Polytime checker for LF

Jatin Shah

February 4, 2004

We denote natural numbers by $\bar{0}, \bar{1}, \ldots$. Additionally, we assume that we have inference rules for basic arithmetic and comparison operators, in particular addition ($+$), equality ($=$) and less than ($<$).

Let $M, N, \ldots$ denote LF objects and $A, B, \ldots$ LF types.

Definition 0.1 *For an LF object M, we define $\tau(M) = a$ if M has type $aN_1N_2\ldots N_k$.*

Definition 0.2 *If a denotes a LF type family, we define a* size *function $\#_a(\cdot)$ for every canonical LF object as shown below:*

$$
\begin{array}{rcll}
\#_a(M) & = & \bar{0} & \textit{if } \tau(M) \prec a \\
\textit{Otherwise,} & & & \\
\#_a(c) & = & \bar{1} & \textit{if } \tau(c) \succeq a \textit{ for an LF object constant } c \\
\#_a(x) & = & \bar{1} & \textit{if } \tau(x) \succeq a \textit{ for an LF object variable } x \\
\#_a(M_1M_2) & = & \left\{ \begin{array}{ll} \#_a(M_1) & \textit{if } \tau(M_1) \succ \tau(M_2) \\ \infty & \textit{otherwise} \end{array} \right. & \textit{if } M_1 \textit{ is of the form } \lambda x.M_1' \\
\#_a(M_1M_2) & = & \#_a(M_1) + \#_b(M_2) & \textit{if } M_1 \textit{ is not of the form } \lambda x.M_1',\ \tau(M_2) = bN_1N_2\ldots N_{k_1} \\
 & & & \textit{and } \tau(M_1) = bN_1N_2\ldots N_{k_1} \to aP_1P_2\ldots P_{k_2} \\
\#_a(M_1M_2) & = & \#_a(M_1) + \#_a(M_2) & \textit{if } M_1 \textit{ is not of the form } \lambda x.M_1',\ \tau(M_2) = bN_1N_2\ldots N_{k_1} \\
 & & & \textit{and } \tau(M_1) = \Pi x : bN_1N_2\ldots N_{k_1}.aP_1P_2\ldots P_{k_2} \\
\#_a(\lambda x.M) & = & \#_a(M) &
\end{array}
$$

The Horn fragment of LF is given as:

$$
\begin{array}{lcl}
D & ::= & H \mid \Pi x : A.D \mid G \to D \\
G & ::= & H \\
N & ::= & \#_a(M) \mid N_1 + N_2 \mid \bar{n} \mid \infty
\end{array}
$$

where H denotes LF types and N are natural numbers.

1 Polytime checker

A judgment in the polytime checker is given by $\Gamma/\Delta \vdash_a D/N$, where Γ and Δ are LF type contexts, a is LF type family, N is a natural number and D is a LF type.

Definition 1.1 *We define head of a LF type $\Pi x_1 : A_1.\Pi x_2 : A_2.\ldots\Pi x_n : A_n.aM_1M_2\ldots M_n$ as $head(\Pi x_1 : A_1.\Pi x_2 : A_2.\ldots\Pi x_n : A_n.aM_1M_2\ldots M_n) = a$*

For a type family b, let $\Sigma_b \subseteq \Sigma$ be all objects $c : D \in \Sigma$ such that $head(D) = b$.

