayala-Rincón and F. Kamareddine.
Unification via the λs_e-Style of Explicit Substitution.
Logical journal of the IGPL, 9(4):489–523, 2001.

A. Arbiser, F. Kamareddine and A. Ríos.
The Weak Normalization of the Simply Typed lambda-s_e-calculus,
Logical J. of the IGPL, 15(2):121–147, 2007.

S. van Bakel.
Intersection Type Assignment Systems.
Theoret. comput. sci., 151:385-435, 1995.

R. Bloo and K. Rose.
Preservation of strong normalization in named lambda calculi with explicit substitution and garbage collection.
In *CSN-95: Computer Science in the Netherlands*, 62–72, 1995

A. Church.
A set of postulates for the foundation of logic,
Annals of Math 33(2):346–366, 1932.

A. Church.
A Formulation of the Simple Theory of Types,
J. Symbolic Logic 5(2):56–68, 1940.

References

M. Coppo and M. Dezani-Ciancaglini.
A new type assignment for lambda-terms.
Archiv für mathematische logik, 19:139–156, 1978.

M. Coppo and M. Dezani-Ciancaglini.
An Extension of the Basic Functionality Theory for the λ-Calculus.
Notre Dame Journal of Formal Logic, 21(4):685–693, 1980.

M. Coppo, M. Dezani-Ciancaglini and B. Venneri.
Principal Type Schemes and λ-calculus Semantics.
In J.P. Seldin and J.R. Hindley (eds), *To H.B. Curry: Essays on combinatory logic, lambda calculus and formalism*, pp. 536–560. Academic Press, 1980.

M. Coppo, M. Dezani-Ciancaglini, and B. Venneri.
Functional Characters of Solvable Terms..
Mathematical Logic Quarterly 27, pp. 45–58, 1981.

L. Damas and R. Milner.
Principal Type-Schemes for Functional Programs.
ACM Symposium on Principles of Programming Languages (POPL'82), 207–212, 1982. ACM Press.

N.G. de Bruijn.
Lambda-Calculus Notation with Nameless Dummies, a Tool for Automatic Formula Manipulation, with Application to the Church-Rosser Theorem.
Indag. Mat., 34(5):381–392, 1972.

G. Dowek, T. Hardin, and C. Kirchner.
Higher-order unification via explicit substitutions, extended abstract.
Proceedings of LICS'95 366–374, 1995.

J. Goubault-Larrecq.
A Proof of Weak Termination of Typed $\lambda\sigma$-Calculi.
In *Selected Papers from TYPES'96*, LNCS 1512:134–151. Springer, 1998.

References

B. Guillaume.
The λse-calculus does not preserve strong normalisation.
J. of func. program., 10(4):321–325, 2000.

J. R. Hindley.
Basic Simple Type Theory.
Cambridge Tracts in Theoretical Computer Science, 42, 1997. Cambridge University Press.

T. Jim.
What are principal typings and what are they good for?
ACM Symposium on Principles of Programming Languages (POPL'96), 42–53, 1996. ACM Press.

F. Kamareddine, editor.
Thirty Five Years of Automating Mathematics.
Kluwer, 2003.

F. Kamareddine and A. Ríos.
A λ-calculus à la de Bruijn with Explicit Substitutions.
In *Proc. of PLILP'95*, *LNCS* 982:45–62. Springer, 1995.

F. Kamareddine and A. Ríos.
Extending a λ-calculus with Explicit Substitution which Preserves Strong Normalisation into a Confluent Calculus on Open Terms.
J. of Func. Programming, 7:395–420, 1997.

F. Kamareddine and A. Ríos.
Relating the $\lambda\sigma$- and λs-styles of explicit substitutions.
J. of Logic and Comp., 10(3):349–380, 2000.

References

D. Kesner.
A Theory of Explicit Substitutions with Safe and Full Composition.
Logical Methods in Computer Science, 5(3):1–29, 2009.

A. J. Kfoury and J. B. Wells.
Principality and type inference for intersection types using expansion variables.
Theor. Comput. Sci., 311(1–3):1–0. 2004.

