Languages details - draft

June 29, 2012

1 Labeled language

1.1 Statics

1.1.1 Language

This is a description of actual language, as presented in [1].

$$\mathrm{hyp_L}\frac{w \in \Omega}{\Omega; (x : A@w) \cup \Gamma \vdash \mathrm{hyp_L} x : A@w}$$

$$\mathrm{lam_L}\frac{\Omega; (x : A@w) \cup \Gamma \vdash M : A'@w}{\Omega; \Gamma \vdash \lambda_L x, M : A \to A'@w}$$

$$\mathrm{appl_L}\frac{\Omega; \Gamma \vdash M : A \to A'@w \quad \Omega; \Gamma \vdash N : A@w}{\Omega; \Gamma \vdash \mathrm{appl_L} M N : A'@w}$$

$$\mathrm{box_L}\frac{w \in \Omega \quad w' \mathrm{fresh} \quad w' \cup \Omega; \Gamma \vdash M : A@w'}{\Omega; \Gamma \vdash \mathrm{box_L} w'.M : \Box A@w}$$

$$\mathrm{unbox_L}\frac{\Omega; \Gamma \vdash M : \Box A@w}{\Omega; \Gamma \vdash \mathrm{unbox_L} M : A@w}$$

$$\mathrm{here_L}\frac{\Omega; \Gamma \vdash M : A@w}{\Omega; \Gamma \vdash \mathrm{here_L} M : \Diamond A@w}$$

$$\mathrm{letd_L}\frac{\Omega; \Gamma \vdash M : \Diamond A@w \quad w' \mathrm{fresh} \quad \cup \Omega; (A, w') \cup \Gamma \vdash N : B@w}{\Omega; \Gamma \vdash \mathrm{letdia_L} x := M \mathrm{in} N : B@w}$$

$$\mathrm{fetch_L}\frac{w \in \Omega \quad \Omega; \Gamma \vdash M : \Box A@w'}{\Omega; \Gamma \vdash \mathrm{fetch_L} w' M : \Box A@w'}$$

$$\mathrm{get_L}\frac{w \in \Omega \quad \Omega; \Gamma \vdash M : \Diamond A@w'}{\Omega; \Gamma \vdash \mathrm{get_L} w'.M : \Diamond A@w'}$$

1.1.2 Implementation

Note 1 *This is just current, not final implementation of labeled language. Planned changes include using locally nameless instead of de Bruijn indices for variables and replacing current implementation of Ω (using fsets from Metatheory library) with one using lists and their permutations.*
Both changes are due to compatibility in translation with label-free language.

$$\mathrm{hyp_L}\frac{w \in \Omega \quad \Gamma[n] = (A, w)}{\Omega; \Gamma \vdash \mathrm{hyp_L} n : A@w}$$

$$\mathrm{lam_L}\frac{\Omega; (A, w) :: \Gamma \vdash M : A'@w}{\Omega; \Gamma \vdash \lambda_L A, M : A \to A'@w}$$

$$\mathrm{appl_L}\frac{\Omega; \Gamma \vdash M : A \to A'@w \quad \Omega; \Gamma \vdash N : A@w}{\Omega; \Gamma \vdash \mathrm{appl_L} M N : A'@w}$$

$$\mathrm{box_L}\frac{w \in \Omega \quad \forall_w w \notin \mathrm{L} \to w' \cup \Omega; \Gamma \vdash M^{w'} : A@w'}{\Omega; \Gamma \vdash \mathrm{box_L} M : \Box A@w}$$

$$\mathrm{unbox_L}\frac{\Omega; \Gamma \vdash M : \Box A@w}{\Omega; \Gamma \vdash \mathrm{unbox_L} M : A@w}$$

$$\mathrm{here_L}\frac{\Omega; \Gamma \vdash M : A@w}{\Omega; \Gamma \vdash \mathrm{here_L} M : \Diamond A@w}$$

$$\mathrm{letd_L}\frac{w \in \Omega \quad \Omega; \Gamma \vdash M : \Diamond A@w \quad \forall_{w'} w' \notin \mathrm{L} \to w' \cup \Omega; (A, w') :: \Gamma \vdash N^{w'} : B@w}{\Omega; \Gamma \vdash \mathrm{letdia_L} M \mathrm{in} N : B@w}$$

$$\mathrm{fetch_L}\frac{w \in \Omega \quad \Omega; \Gamma \vdash M : \Box A@w'}{\Omega; \Gamma \vdash \mathrm{fetch_L} w' M : \Box A@w'}$$

$$\mathrm{get_L}\frac{w \in \Omega \quad \Omega; \Gamma \vdash M : \Diamond A@w'}{\Omega; \Gamma \vdash \mathrm{get_L} w' M : \Diamond A@w'}$$

1.2 Dynamics

1.2.1 Values

$$\mathrm{value}_{\mathrm{L}}(\lambda_L A, M)$$

$$\mathrm{value}_{\mathrm{L}}(\mathrm{box}_{\mathrm{L}} M)$$

$$\mathrm{value}_{\mathrm{L}}(M) \to \mathrm{value}_{\mathrm{L}}(\mathrm{here}_{\mathrm{L}} M)$$

1.2.2 Reductions

For reduction steps, we silently use assumption that all terms are locally closed with respect to worlds. This requirement may later be moved to static semantics, as is is widely used both in definitions and lemmas.

