LinML

Ada Vienot

1 Typing

The types in LinML have the following syntax :

$$
\begin{array}{rcll}
\tau, \sigma & ::= & 0 \mid 1 \mid \top & \text{(type constants)} \\
& \mid & a & \text{(type variables)} \\
& \mid & \tau \multimap \sigma & \text{(linear abstraction)} \\
& \mid & \tau \to \sigma & \text{(unchecked abstraction)} \\
& \mid & \tau + \sigma & \text{(additive disjunction)} \\
& \mid & \tau * \sigma & \text{(multiplicative conjunction)} \\
& \mid & \tau \,\&\, \sigma & \text{(multiplicative disjunction)} \\
& \mid & !\tau & \text{(of course)}
\end{array}
$$

Figure 1: LinML types

1.1 Typing rules

We define the inductive predicate exp over types as follows :

$$
\exp(\tau) = \begin{cases} \text{true} & \text{if } \tau = * \text{ or } \tau = !\sigma \\ \text{false} & \text{elsewise} \end{cases}
$$

In all the following, typing contexts $\Gamma \| \Delta$ are to be seen as a pair with :

- Γ, called a *linear context*, is a multiset of pairs $(x : \tau)$, with x a variable and τ a type such that $\exp(\tau) = \text{false}$
- Δ, called an *exponential context*, is a set of pairs $(x : \tau)$, with x a variable and τ a type such that $\exp(\tau) = \text{true}$

Since $\exp(\tau)$ implies the existence of context weakening and contraction rules for τ, we treat the context Δ as the kind of contexts we manipulate in intuitionistic sequent calculus, that is : they can be duplicated, erased, and one binding is allowed to erase another. This is not the case of the context Γ containing linear bindings.

1.1.1 Terms

Let t, u be terms, β, β' booleans, τ, σ types, and $(\Gamma \parallel \Delta), (\Gamma' \parallel \Delta')$ typing contexts. The typing judgements for terms have the following shape :

$$\Gamma \parallel \Delta \vdash t : \tau \Rightarrow \Gamma' \rtimes \beta$$

Γ' is the linear context where all the necessary bindings to type $t : \tau$ have been consumed.
$\beta \in \{\top, \bot\}$ is the called the "modality of the judgement".

A program p is well typed of type τ iff :

$$\varnothing \parallel \varnothing \vdash p : \tau \Rightarrow \varnothing \rtimes \beta$$

The typing rules are the following:

$$\frac{}{\Gamma \parallel \Delta \vdash * : 1 \Rightarrow \Gamma \rtimes \bot}\ (1) \qquad \frac{}{\Gamma \parallel \Delta \vdash \langle\rangle : \top \Rightarrow \Gamma \rtimes \top}\ (\top)$$

$$\frac{\Gamma \parallel \Delta \vdash t : 0 \Rightarrow \Gamma' \rtimes \beta'}{\Gamma \parallel \Delta \vdash \textbf{refute}\ \tau\ \textbf{with}\ t : \tau \Rightarrow \Gamma' \rtimes \top}\ (0)$$

$$\frac{(x : \tau) \in \Gamma \qquad \neg \exp(\tau)}{\Gamma \parallel \Delta \vdash x : \tau \Rightarrow \Gamma \setminus (x : \tau) \rtimes \bot}\ (\text{vlin}) \qquad \frac{(x : \tau) \in \Delta \qquad \exp(\tau)}{\Gamma \parallel \Delta \vdash x : \tau \Rightarrow \Gamma \rtimes \bot}\ (\text{vexp})$$

$$\frac{\Gamma \parallel \Delta \vdash t : \tau \Rightarrow \Gamma' \rtimes \beta' \qquad \Gamma' \parallel \Delta \vdash u : \sigma \Rightarrow \Gamma'' \rtimes \beta''}{\Gamma \parallel \Delta \vdash (t, u) : \tau * \sigma \Rightarrow \Gamma'' \rtimes \beta' \vee \beta''}\ (\otimes)$$

$$\frac{\Gamma \parallel \Delta \vdash t : \tau \Rightarrow \Gamma' \rtimes \beta' \qquad \Gamma \parallel \Delta \vdash u : \sigma \Rightarrow \Gamma'' \rtimes \beta'' \qquad \beta' \implies \Gamma' \subseteq \Gamma'' \qquad \beta'' \implies \Gamma'' \subseteq \Gamma'}{\Gamma \parallel \Delta \vdash \langle t, u\rangle : \tau \mathbin{\&} \sigma \Rightarrow \Gamma' \cap \Gamma'' \rtimes \beta' \wedge \beta''}\ (\&)$$

