A newborn in the $\lambda\delta$ family: introducing $\lambda\delta$-2B

Ferruccio Guidi

DISI, University of Bologna, Bologna, Italy

ferruccio.guidi@unibo.it

November 14, 2018

1. The $\lambda\delta$ family: design requirements

We designed the systems of the $\lambda\delta$ family to meet the next features:

- predicative higher-order abstraction (de Bruijn unified binder λ);
- telescopic explicit substitution in terms (abbreviation δ);
- type checking reduces to validation (type cast ©);
- valid terms are typed (infinite type sequences are possible);
- small-step conversion (context-sensitive conversion);
- type construction and type conversion separated in type inference.
- infinite levels of terms (no: T or K are sorts if $\Gamma \vdash M : T : K$);
- relaxed typing (no: Γ is valid if $\Gamma \vdash M : T$);
- desirable invariants hold (confluence, normalization, preservation);
- the invariants are formally specified and machine-checked.

Our systems are outside both the Automath family and the PTS family.

2. Defining $\lambda\delta$-2B (1 of 3)

Morphology & syntax. Alphabet: ( ) . $\star$ # © @ λ δ Λ Δ [integers: s, i].

Terms: $T, U, V, W \coloneqq \star s \mid \#i \mid (\lambda W).T \mid (\delta V).T \mid (@V).T \mid (©U).T$

Local environments: $L, K \coloneqq \star \mid K.(\Lambda W) \mid K.(\Delta V)$

Reduction steps & type inference steps. The function $\uparrow_h$ is a parameter.

(theta)	$L \vdash (@V).(\delta W).T \mapsto_\theta (\delta W).(@\Uparrow^1 V).T$
(beta)	$L \vdash (@V).(\lambda W).T \mapsto_\beta (\delta(©W).V).T$
(delta)	$K.(\Delta W) \vdash \#1 \mapsto_\delta \Uparrow^1 W$
(zeta)	$L \vdash (\delta W).\Uparrow^1 T \mapsto_\zeta T$
(epsilon)	$L \vdash (©U).T \mapsto_\epsilon T$
(ess)	$L \vdash \star s \mapsto_s \star\uparrow_h s$ [applies endlessly]
(ell)	$K.(\Lambda W) \vdash \#1 \mapsto_l \Uparrow^1 W$
(ee)	$L \vdash (©U).T \mapsto_e U$

3. Defining $\lambda\delta$-2B (2 of 3)

Parallel rt-transition with n t-steps. $L \vdash T_1 \Rightarrow_n T_2$ [depends on h].

$$\frac{}{L \vdash \star|\#i \Rightarrow_0 \star|\#i}\ 1R \qquad \frac{K \vdash \#i \Rightarrow_n T}{K.(\Lambda|\Delta W) \vdash \#{\uparrow}i \Rightarrow_n \Uparrow^1 T}\ 1L$$

$$\frac{K \vdash W_1 \Rightarrow_0 W_2 \quad K.(\Lambda|\Delta W_1) \vdash T_1 \Rightarrow_n T_2}{K \vdash (\lambda|\delta W_1).T_1 \Rightarrow_n (\lambda|\delta W_2).T_2}\ 1B$$

$$\frac{L \vdash V_1 \Rightarrow_0 V_2 \quad L \vdash T_1 \Rightarrow_n T_2}{L \vdash (@V_1).T_1 \Rightarrow_n (@V_2).T_2}\ 1A \qquad \frac{L \vdash U_1 \Rightarrow_n U_2 \quad L \vdash T_1 \Rightarrow_n T_2}{L \vdash (©U_1).T_1 \Rightarrow_n (@U_2).T_2}\ 1K$$

