A Linear Dependent Type Theory*

Zhaohui Luo[1] and Yu Zhang[2]

[1] Royal Holloway, Univ of London
zhaohui.luo@hotmail.co.uk
[2] Institute of Software, Chinese Academy of Sciences
yzhang@ios.ac.cn

Introduction. Introducing linear types [2] (and, in general, resource sensitive types [6]) into a type theory with dependent types has been an interesting but difficult topic. One of the most difficult issues is whether to allow types to depend on linear variables. For instance, for $f : A \multimap A$, the equality type $Eq_A(f\ x,\ x)$ depends on the linear variable x of type A. In all of the existing research so far (see, for example, [1, 3, 5]), types are only allowed to depend on intuitionistic variables, but not on linear variables. In this paper, we present LDTT, a dependent type theory where types may depend on linear variables.

LDTT: a Linear Dependent Type Theory. A context in LDTT may contain two forms of entries: the usual (intuitionistic) entries $x{:}A$ and the linear ones $y{::}A$, where y is called a linear variable. For any term t, $FV(t)$ is the set of free variables occurring in t and, for any context Γ, if $FV(t) \subseteq FV(\Gamma)$, then $D_\Gamma(t)$ is the set of dependent variables of t w.r.t. Γ, defined as follows: (1) $FV(t) \subseteq D_\Gamma(t)$, and (2) for any $x \in D_\Gamma(t)$, $FV(\Gamma_x) \subseteq D_\Gamma(t)$. In LDTT, we have the following variable typing rule:

$$(V) \qquad \frac{\Gamma,\ x{:}\bar{:}A,\ \Gamma'\ \mathit{valid} \quad (\text{for all } y{::}\Gamma_y \in \Gamma,\ y \in D_\Gamma(x)) \quad \Gamma' \text{ intuitionistic}}{\Gamma,\ x{:}\bar{:}A,\ \Gamma' \vdash x : A} \quad (\bar{:} \in \{:, ::\})$$

where 'Γ' intuitionistic' means that Γ' does not contain any linear entries.

We have two forms of Π-types: the intuitionistic $\Pi x{:}A.B$ and linear $\Pi x{::}A.B$, whose rules are given in Figure 1. For both, the formation and introduction rules are not unusual, although their elimination rules need some explanations. For intuitionistic Π-types, in order to type $f(a)$ under context Γ, a is required to be typable in $\overline{\Gamma}$, the intuitionistic part of Γ, obtained from Γ by removing all the entries whose variables are in $FV_{LD}(\Gamma)$, the set of linear dependent variables in Γ.[1] The elimination rule for linear Π-types involves the operation $Merge(\Gamma;\Delta)$, which is only defined, notation $Merge(\Gamma;\Delta)\downarrow$, if $\overline{\Gamma} \equiv \overline{\Delta}$ and $FV_{LD}(\Gamma) \cap FV_{LD}(\Delta) = \emptyset$: (1) $Merge(\langle\rangle;\langle\rangle) = \langle\rangle$; (2) If $x \in FV_{LD}(\Gamma,\Delta)$, $Merge(\Gamma, x{:}\bar{:}A;\ \Delta) = Merge(\Gamma;\ \Delta, x{:}\bar{:}A) = Merge(\Gamma;\Delta), x{:}\bar{:}A$, where $\bar{:}$ is either : or ::; and (3) $Merge(\Gamma, x{:}A;\ \Delta, x{:}A) = Merge(\Gamma;\Delta), x{:}A$.

LDTT also contains equality types $Eq_A(a,b)$, whose rules are given in Figure 2. The Eq-formation rule involves another context merge operation $merge(\Gamma;\Delta)$, which is only defined, notation $merge(\Gamma;\Delta)\downarrow$, under the condition that, if $x{:}\bar{:}A \in \Gamma$ and $x{:}\bar{:}B \in \Delta$, then (1) $\bar{:}$ is either both : or both :: and (2) $A \equiv B$: (1) $merge(\Gamma;\langle\rangle) = \Gamma$, and (2) for $\bar{:}$ being either : or ::, $merge(\Gamma;\ x{:}\bar{:}A, \Delta)$ is equal to (i) $merge(\Gamma;\Delta)$, if $x \in FV(\Gamma)$, and (ii) $merge(\Gamma, x{:}\bar{:}A;\ \Delta)$, otherwise. Its elimination rule involves the $Merge$-operation defined earlier.

In linear logic, every linear variable occurs free for exactly once in a typed term. In LDTT, every linear variable *occurs essentially* for exactly once in a typed term – a property we call

*Partially supported by EU COST Action CA15123 and CAS/SAFEA International Partnership Program.

