# A Linear Dependent Type Theory

| Zhaohui Luo          | Yu Zhang                    |
|----------------------|-----------------------------|
| Royal Holloway       | Institute of Software       |
| University of London | Chinese Academy of Sciences |

#### Linear types and dependent types

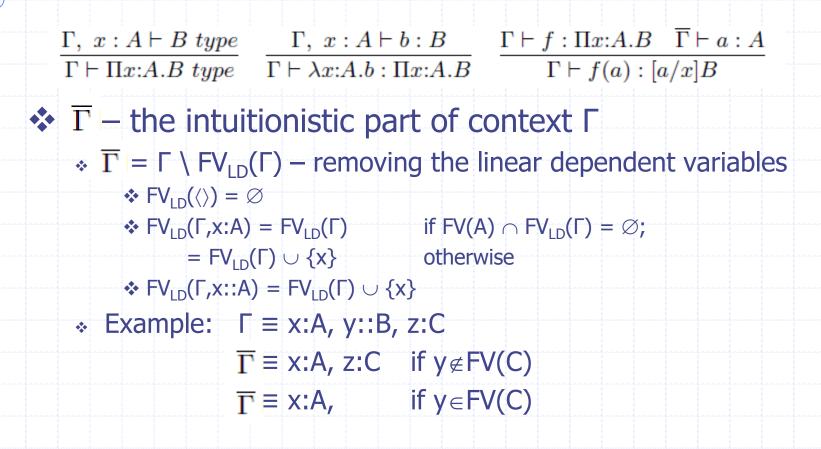
- ✤ Linear types (Girard 1987): A–•B
- ✤ Dependent types (Martin-Löf 1970s): Πx:A.B[x]
- How to combine them?
  - In most of existing work (Pfenning et al 2002, Krishnaswami et al 2015, Vákár 2015)
    - ✤ B[x] only when x is intuitionistic.
    - ↔ Hence it is possible to separate intuitionistic  $\Gamma$  and linear  $\Delta$ :  $\Gamma$ ;  $\Delta$  |- a : A
    - \*  $\Delta$  depends on  $\Gamma$ , but not the other way around.
  - \* McBride (2016)
    - "Prices" in contextual entries and typing and allow type dependency on 0-priiced variables – discussion later.
    - Independent with this work (we became aware of Conor's work only two weeks ago – detailed comparison due.)

This paper: LDTT, where types can depend on linear variables.

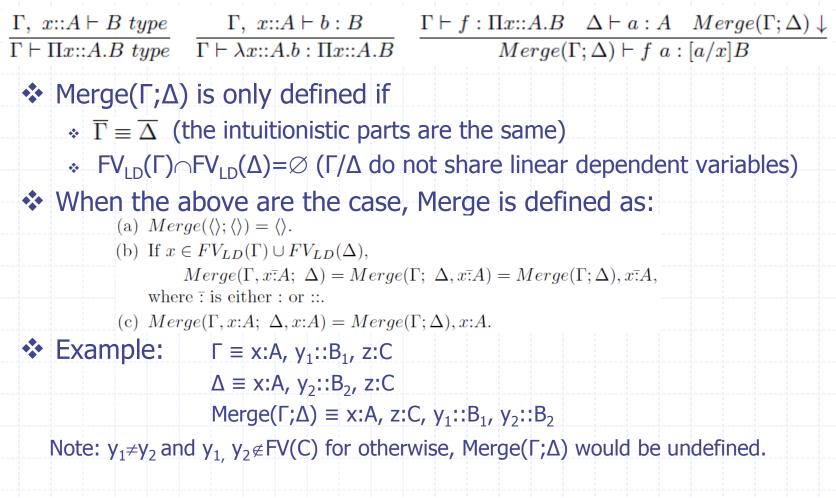
### LDTT: Linear and Intuitionistic Variables

- Contexts are sequences of two forms of entries: x:A, y::B[x], z:C[x,y], ...
  - \* Intuitionistic variables x : A
  - ✤ Linear variables
    y :: B
- Types dependent on linear variables
  - \* Example: x::A, f :  $A \rightarrow A \vdash Eq_A(f x, x)$  type

#### Intuitionistic $\Pi$ -types



#### Linear *Π*-types



May 2016

**TYPES 2016** 

### **Equality Types**

Formation rule  $\Gamma \vdash a: A \quad \Delta \vdash b: A \quad merge(\Gamma; \Delta) \downarrow$  $merge(\Gamma; \Delta) \vdash E\overline{q_A(a, b) \ type}$ \* merge( $\Gamma$ ;Δ) is defined only when var-sharing is OK:  $x \ge A \in \Gamma$ ,  $x \ge B \in \Delta \Rightarrow A \equiv B$  and 2 is both : or both :: \* merge( $\Gamma$ ;Δ) is defined as (a)  $merge(\Gamma; \langle \rangle) = \Gamma.$ (b)  $merge(\Gamma; x; A, \Delta) = \begin{cases} merge(\Gamma; \Delta) & \text{if } x \in FV(\Gamma) \\ merge(\Gamma, x; A; \Delta) & \text{otherwise} \end{cases}$ Examples: \* x::A, f : A- $A \vdash f x$  : A and x::A  $\vdash x$  : A  $\rightarrow$  x::A, f : A- $A \vdash Eq_A(f x, x)$  type \* x::A  $\mid$  x : A and y::A  $\mid$  y : A  $\rightarrow$  x::A, y::A  $\mid$  Eq(x,y) type