We do not list trivial inductive reduction steps here.

$$\mathrm{appl}_{\mathrm{L}}(\lambda_L A, M)N, w \mapsto [N/0]M, w$$

$$\mathrm{unbox}_{\mathrm{L}}(\mathrm{box}_{\mathrm{L}} M), w \mapsto M^w, w$$

$$\mathrm{letd}_{\mathrm{L}}(\mathrm{here}_{\mathrm{L}} M)\mathrm{in}N, w \mapsto [M/0]N^w, w$$

$$\mathrm{value}_{\mathrm{L}} M \to \mathrm{fetch}_{\mathrm{L}} w M, w' \mapsto \{w'/w\}M, w'$$

$$\mathrm{value}_{\mathrm{L}} M \to \mathrm{get}_{\mathrm{L}} w(\mathrm{here}_{\mathrm{L}} M), w' \mapsto \mathrm{here}_{\mathrm{L}}\{w'/w\}M, w'$$

1.3 Substitution

Since the labeled language has only one context stack, both definition and actual implementation of substitution functions is streight forward. For the sake of completeness, we will provide the definitions.

1.3.1 "world" substitution

We use helper function shift, which does the following:

- On free world, it is an identity function
- On bound world, it increases the binding number by 1

$$\{w_1/w_2\}(\mathrm{hyp_L}n) \Rightarrow \mathrm{hyp_L}n$$
$$\{w_1/w_2\}(\lambda_L A, M) \Rightarrow \lambda_L A, \{w_1/w_2\}M$$
$$\{w_1/w_2\}(\mathrm{appl_L}MN) \Rightarrow \mathrm{appl_L}\{w_1/w_2\}M\{w_1/w_2\}N$$
$$\{w_1/w_2\}(\mathrm{box_L}M) \Rightarrow \mathrm{box_L}\{\mathrm{shift}w_1/\mathrm{shift}w_2'\}M$$
$$\{w_1/w_2\}(\mathrm{unbox_L}M) \Rightarrow \mathrm{unbox_L}\{w_1/w_2\}M$$
$$\{w_1/w_2\}(\mathrm{here_L}n) \Rightarrow \mathrm{here_L}\{w_1/w_2\}M$$
$$\{w_1/w_2\}(\mathrm{letd_L}MN) \Rightarrow \mathrm{letd_L}\{w_1/w_2\}M\{\mathrm{shift}w_1/\mathrm{shift}w_2\}M$$
$$\{w_1/w_2\}(\mathrm{fetch_L}w_2M) \Rightarrow \mathrm{fetch_L}w_1\{w_1/w_2\}M$$
$$\{w_1/w_2\}(\mathrm{fetch_L}wM) \Rightarrow \mathrm{fetch_L}w\{w_1/w_2\}M$$
$$\{w_1/w_2\}(\mathrm{get_L}w_2M) \Rightarrow \mathrm{get_L}w_1\{w_1/w_2\}M$$
$$\{w_1/w_2\}(\mathrm{get_L}wM) \Rightarrow \mathrm{get_L}w\{w_1/w_2\}M$$

1.3.2 "term" substitution

$$[M/n]\mathrm{hyp_L}n \Rightarrow M$$
$$[M/n]\mathrm{hyp_L}m \Rightarrow \mathrm{hyp_L}m$$
$$[M/n]\lambda_L A, N \Rightarrow \lambda_L A, [M/\mathrm{S}n]N$$
$$[M/n]\mathrm{appl_L}N_1N_2 \Rightarrow \mathrm{appl_L}[M/n]N_1[M/n]N_2$$
$$[M/n]\mathrm{box_L}N \Rightarrow \mathrm{box_L}[M/n]N$$
$$[M/n]\mathrm{unbox_L}N \Rightarrow \mathrm{unbox_L}[M/n]N$$
$$[M/n]\mathrm{here_L}N \Rightarrow \mathrm{here_L}[M/n]N$$
$$[M/n]\mathrm{letd_L}N_1\mathrm{in}N_2 \Rightarrow \mathrm{letd_L}[M/n]N_1\mathrm{in}[M/\mathrm{S}n]N_2$$
$$[M/n]\mathrm{fetch_L}N \Rightarrow \mathrm{fetch_L}[M/n]N$$
$$[M/n]\mathrm{get_L}N \Rightarrow \mathrm{get_L}[M/n]N$$

2 Label-free language

2.1 Statics

2.2 Dynamics

3 Translation

References

[1] Tom Murphy VII, Karl Crary, Robert Harper, Frank Pfenning A Symetric Modal Lambda calculus for Distributed Computing