[[[Is our distinction between Π and $\to$ type accurate? After all, everything is a Π type]]] **[[[Exactly how is the type subordination used in poly-time checker? Does it correspond to unary/binary encodings? I do not seem to need it for the proof]]]**

Definition 1.2 *[Ver97, p. 576-577] Let function g be given. We say that function f is g-star-shaped iff for all $x \geq 1$ and $0 < t < 1$, $f(tx) \leq g(t)f(x)$. We say that f is g-co-star-shaped iff for all x and $0 < t < 1$, $f(tx) \geq g(t)f(x)$*

$$\frac{}{\vdash \cdot \ \mathsf{poly}}\ poly_z \qquad \frac{\vdash \Sigma\ \mathsf{poly} \quad \cdot/\cdot \vdash_{head(D)} D/\bar{0}}{\vdash \Sigma, c : D\ \mathsf{poly}}\ poly_s$$

$$\frac{\Gamma/\Delta \vdash a : A \to B \to Type \quad \Gamma \vdash M_1 : A \quad \Gamma/\Delta \vdash M_2 : B \quad N \leq \#_a(M_1)}{\Gamma/\Delta \vdash_a aM_1M_2/N}\ poly_{base}$$

$$\frac{\Gamma/\Delta \vdash a : A \to B \to Type \quad \Gamma \vdash M_1 : A \quad \Delta \vdash M_2 : B \quad \Gamma/\Delta \vdash_a D/N + \#_a(M_1)}{\Gamma/\Delta \vdash_a aM_1M_2 \to D/N}\ poly_{recur-fn}$$

$$\frac{\Gamma/\Delta \vdash b : C_1 \to C_2 \to Type \quad \Gamma/\Delta \vdash M_1 : C_1 \quad \Gamma/\Delta \vdash M_2 : C_2 \quad \vdash \Sigma_b\ \mathsf{poly} \quad \vdash \Sigma_b\ \mathsf{nsi} \quad \Gamma/\Delta \vdash_a D/N}{\Gamma/\Delta \vdash_a bM_1M_2 \to D/N}\ poly_{nsi-fn}$$

$$\frac{\Gamma/\Delta \vdash b : C_1 \to C_2 \to Type \quad \Gamma \vdash M_1 : C_1 \quad \Gamma \vdash M_2 : C_2 \quad \vdash \Sigma_b\ \mathsf{poly} \quad \Gamma/\Delta \vdash_a D/N}{\Gamma/\Delta \vdash_a bM_1M_2 \to D/N}\ poly_{normal-fn}$$

$$\frac{\Gamma, x : A/\Delta \vdash_a D/N}{\Gamma/\Delta \vdash_a \Pi x : A.D/N}\ poly_{var-normal} \qquad \frac{\Gamma/\Delta, x : A \vdash_a D/N}{\Gamma/\Delta \vdash_a \Pi x : A.D/N}\ poly_{var-recur}$$

Figure 1: Inference rules for identifying LF signatures representing polynomial time logic programs

$$\frac{}{\vdash \cdot \ \mathsf{nsi}}\ nsi_z \qquad \frac{\vdash \Sigma\ \mathsf{nsi} \quad \cdot/\cdot \vdash_{head(D)} D}{\vdash \Sigma, c : D\ \mathsf{nsi}}\ nsi_s$$

$$\frac{\Gamma \vdash_a M_1 : A \quad \#_a(M_2) \leq \#_a(M_1)}{\Gamma \vdash_a aM_1M_2\ \mathsf{nsi}}\ nsi_{base}$$

$$\frac{\Gamma \vdash_a M_1 : A \quad \begin{array}{c}\#_a(M_2) \leq \#_a(M_1)\\ \vdots \\ \Gamma \vdash_a D\ \mathsf{nsi}\end{array}}{\Gamma \vdash_a aM_1M_2 \to D\ \mathsf{nsi}}\ nsi_{recur}$$

$$\frac{\Gamma \vdash head(D) : A \to B \to Type \quad \vdash \Sigma_b\ \mathsf{nsi} \quad \begin{array}{c}\#_a(M_2) \leq \#_a(M_1)\\ \vdots \\ \Gamma \vdash_a D\ \mathsf{nsi}\end{array}}{\Gamma \vdash_a bM_1M_2 \to D\ \mathsf{nsi}}\ nsi_{fn}$$

$$\frac{\Gamma, x : A \vdash_a D\ \mathsf{nsi}}{\Gamma \vdash_a \Pi x : A.D\ \mathsf{nsi}}\ nsi_{var}$$