#### Introduction and elimination rules

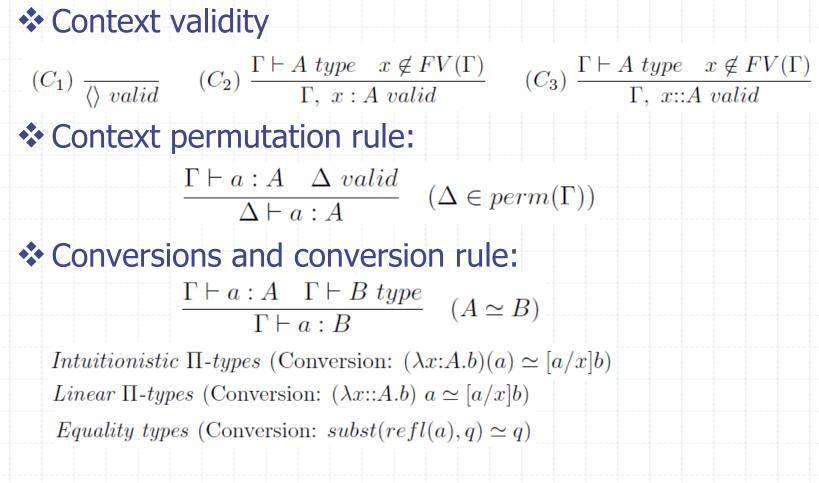
 $\frac{\Gamma \vdash a : A}{\Gamma \vdash refl(a) : Eq_A(a, a)}$ 

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

# Variable Typing

 $\Gamma, x : A, \Gamma' valid$  (for all  $y :: \Gamma_y \in \Gamma, y \in D_{\Gamma}(x)$ )  $\Gamma'$  intuitionistic  $(\overline{:} \in \{:, ::\})$  $\Gamma, x:A, \Gamma' \vdash x : A$ where \*  $D_{\Gamma}(x)$  is defined as:  $\mathbf{x} \in \mathsf{D}_{\mathsf{\Gamma}}(\mathsf{x});$ ♦ For any y∈D<sub>Γ</sub>(x), FV(Γ<sub>v</sub>) ⊆ D<sub>Γ</sub>(x). Examples: ✤ Judgements derivable intuitionistically are derivable. \* x::A,y:B(x) |- x:A and x::A,y:B(x) |- y:B(x) are derivable since  $x \in B(x)$ . \* x::A, x'::A, y:B(x) | - y : B(x) is *not* derivable if  $x' \notin B(x)$ .

### Other Rules (for completeness)



May 2016

**TYPES 2016** 

### Weak Linearality

- Defn (essential occurrences) Let \(\Gamma\) a:A. The multiset \(E\_\Gamma\) of variables essentially occurring in a under \(\Gamma\) is inductively defined as follows (Eq-types omitted):
  - \* Variable typing:  $E_{\Gamma, x:A, \Gamma'}(x) = D_{\Gamma, x:A, \Gamma'}(x)$
  - \*  $\lambda$ -typing:  $E_{\Gamma}(\lambda x : A.b) = E_{\Gamma,x:A}(b) \setminus \{x\}$
  - \* Intuitionistic applications:  $E_{\Gamma}(f(a)) = E_{\Gamma}(f) \cup E_{\overline{\Gamma}}(a)$
  - Linear applications:  $E_{Merge(\Gamma;\Delta)}(f \ a) = E_{\Gamma}(f) \cup E_{\Delta}(a)$
- Theorem (weak linearality)

In LDTT, every linear variable occurs essentially for exactly once in a well-typed term. Formally,

$$\Gamma$$
, y::B,  $\Gamma' \mid -a : A \rightarrow y \in E_{\Gamma,y::B,\Gamma'}(a)$  only once.

# Implementation

#### Type checking algorithm

- ✤ Follows the traditional algorithm for type inference/checking.
- Decidability, if assuming meta-theoretic results (expected).
- Prototype implementation in Haskell
  - ✤ Merging oprns correspond to splitting oprns.
  - Available online: https://github.com/yveszhang/ldtyping

# **Related Work**

#### Work on linearity in dependent types

- Eg, (Pfenning et al, I&C02), (Krishnaswami et al, POPL15), (Vákár, FoSSaCS 15)
- ✤ Lambek calculus with dependent types (Luo, TYPES 2015)
- ✤ Types in all above are non-dependent on linear/Lambek variables

#### McBride 2016 (Walder Festschrift)

More general setting: considering "prices" in {0,1,w}:

 $ρ_1 x_1 : A_1, ..., ρ_n x_n : A_n \mid - ρ a:A$ 

and different  $\Pi$ -types ( $\rho x:A$ ) $\rightarrow$ B:

- ↔ ( $\omega$ x:A)→B corresponds to intuitionistic  $\Pi$ -types
- ↔ (1x:A)→B corresponds to linear  $\Pi$ -types
- ✤ Type dependency B[x] only on "0-priced" variables x.
- Independent with the current work and comparison to be done.

### **Future Work**

# LDTT: allowing types to depend on linear variables

- Simplicity
  - LDTT gives a "straightforward" extension with linearality
  - cf, McBride's work, analysis to be done
- ✤ Examples of reasoning
  - ✤ to be done with our prototype implementation

Extension to other linear/Lambek type constructors