$$\frac{L \vdash V_1 \Rightarrow_0 V_2, L \vdash W_1 \Rightarrow_0 W_2, L \vdash T_1 \Rightarrow_n T_2}{L \vdash (@V_1).(\lambda W_1).T_1 \Rightarrow_n (\delta(©W_2).V_2).T_2}\ 1\beta \qquad \frac{L \vdash V_1 \Rightarrow_0 V_2, L \vdash W_1 \Rightarrow_0 W_2, L \vdash T_1 \Rightarrow_n T_2}{L \vdash (@V_1).(\delta W_1).T_1 \Rightarrow_n (\delta W_2).(@\Uparrow^1 V_2).T_2}\ 1\theta$$

$$\frac{K \vdash W_1 \Rightarrow_n W_2}{K.(\Delta W_1) \vdash \#1 \Rightarrow_n \Uparrow^1 W_2}\ 1\delta \qquad \frac{K \vdash T_1 \Rightarrow_n T_2}{K \vdash (\delta W).\Uparrow^1 T_1 \Rightarrow_n T_2}\ 1\zeta \qquad \frac{K \vdash T_1 \Rightarrow_n T_2}{K \vdash (©U).T_1 \Rightarrow_n T_2}\ 1\epsilon$$

$$\frac{}{L \vdash \star s \Rightarrow_1 \star{\uparrow_h} s}\ 1s \qquad \frac{K \vdash W_1 \Rightarrow_n W_2}{K.(\Lambda W_1) \vdash \#1 \Rightarrow_{\uparrow n} \Uparrow^1 W_2}\ 1l \qquad \frac{K \vdash U_1 \Rightarrow_n U_2}{K \vdash (©U_1).T \Rightarrow_{\uparrow n} U_2}\ 1e$$

This is fully parallel and substitution is linear in a weak head transition.

Parallel rt-computation with n t-steps. $L \vdash T_1 \Rightarrow_n^* T_2$ [trans. closure].

$$\frac{L \vdash T_1 \Rightarrow_n T_2}{L \vdash T_1 \Rightarrow_n^* T_2}\ 2R \qquad \frac{L \vdash T_1 \Rightarrow_{n_1}^* T \quad L \vdash T \Rightarrow_{n_2}^* T_2}{L \vdash T_1 \Rightarrow_{n_1+n_2}^* T_2}\ 2T$$

4. Defining $\lambda\delta$-2B (3 of 3)

Extended validity (Automath-like) $L \vdash T\ !^*$ [depends on h].

$$\frac{}{L \vdash \star s\ !^*}\ 3S \qquad \frac{K \vdash W\ !^*}{K.(\Lambda|\Delta W) \vdash \#1\ !^*}\ 3U \qquad \frac{K \vdash \#i\ !^*}{K.(\Lambda|\Delta W) \vdash \#\uparrow i\ !^*}\ 3L$$

$$\frac{K \vdash W\ !^* \qquad K.(\Lambda|\Delta W) \vdash T\ !^*}{K \vdash (\lambda|\delta W).T\ !^*}\ 3B$$

$$\frac{L \vdash W\ !^* \qquad L \vdash T\ !^* \qquad L \vdash W \Rightarrow_0^* U \qquad L \vdash T \Rightarrow_1^* U}{L \vdash (\copyright W).T\ !^*}\ 3K$$

$$\frac{L \vdash V\ !^* \qquad L \vdash T\ !^* \qquad L \vdash V \Rightarrow_1^* W \qquad L \vdash T \Rightarrow_n^* (\lambda W).U}{L \vdash (@V).T\ !^*}\ 3A$$

Notice that in rule $3A$ we can choose the value of n at will (0 is OK).

Extended type $L \vdash T\ :^*\ U$ by def. $L \vdash (\copyright U).T\ !^*$ [depends on h].

Restricted validity (PTS-like) $L \vdash T\ !$ [depends on h].