Figure 2: Inference rules for identifying LF signatures representing non-size increasing logic programs

Theorem 1.3 *[Ver97, p. 576-577] Given a recurrence* $T(x) = \Sigma_{i=1}^{k} a_i T(x/c_i) + f(x)$ *for all reals* $x > K$, $T(x) = b$ *for all reals* $1 \leq x \leq K$ *for some real constants* $a_i \geq 1$, $c_i > 1$ *for* $1 \leq i \leq k$ *and* $b > 0$ *and* f *is a function defined on reals. Also,* $K \geq max_i\{c_i\}$

1. *If* $f(x) \geq d$ *over* $[1, K]$ *for some* $d > 0$, *there exists* g *such that* f *is* g*-star-shaped, and* $\Sigma_{i=1}^{k} a_i g(1/c_i) < 1$ *then* $T(x) = \Theta(f(x))$.
2. *If* $f(x) = h(x)(\log x)^l + d(x \geq 1)$ *for some* $l \geq 0, d > 0$, *there is a* g *such that* h *is* g*-star-shaped, and* $\Sigma_{i=1}^{k} a_i g(1/c_i) = 1$, *then* $T(x) = O(f(x) \log x)$.
3. *If* $f(x) = h(x)(\log x)^l (x \geq 1, l \geq 0)$, *there exists* g *such that* h *is* g*-co-star-shaped, and* $\Sigma_{i=1}^{k} a_i g(1/c_i) \geq 1$, *then* $T(x) = \Omega(f(x) \log x)$.

Corollary 1.4 *Given a recurrence* $T(n) = \Sigma_{i=1}^{k} T(x_i) + f(n)$ *for all reals* $n > 0$, $T(1) = b$ *for some real constants* $x_i > 0$ *for* $1 \leq i \leq k$ *and function* f *is a polynomial such that* $f(1) > 0$.

1. *If* $n > \Sigma_{i=1}^{k} x_i$, *then* $T(n) = \Theta(f(n))$.
2. *If* $n = \Sigma_{i=1}^{k} x_i$, *then* $T(n) = O(f(n) \log n) = O(n f(n))$.

Proof: Choose $g(x) = x$ and $c_i = n/x_i$ for $1 \leq i \leq k$ in Theorem 1.3. □

Theorem 1.5 *Given a LF type signature* Σ *such that* $\mathcal{D} :: \vdash \Sigma\ \mathsf{poly}$. *Then for all* D *such that* $c : D \in \Sigma$ *with* $head(D) = f : A \to B \to Type$ *and mode* $m_f = \langle +, - \rangle$ *there exists a polynomial* $p(n)$ *such that for all LF objects* $M_1 : A$, *if there exists* $M_2 : B$ *and* $N : f\ M_1\ M_2$ *then* $\#_f(N) = p(\#_f(M_1))$.

Proof: Induction on size of Σ and then for $c : D \in \Sigma$ on size of M_1. □

Theorem 1.6 *Given a LF type signature* Σ *such that* $\mathcal{D} :: \vdash \Sigma\ \mathsf{nsi}$. *Then for all* D *such that* $c : D \in \Sigma$ *with* $head(D) = f : A \to B \to Type$ *and mode* $m_f = \langle +, - \rangle$, *if for all LF objects* $M_1 : A$, *there exists* $M_2 : B$ *and* $N : f\ M_1\ M_2$ *then* $\#_f(M_2) \leq \#_f(M_1)$.

Proof: Induction on size of Σ and then for $c : D \in \Sigma$ on size of M_1. □

Acknowledgements

Dana Angluin, Harald Gangziger, Chetana Shah, Bridgette Pientka, Helmut Schwichtenberg

References

[Ver97] Rakesh M. Verma. General techniques for analyzing recursive algorithms with applications. *SIAM Journal on Computing*, 26(2):568–581, 1997.