$$\frac{\Gamma \parallel \Delta \vdash t : \sigma \Rightarrow \Gamma' \rtimes \beta'}{\Gamma \parallel \Delta \vdash (t :> \_ + \tau) : \sigma + \tau \Rightarrow \Gamma' \rtimes \beta'}\ (\oplus\text{-l}) \qquad \frac{\Gamma \parallel \Delta \vdash u : \tau \Rightarrow \Gamma' \rtimes \beta'}{\Gamma \parallel \Delta \vdash (u :> \sigma + \_) : \sigma + \tau \Rightarrow \Gamma' \rtimes \beta'}\ (\oplus\text{-r})$$

$$\frac{\varnothing \parallel \Delta \vdash t : \tau \Rightarrow \varnothing \rtimes \beta \qquad \Gamma = \Gamma'}{\Gamma \parallel \Delta \vdash !t : !\tau \Rightarrow \Gamma \rtimes \bot}\ (!) \qquad \frac{\Gamma \parallel \Delta \vdash !t : !\tau \Rightarrow \Gamma' \rtimes \beta'}{\Gamma \parallel \Delta \vdash !!t : !\tau \Rightarrow \Gamma' \rtimes \beta'}\ (\text{!-dig})$$

$$\frac{(\Gamma \setminus x) \cdot (x : \tau) \parallel \Delta \setminus x \vdash t : \sigma \Rightarrow \Gamma' \rtimes \beta' \qquad x \in \Gamma' \implies \beta' \qquad \neg \exp(\tau) \qquad \tau \neq \top}{\Gamma \parallel \Delta \vdash \textbf{fun}\ (x : \tau) \multimap t : \tau \multimap \sigma \Rightarrow \Gamma' \setminus (x : \tau) \rtimes \beta'}\ (\multimap)$$

$$\frac{\Gamma \setminus x \parallel (\Delta \setminus x) \cup (x : \tau) \vdash t : \sigma \Rightarrow \Gamma' \rtimes \beta' \qquad \exp(\tau)}{\Gamma \parallel \Delta \vdash \textbf{fun}\ (x : \tau) \multimap t : \tau \multimap \sigma \Rightarrow \Gamma' \rtimes \beta'}\ (\multimap\text{-!})$$

$$\frac{\Gamma \parallel \Delta \vdash t : \sigma \multimap \tau \Rightarrow \Gamma' \rtimes \beta' \qquad \Gamma' \parallel \Delta \vdash t' : \sigma \Rightarrow \Gamma'' \rtimes \beta''}{\Gamma \parallel \Delta \vdash tt' : \tau \Rightarrow \Gamma'' \rtimes \beta' \vee \beta'}\ (\multimap\text{-app})$$

$$\frac{\Gamma \parallel \Delta \vdash t : !(\sigma \multimap \tau) \Rightarrow \Gamma \rtimes \bot \qquad \Gamma \parallel \Delta \vdash t' : \sigma \Rightarrow \Gamma' \rtimes \beta'}{\Gamma \parallel \Delta \vdash tt' : \tau \Rightarrow \Gamma'' \rtimes \beta'}\ (\multimap\text{-!app})$$

$$\frac{\Gamma \parallel \Delta \vdash t : \sigma \Rightarrow \Gamma' \rtimes \beta' \quad \Gamma' \cdot (x : \sigma) \parallel \Delta \setminus x \vdash t' : \tau \Rightarrow \Gamma'' \rtimes \beta'' \quad x \in \Gamma'' \implies \beta'' \quad x \notin \Gamma' \quad \neg \exp(\sigma) \quad \sigma \neq \top}{\Gamma \parallel \Delta \vdash \textbf{give}\ x = t\ \textbf{in}\ t' : \tau \Rightarrow (\Gamma'' \setminus x) \rtimes \beta' \vee \beta''}\ (\text{give})$$

$$\frac{\Gamma \parallel \Delta \vdash t : \sigma \Rightarrow \Gamma' \rtimes \beta' \qquad \Gamma' \parallel \Delta \cdot (x : \sigma) \vdash t' : \tau \Rightarrow \Gamma'' \rtimes \beta'' \qquad \exp(\sigma)}{\Gamma \parallel \Delta \vdash \textbf{give}\ x = t\ \textbf{in}\ t' : \tau \Rightarrow \Gamma'' \rtimes \beta' \vee \beta''}\ (\text{!-give})$$