Rules 4 are like rules 3 above except for rule $4A$ that is as follows:

$$\frac{L \vdash V\ ! \qquad L \vdash T\ ! \qquad L \vdash V \Rightarrow_1^* W \qquad L \vdash T \Rightarrow_1^* (\lambda W).U}{L \vdash (@V).T\ !}\ 4A$$

Restricted type $L \vdash T\ :\ U$ by def. $L \vdash (\copyright U).T\ !$ [depends on h].

5. Arity assignment and strong normalization

Strong normalization holds for unbound rt-computation $L \vdash T_1 \Rightarrow^* T_2$, which we define like $L \vdash T_1 \Rightarrow_n^* T_2$ without the bound n on all its rules.

Due to rule $1s$, we must take terms up to the next equivalence relation:

$$\frac{}{\star s_1 \doteq \star s_2}\,5S \qquad \frac{}{\#i \doteq \#i}\,5R \qquad \frac{U_1 \doteq U_2 \quad T_1 \doteq T_2}{(\lambda|\delta|@|©U_1).T_1 \doteq (\lambda|\delta|@|©U_2).T_2}\,5P$$

We define inductively strongly normalizing terms with the next rule:

$$\frac{(\forall T_2)\ L \vdash T_1 \Rightarrow T_2 \supset (T_1 \doteq T_2 \supset \bot) \supset L \vdash \Rightarrow^*\mathbf{SN}(T_2)}{L \vdash \Rightarrow^*\mathbf{SN}(T_1)}\,\textit{rt-sn}$$

Strong normalization follows from: $L \vdash T\ !^*$ implies $(\exists A)\ L \vdash T : A$ according to the next simple type assignment with one base type $\star$.

$$\frac{}{L \vdash \star s : \star}6S \quad \frac{K \vdash \#i : A}{K.(\Lambda|\Delta W) \vdash \#{\uparrow}i : A}6L \quad \frac{K \vdash W : B}{K.(\Lambda|\Delta W) \vdash \#1 : B}6U \quad \frac{L \vdash V : B,\ L \vdash T : B \to A}{L \vdash (@V).T : A}6A$$

$$\frac{K \vdash W : B,\ K.(\Lambda W) \vdash T : A}{K \vdash (\lambda W).T : B \to A}6Y \quad \frac{K \vdash W : B,\ K.(\Delta W) \vdash T : A}{K \vdash (\delta W).T : A}6D \quad \frac{L \vdash W : A,\ L \vdash T : A}{L \vdash (©W).T : A}6K$$

Notice that we can set the expressive power of $\lambda\delta$-2B at the level of $\lambda\!\to$.

6. Transition in environment and big-tree theorem

Parallel r-transition in environment is: $\dfrac{}{\star \Rightarrow_0 \star}\,7S \quad \dfrac{K_1 \Rightarrow_0 K_2 \quad K_1 \vdash W_1 \Rightarrow_0 W_2}{K_1.(\Lambda|\Delta W_1) \Rightarrow_0 K_2.(\Lambda|\Delta W_2)}\,7B$

Structural induction on a closure $[L, T]$ relies on $\left[L_1, T_1\right] \sqsupset \left[L_2, T_2\right]$ (s-step), whose transitive closure $\left[L_1, T_1\right] \sqsupset^+ \left[L_2, T_2\right]$ is well founded.

$$\dfrac{}{[K.(\Lambda|\Delta W), \#1] \sqsupset [K, W]}\,8U \quad \dfrac{}{\left[K.(\Lambda|\Delta W), \Uparrow^1 T\right] \sqsupset [K, T]}\,8L \quad \dfrac{}{[L, (@|©V).T] \sqsupset [L, T]}\,8F$$

$$\dfrac{}{[L, (\lambda|\delta|@|©V).T] \sqsupset [L, V]}\,8P \quad \dfrac{}{[L, (\lambda|\delta W).T] \sqsupset [L.(\Lambda|\Delta W), T]}\,8B$$

Big-tree induction on a valid closure relies on $\left[L_1, T_1\right] \succ \left[L_2, T_2\right]$ (rst-step), whose transitive closure $\left[L_1, T_1\right] > \left[L_2, T_2\right]$ is well founded.