[1] Formally, $FV_{LD}(\Gamma)$ is defined as follows: (1) $FV_{LD}(\langle\rangle) = \emptyset$; (2) $FV_{LD}(\Gamma, x{::}A) = FV_{LD}(\Gamma) \cup \{x\}$; (3) if $FV(A) \cap FV_{LD}(\Gamma) = \emptyset$, then $FV_{LD}(\Gamma, x{:}A) = FV_{LD}(\Gamma)$; otherwise, $FV_{LD}(\Gamma, x{:}A) = FV_{LD}(\Gamma) \cup \{x\}$.

Intuitionistic Π-*types* (Conversion: $(\lambda x{:}A.b)(a) \simeq [a/x]b$)

$$\frac{\Gamma,\ x:A \vdash B\ type}{\Gamma \vdash \Pi x{:}A.B\ type} \quad \frac{\Gamma,\ x:A \vdash b:B}{\Gamma \vdash \lambda x{:}A.b : \Pi x{:}A.B} \quad \frac{\Gamma \vdash f : \Pi x{:}A.B \quad \overline{\Gamma} \vdash a : A}{\Gamma \vdash f(a) : [a/x]B}$$

Linear Π-*types* (Conversion: $(\lambda x{::}A.b)\ a \simeq [a/x]b$)

$$\frac{\Gamma,\ x{::}A \vdash B\ type}{\Gamma \vdash \Pi x{::}A.B\ type} \quad \frac{\Gamma,\ x{::}A \vdash b:B}{\Gamma \vdash \lambda x{::}A.b : \Pi x{::}A.B} \quad \frac{\Gamma \vdash f : \Pi x{::}A.B \quad \Delta \vdash a : A \quad Merge(\Gamma;\Delta)\downarrow}{Merge(\Gamma;\Delta) \vdash f\ a : [a/x]B}$$

Figure 1: Intuitionistic and linear Π-types

Equality types (Conversion: $subst(refl(a), q) \simeq q$)

$$\frac{\Gamma \vdash a : A \quad \Delta \vdash b : A \quad merge(\Gamma;\Delta)\downarrow}{merge(\Gamma;\Delta) \vdash Eq_A(a,b)\ type} \quad \frac{\Gamma \vdash a : A}{\Gamma \vdash refl(a) : Eq_A(a,a)}$$

$$\frac{\Gamma \vdash p : Eq_A(a,b) \quad \Delta \vdash q : B[a] \quad Merge(\Gamma;\Delta), x{:}^{\bar{:}}A \vdash B[x]\ type\ (\bar{:} \in \{:,::\}) \quad Merge(\Gamma;\Delta)\downarrow}{Merge(\Gamma;\Delta) \vdash subst(x.B,p,q) : B[b]}$$

Figure 2: Equality types

weak linearity. More precisely, for $\Gamma \vdash a : A$, the multiset of variables essentially occurring in a under Γ, $E_\Gamma(a)$, is defined by induction on derivations (we omit the part of the definition for Eq-types): (1) for (V) above, $E_{\Gamma,\ x{:}^{\bar{:}}A,\ \Gamma'}(x) = D_{\Gamma,\ x{:}^{\bar{:}}A,\ \Gamma'}(x)$; (2) for the λ-typing rules, $E_\Gamma(\lambda x{:}^{\bar{:}}A.b) = E_{\Gamma,x{:}^{\bar{:}}A}(b)\backslash\{x\}$, where $\bar{:} \in \{:,::\}$; (3) for intuitionistic applications, $E_\Gamma(f(a)) = E_\Gamma(f) \cup E_{\overline{\Gamma}}(a)$; (4) for linear applications, $E_{Merge(\Gamma;\Delta)}(f\ a) = E_\Gamma(f) \cup E_\Delta(a)$.

Theorem (weak linearity) If $\Gamma \vdash a : A$ and $x{::}\Gamma_x \in \Gamma$, then $x \in E_\Gamma(a)$ only once. □

Implementation and future work. We have implemented a prototype of the type checking algorithm for LDTT, which includes the rules in Figures 1 and 2. The code can be found at `https://github.com/yveszhang/ldtyping`.

LDTT as presented shows a way to introduce types that may depend on linear variables. Future work includes the study of extensions to other type constructors. Dependent types were introduced into the Lambek calculus in [4] and it would be interesting to see how our work above can be incorporated to allow type dependency on Lambek variables.

References

[1] I. Cervesato and F. Pfenning. A linear logical framework. *Information and Computation*, 179, 2002.

[2] J.-Y. Girard. Linear logic. *Theoret. Comput. Sci.*, 50, 1987.

[3] N. Krishnaswami, P. Pradic, and N. Benton. Integrating dependent and linear types. *POPL 2015*.

[4] Z. Luo. A Lambek Calculus with Dependent Types. *TYPES 2015*.

[5] M. Vákár. A categorical semantics for linear logical frameworks. *FoSSaCS 2015*.

[6] D. Walker. Substructural type systems. In B. Pierce, editor, *Advanced Topics in Types and Programming Languages*, pages 3–43. MIT, 2005.