S. Lengrand, P. Lescanne, D. Dougherty, M. Dezani-Ciancaglini, and S. van Bakel.
Intersection types for explicit substitutions.
Information and Computation, 189(1):1742, 2004.

R. Lins.
A new formula for the execution of categorical combinators.
In *Siekmann, J.H.(ed.) 8th International Conference on Automated Deduction.* LNCS, vol. 230, Springer, Heidelberg, 1986.

P.-A. Melliès.
Typed Lambda-Calculi with Explicit Substitutions may not terminate.
In *Proc. of TLCA'95*, pages 328–334. Springer, 1995.

C. Muñoz.
A Left-linear Variant of $\lambda\sigma$
In *Proc. of PLILP/ALP/HOA'97*, LNCS 1298:224–239. Springer, 1997.

G. Pottinger.
A type assignment for the strongly normalizable λ-terms.
In J.P. Seldin and J. R. Hindley (eds), *To H. B. Curry: Essays on combinatory logic, lambda calculus and formalism*, pp. 561–578. Academic Press, 1980.

References

A. Ríos.
Contributions à l'étude des λ-calculs avec des substitutions explicites
Thèse de Doctorat d'Université , Université Paris VII, 1993.

E. Ritter.
Characterising explicit substitutions which preserve termination.
In *J.-Y. Girard(ed), TLCA99*, LNCS, 1581:325-339. Springer-Verlag, 1999.

S. Ronchi Della Rocca and B. Venneri.
Principal Type Scheme for an Extended Type Theory.
Theoret. comput. sci., 28:151–169, 1984.

S. Ronchi Della Rocca.
Principal Type Scheme and Unification for Intersection Type Discipline.
Theoret. comput. sci., 59:181–209, 1988.

E. Sayag and M. Mauny.
Characterization of principal type of normal forms in intersetion type system.
In *Proc. of FSTTCS'96*, LNCS, 1180:335–346. Springer, 1996.

E. Sayag and M. Mauny.
A new presentation of the intersection type discipline through principal typings of normal forms.
Tech. rep. RR-2998, INRIA, 1996.

J. Wells.
The essence of principal typings.
In *Proc.ICALP 2002, LNCS*, 2380:913–925. Springer Verlag, 2002.

References

D. Ventura, M. Ayala and F. Kamareddine.
Principal Typing for Explicit Substitutions Calculi.
In *Proc. Logic and Theory of Algorithms, 4th Conference on Computability in Europe (CiE 2008)*, LNCS, Vol. 5028, pages 548–558, 2008.

D. Ventura and M. Ayala-Rincón and F. Kamareddine.
Intersection Type System with de Bruijn Indices.
The many sides of logic. Studies in logic 21:557–576, College publications. London, 2009.

D. Ventura and M. Ayala-Rincón and F. Kamareddine.
Principal Typings in a Restricted Intersection Type System for Beta Normal Forms with de Bruijn Indices.
In *Proc. of WRS'09*. EPTCS 15:69–82, 2010.

D. Ventura and M. Ayala-Rincón and F. Kamareddine.
Intersection Type Systems and Explicit Substitutions Calculi.
In *Proc. of WoLLIC'10*. LNCS (FoLLI-LNAI subseries), 6188:232–246. Berlin Heidelberg: Springer-Verlag, 2010.

The λx-calculus rewriting rules [Lins86, BR95]

$$x\langle x := N\rangle \longrightarrow N \tag{xv}$$

$$x\langle y := N\rangle \longrightarrow x \quad \text{if } x \not\equiv y \tag{xvgc}$$

$$(\lambda_x.M)\langle y := N\rangle \longrightarrow \lambda_x.M\langle y := N\rangle \tag{xab}$$

$$(M_1\ M_2)\langle y := N\rangle \longrightarrow (M_1\langle y := N\rangle\ M_2\langle y := N\rangle) \tag{xap}$$

where $x \not\equiv y$ and $x \notin FV(N)$ in (xab)