Execution Time of Lambda-Terms via Non Uniform Semantics and Intersection Types

Daniel de Carvalho

Institut de Mathématiques de Luminy

Definition 1 *A λ-algebra is of the form $(\mathcal{D}, \cdot, \mathcal{I})$ such that $(\mathcal{D}, \cdot)$ is a magma and $\mathcal{I}$ is a map from $\Lambda(\mathcal{D}) \times \mathcal{D}^{\mathcal{V}}$ to $\mathcal{D}$ such that :*

- *for $x \in \mathcal{V}$ and $\rho \in \mathcal{D}^{\mathcal{V}}$, $\mathcal{I}(x, \rho) = \rho(x)$;*
- *for $d \in \mathcal{D}$ and $\rho \in \mathcal{D}^{\mathcal{V}}$, $\mathcal{I}(c_d, \rho) = d$;*
- *for $v, u \in \Lambda(\mathcal{D})$ and $\rho \in \mathcal{D}^{\mathcal{V}}$, $\mathcal{I}((v)u, \rho) = \mathcal{I}(v, \rho) \cdot \mathcal{I}(u, \rho)$;*
- *for $x \in \mathcal{V}$, $v \in \Lambda(\mathcal{D})$, $d \in \mathcal{D}$, $\mathcal{I}(\lambda x.v, \rho) \cdot d = \mathcal{I}(v, \rho[x := d])$;*
- *for $v \in \Lambda(\mathcal{D})$ and $\rho_1, \rho_2 \in \mathcal{D}^{\mathcal{V}}$ such that $\rho_{1|FV(v)} = \rho_{2|FV(v)}$, $\mathcal{I}(v, \rho_1) = \mathcal{I}(v, \rho_2)$;*
- *for $v, u \in \Lambda(\mathcal{D})$, we have $(v =_\beta u \Rightarrow \forall \rho \in \mathcal{D}^{\mathcal{V}}\ \mathcal{I}(v, \rho) = \mathcal{I}(u, \rho))$.*

Definition 2 *A context* Γ *is a function from* $\mathcal{V}$ *to* $\mathcal{M}_f(D)$ *such that* $\{x \in \mathcal{V}\ ;\ \Gamma(x) \neq []\}$ *is finite. If* $x_1, \ldots, x_n \in \mathcal{V}$ *are distinct and* $a_1, \ldots, a_n \in \mathcal{M}_f(D)$*, then* $x_1 : a_1, \ldots, x_n : a_n$ *denotes the context* $\{(x_i, a_i)\ ;\ 1 \leq i \leq n\} \cup \{(y, [])\ ;\ y \in \mathcal{V} \setminus \{x_1, \ldots, x_n\}\}$.

Definition 3 *The typing rules of System* R *are the following :*

$$\frac{}{x : [\alpha] \vdash_R x : \alpha}$$

$$\frac{\Gamma, x : a \vdash_R v : \alpha}{\Gamma \vdash_R \lambda x.v : i(a, \alpha)}$$

$$\frac{\Gamma_0 \vdash_R v : i([\alpha_1, \ldots, \alpha_n], \alpha) \qquad \Gamma_1 \vdash_R u : \alpha_1, \ldots, \Gamma_n \vdash u : \alpha_n}{\Gamma_0 + \Gamma_1 + \ldots + \Gamma_n \vdash_R (v)u : \alpha}$$

Relating semantics and types

Theorem 1 *For any closed term t, we have $[\![t]\!] = \{\alpha \in D\ ;\ \vdash_R t : \alpha\}$.*

Some properties of System R

Theorem 2 *For $t \in \Lambda$, t is normalizable if, and only if, there exist $\alpha \in D$ in which $[]$ has only negative occurrences and $\Gamma \in \Phi$ in which $[]$ has only positive occurrences such that $\Gamma \vdash_R t : \alpha$.*

Theorem 3 *For $t \in \Lambda$, t is head-normalizable if, and only if, t is typable in System R.*

Principal typing property

Definition 4 *Principal typing of normal terms :*

$$\frac{}{x:[\gamma]\vdash_P x:\gamma}\ \gamma\in A$$

$$\frac{\Gamma, x:a\vdash_P t:\alpha}{\Gamma\vdash_P \lambda x.t:a\alpha}$$

$$\frac{\Gamma_1\vdash_P u_1:\alpha_1 \quad \ldots \quad \Gamma_n\vdash_P u_n:\alpha_n}{\sum_{i=1}^{n}\Gamma_i+\{(x,[[\alpha_1]\ldots[\alpha_n]\gamma])\}\vdash_P (x)u_1\ldots u_n:\gamma}\ (*)$$