$$\dfrac{\left[L_1, T_1\right] \sqsupset \left[L_2, T_2\right]}{\left[L_1, T_1\right] \succ \left[L_2, T_2\right]}\,9R1 \quad \dfrac{L \vdash T_1 \Rightarrow T_2 \quad T_1 \stackrel{\star}{=} T_2 \supset \bot}{\left[L, T_1\right] \succ \left[L, T_2\right]}\,9R2$$

In particular we define inductively well-founded closures with the rule:

$$\dfrac{(\forall L_2, T_2)\ \left[L_1, T_1\right] \succ \left[L_2, T_2\right] \supset {>}\mathbf{SN}(L_2, T_2)}{{>}\mathbf{SN}(L_1, T_1)}\,rst\text{-}sn$$

An induction principle originates from the big-tree theorem stating that:

$$L \vdash T \,\vdots\, A \text{ (and thus } L \vdash T\ !^*) \text{ implies } {>}\mathbf{SN}(L, T).$$

7. Confluence and preservation

With the next rules we define the building blocks for the confluence of rt-computation and the preservation of validity through rt-computation:

$$\frac{L_0 \vdash T_0 \Rightarrow_0 T_1 \quad L_0 \vdash T_0 \Rightarrow_0 T_2 \quad L_0 \Rightarrow_0 L_1 \quad L_0 \Rightarrow_0 L_2}{(\exists T)\ L_1 \vdash T_1 \Rightarrow_0 T\ \&\ L_2 \vdash T_2 \Rightarrow_0 T}\ \mathbf{D}(L_0, T_0)$$

$$\frac{L_0 \vdash T_0\ !^* \quad L_0 \vdash T_0 \Rightarrow_{n_1} T_1 \quad L_0 \vdash T_0 \Rightarrow_{n_2} T_2 \quad L_0 \Rightarrow_0 L_1 \quad L_0 \Rightarrow_0 L_2}{(\exists T)\ L_1 \vdash T_1 \Rightarrow^*_{n_2 - n_1} T\ \&\ L_2 \vdash T_2 \Rightarrow^*_{n_1 - n_2} T}\ \mathbf{K}(L_0, T_0)$$

$$\frac{L_0 \vdash T_0\ !^* \quad L_0 \vdash T_0 \Rightarrow^*_{n_1} T_1 \quad L_0 \vdash T_0 \Rightarrow^*_{n_2} T_2 \quad L_0 \Rightarrow_0 L_1 \quad L_0 \Rightarrow_0 L_2}{(\exists T)\ L_1 \vdash T_1 \Rightarrow^*_{n_2 - n_1} T\ \&\ L_2 \vdash T_2 \Rightarrow^*_{n_1 - n_2} T}\ \mathbf{C}(L_0, T_0)$$

$$\frac{L_0 \vdash T_0\ !^* \quad L_0 \vdash T_0 \Rightarrow_n T_1 \quad L_0 \Rightarrow_0 L_1}{L_1 \vdash T_1\ !^*}\ \mathbf{P}(L_0, T_0)$$

Validity makes the pair (ϵ, e) confluent producing the kite $\mathbf{K}(L_0, T_0)$.

The rules $\mathbf{C}(L_0, T_0)$, $\mathbf{P}(L_0, T_0)$ are mutually dependent and we have:

$$\frac{(\forall L_0, T_0)\ [L, T] \sqsupset^+ \left[L_0, T_0\right] \supset \mathbf{D}(L_0, T_0)}{\mathbf{D}(L, T)}\ \mathrm{Th1}(L, T)$$

$$\frac{(\forall L_0, T_0)\ [L, T] > \left[L_0, T_0\right] \supset \mathbf{C}(L_0, T_0)\ \&\ \mathbf{P}(L_0, T_0)}{\mathbf{K}(L, T)\ \&\ \mathbf{C}(L, T)\ \&\ \mathbf{P}(L, T)}\ \mathrm{Th2}(L, T)$$

$\mathbf{D}(L, T)$ is immediate, the big-tree induction yields $\mathbf{C}(L, T)$ & $\mathbf{P}(L, T)$.