$$\frac{\Gamma \parallel \Delta \vdash t : \sigma \Rightarrow \Gamma' \rtimes \beta' \qquad \overrightarrow{\Gamma' \parallel \Delta \vdash p_i : \sigma \Uparrow \gamma_i} \qquad \overrightarrow{(\Gamma' \parallel \Delta) \cup \gamma_i \vdash m_i : \tau \Rightarrow \Gamma_i \rtimes \beta_i} \qquad \overrightarrow{\neg \beta_i \implies \gamma_i \cap \Gamma_i = \varnothing}}{\Gamma \parallel \Delta \vdash \textbf{match}\ t\ \textbf{with}\ \overrightarrow{p_i \to m_i} : \tau \Rightarrow \left(\bigcap (\Gamma_i \setminus \gamma_i)\right) \rtimes \left(\beta' \vee \left(\bigvee \beta_i\right)\right)}\ (\text{match-r})$$

Figure 2: LinML terms typing rules

1.1.2 Patterns

Let p, p' be patterns, τ, σ types, Γ, Γ' linear typing contexts, and γ, γ' a multiset of bindings. The typing judgements for patterns have the following shape :

$$\Gamma \vdash p : \tau \Uparrow \gamma$$

γ contains the bindings that evaluating p added to the environment.

The typing rules are the following:

$$\frac{}{\Gamma \vdash * : 1 \Uparrow \varnothing}\ (\mathscr{P}\text{-1}) \qquad \frac{\neg\exp(\tau) \implies x \notin \Gamma \qquad \tau \neq \top}{\Gamma \vdash x : \tau \Uparrow \{x : \tau\}}\ (\mathscr{P}\text{-var}) \qquad \frac{\Gamma \vdash p : \tau \Uparrow \gamma}{\Gamma \vdash (p : \tau) : \tau \Uparrow \gamma}\ (\mathscr{P}\text{-ty})$$

$$\frac{\Gamma \vdash p : \sigma \Uparrow \gamma}{\Gamma \vdash \langle p, - \rangle : \sigma \,\&\, \tau \Uparrow \gamma}\ (\mathscr{P}\text{-\&left}) \qquad \frac{\Gamma \vdash p : \tau \Uparrow \gamma}{\Gamma \vdash \langle -, p \rangle : \sigma \,\&\, \tau \Uparrow \gamma}\ (\mathscr{P}\text{-\&right})$$

$$\frac{\Gamma \vdash p : \sigma \Uparrow \gamma}{\Gamma \vdash (p <: \_ + \tau) : \sigma \oplus \tau \Uparrow \gamma}\ (\mathscr{P}\text{-}\oplus\text{left}) \qquad \frac{\Gamma \vdash p : \tau \Uparrow \gamma}{\Gamma \vdash (p <: \sigma + \_) : \sigma \oplus \tau \Uparrow \gamma}\ (\mathscr{P}\text{-}\oplus\text{right})$$

$$\frac{\Gamma \vdash p : \sigma \Uparrow \gamma \qquad \Gamma \vdash p' : \tau \Uparrow \gamma' \qquad \gamma \cap \gamma' = \varnothing}{\Gamma \vdash (p, p') : \sigma * \tau \Uparrow \gamma \cup \gamma'}\ (\mathscr{P}\text{-}\otimes)$$

$$\frac{\Gamma \vdash p : \tau \Uparrow \gamma}{\Gamma \vdash !p : !\tau \Uparrow \gamma}\ (\mathscr{P}\text{-!}) \qquad \frac{}{\Gamma \vdash \_ : !\tau \Uparrow \varnothing}\ (\mathscr{P}\text{-!weaken})$$

$$\frac{\Gamma \vdash p : \tau \Uparrow \gamma \qquad \Gamma \vdash p' : \tau \Uparrow \gamma' \qquad \gamma = \gamma'}{\Gamma \vdash p \mid p' : \tau \Uparrow \gamma}\ (\mathscr{P}\text{-disj})$$

Figure 3: LinML pattern typing rules

2 LL translation

2.1 Prerequisites

2.2 Defining the translation

In all the following, we will only consider the conservative fragment of LinML.
The typing jugement $\Gamma; \Delta \vdash t : \tau \Rightarrow \Gamma'$ has a direct translation in linear logic. The following theorem holds :

$$\forall \text{ typing jugement } \Gamma; \Delta \vdash t : \tau \Rightarrow \Gamma' \quad \exists \text{ a linear sequent of interface } \Gamma \vdash \otimes(\Gamma' \cup \tau)$$