$(*)$ *the* Γ_i *are disjoint and* $\gamma\in A$ *does not appear in* Γ_i

Theorem 4 *For any normal term* t*, for any* (Δ,β) *such that* $\Delta\vdash_P t:\beta$*, we have* $\Gamma\vdash_R t:\alpha$ *if, and only if, there exist* (Γ'',α'') *and a substitution* r *such that* $(\Delta,\beta)\stackrel{*}{\rightarrow}(\Gamma'',\alpha'')$ *and* $r(\Gamma'',\alpha'')=(\Gamma,\alpha)$.

A machine computing a head-normal form

Definition 5 *For $t, t' \in \Lambda(\mathbb{S})$, we define, by induction on t, $t \succ t'$, where t respects the variable convention :*

- *if $t \in \mathcal{V}$, then $t \succ t'$ is false for any t' ;*
- *if $t = ((v, e), \pi) \in \mathbb{S}$, then :*
  - *if $v \in \mathcal{V}$, then :*
    - *if $v \in \mathrm{dom}(e)$, then $t' = (e(v), \pi)$;*
    - *else, if $\pi = ((v_1, e_1), \ldots, (v_q, e_q))$, then $t' = (x)v_1[e_1] \ldots v_q[e_q]$;*
  - *if $v = \lambda x.u$, then :*
    - *if π is the empty sequence, then $t' = \lambda x.((u, e), \pi)$;*
    - *if $\pi = (c, \pi')$, then $t' = ((u, \{(x, c)\} \cup e), \pi')$.*
  - *if $v = (v_2)v_1$, then $t' = ((v_2, e), ((v_1, e), \pi))$;*
- *if $t = (v)u$, then $t' = (v')u$ avec $v \succ v'$;*
- *if $t = \lambda x.v$, then $t' = \lambda x.v'$ with $v \succ v'$.*

Theorem 5 *For* $t \in \Lambda$*, we have* $l((t,\emptyset),\emptyset) = inf\{|\Pi| \; ; \; \exists(\Gamma,\alpha)\ \Pi \in \Delta(t,(\Gamma,\alpha))\}$.

Theorem 6 *For any closed normal terms* v *and* u*, for any* a,α *such that* $(a,\alpha) \in [\![v]\!]$ *and* $Supp(a) \subseteq [\![u]\!]$*, we have* $l(((v)u,\emptyset),\emptyset) \leq 2|a| + |\alpha| + 1$.

A machine computing the normal form

Definition 6 *For $t, t' \in \Lambda(\mathbb{S})$, we define, by induction on t, $t \succ' t'$, where t respects the variable convention :*

- *if $t \in \mathcal{V}$, then $t \succ' t'$ is false for any t' ;*
- *if $t = ((v, e), \pi) \in \mathbb{S}$, then :*
  - *if $v \in \mathcal{V}$, then :*
    - *if $v \in \mathrm{dom}(e)$, then $t' = (e(v), \pi)$;*
    - *else, if $\pi = (c_1, \ldots, c_q)$, then $t' = (v)(c_1, \emptyset) \ldots (c_q, \emptyset)$;*
  - *if $v = \lambda x.u$, then :*
    - *if π is the empty sequence, then $t' = \lambda x.((u, e), \pi)$;*
    - *if $\pi = (c, \pi')$, alors $t' = ((u, \{(x, c)\} \cup e), \pi')$.*
  - *if $v = (v_2)v_1$, then $t' = ((v_2, e), ((v_1, e), \pi))$;*
- *if $t = (v)u$, then $(t' = (v')u$ avec $v \succ' v')$ or $(t' = (v)u'$ with $u \succ' u'$ and $v \in \Lambda)$;*
- *if $t = \lambda x.v$, then $t' = \lambda x.v'$ with $v \succ' v'$.*

Theorem 7 *For any* $s = ((t, e), (c_1, \ldots, c_q)) \in \mathbb{S}$ *such that* $(t[e])\overline{c_1} \ldots \overline{c_q}$ *is normalizable, for any* (Γ, α)*, for any derivation of* $\Gamma \vdash s : \alpha$ *without empty multisets in axioms, we have* $l'(s) \leq |\Pi|$.

Theorem 8 *For any normalizable term* t*, for any principal typing* (Γ, α) *of its normal form, for any derivation* Π *of* $\Gamma \vdash_R t : \alpha$*, we have* $l'((t, \emptyset), \emptyset) = |\Pi|$.