8. Convertibility and derived type rules

We define contextual convertibility $L \vdash U_1 \Leftrightarrow^*_{0,0} U_2$ with the next rules:

$$\frac{L \vdash U_1 \Rightarrow_0 U_2}{L \vdash U_1 \Leftrightarrow^*_{0,0} U_2}\ 10R \qquad \frac{L \vdash U_2 \Rightarrow_0 U_1}{L \vdash U_1 \Leftrightarrow^*_{0,0} U_2}\ 10X \qquad \frac{L \vdash U_1 \Leftrightarrow^*_{0,0} U \quad L \vdash U \Leftrightarrow^*_{0,0} U_2}{L \vdash U_1 \Leftrightarrow^*_{0,0} U_2}\ 10T$$

We obtain the restricted type rules from $L \vdash T\ !$ iff $(\exists U)\ L \vdash T : U$

$$\frac{K \vdash V : W}{K.(\Delta V) \vdash \#1 : \Uparrow^1 W}\ 11\Delta \qquad \frac{K \vdash W\ !}{K.(\Lambda W) \vdash \#1 : \Uparrow^1 W}\ 11\Lambda \qquad \frac{K \vdash \#i : L}{K.(\Lambda|\Delta V) \vdash \#\uparrow i : \Uparrow^1 U}\ 11L$$

$$\frac{}{L \vdash \star s : \star\uparrow_h s}\ 11S \qquad \frac{K \vdash V\ ! \quad K.(\Lambda|\Delta V) \vdash T : U}{K \vdash (\lambda|\delta V).T : (\lambda|\delta V).U}\ 11B \qquad \frac{L \vdash T : U}{L \vdash (©U).T : U}\ 11K$$

$$\frac{L \vdash V : W \quad L \vdash T : (\lambda W).U}{L \vdash (@V).T : (@V).(\lambda W).U}\ 11A \qquad \frac{L \vdash T : U_1 \quad L \vdash U_1 \Leftrightarrow^*_{0,0} U_2 \quad L \vdash U_2\ !}{L \vdash T : U_2}\ 11C$$

We obtain the extended type rules from $L \vdash T\ !^*$ iff $(\exists U)\ L \vdash T :^* U$.

Rules 12 are like rules 11 above except for rule $11A$ that is replaced by:

$$\frac{K \vdash V :^* W \quad K.(\Lambda W) \vdash T :^* U}{K \vdash (@V).(\lambda W).T :^* (@V).(\lambda W).U}\ 12A1 \qquad \frac{L \vdash T :^* U \quad L \vdash (@V).U\ !^*}{L \vdash (@V).T :^* (@V).U}\ 12A2$$

As of now we can confirm the next main invariants: correctness of types, uniqueness of types up to conversion, preservation of types by reduction.

9. Comments and future work

The current specification of $\lambda\delta$-2B in Matita consists of the following:

Branch	Definitions	Propositions	Loss factor
Additions to the library	148	781	2.2
Structures for the $\lambda\delta$ family	122	868	4.0
Specific structures for $\lambda\delta$-2B	41	854	4.6

We developed this specification in three years (Oct. 2015 to Nov. 2018).

W.r.t. $\lambda\delta$-2A, the present specification stands without the next notions:

- canonical typing of a term (replaced by rt-transition with one t-step);
- degree of a term (we proved preservation w/o induction on the degree).

We are working on the remaining properties of $\lambda\delta$-1A, esp. decidability; on linking the ext. and rest. type systems via formal η-conversion on λ; on $\lambda\delta$-2B denotational semantics (first step: define what a model is).

Thank you