

# Type theory and logic

## Lecture II: dependent type theory

2 July 2014

柯向上

Department of Computer Science  
University of Oxford

Hsiang-Shang.Ko@cs.ox.ac.uk

## Indexed families of sets (predicates)

Common mathematical statements involve predicates and universal/existential quantification.

For example: “For all natural number  $x : \mathbb{N}$ , if  $x$  is not zero, then there exists  $y : \mathbb{N}$  such that  $x$  is equal to  $1 + y$ .”

In type theory, a predicate on  $A$  has type  $A \rightarrow \mathcal{U}$  — a *family of sets* indexed by the domain  $A$ . For example:

$$\vdash \lambda x. \text{“if } x \text{ is zero then } \mathbf{0} \text{ else } \mathbf{1}\text{”} : \mathbb{N} \rightarrow \mathcal{U}$$

**Remark.** The above treatment is in fact unfounded in our current theory. Why?

## Dependent product types (universal quantification)

- Formation:

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash B : A \rightarrow \mathcal{U}}{\Gamma \vdash \Pi A B : \mathcal{U}} \text{ (}\Pi\text{F)}$$

- Introduction:

$$\frac{\Gamma, x : A \vdash t : B x}{\Gamma \vdash \lambda x. t : \Pi A B} \text{ (}\Pi\text{I)}$$

- Elimination:

$$\frac{\Gamma \vdash f : \Pi A B \quad \Gamma \vdash a : A}{\Gamma \vdash f a : B a} \text{ (}\Pi\text{E)}$$

**Notation.** We usually write  $\Pi[x : A] B x$  for  $\Pi A B$ , regarding ' $\Pi[x : A]$ ' as a quantifier.

**Exercise.** Let  $\Gamma := A : \mathcal{U}, B : \mathcal{U}, C : A \rightarrow B \rightarrow \mathcal{U}$ . Derive

$$\Gamma \vdash \_ : (\Pi[x : A] \Pi[y : B] C x y) \rightarrow \Pi[y : B] \Pi[x : A] C x y$$

## Dependent sum types (existential quantification)

- Formation:

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash B : A \rightarrow \mathcal{U}}{\Gamma \vdash \Sigma A B : \mathcal{U}} \quad (\Sigma F)$$

- Introduction:

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B a}{\Gamma \vdash (a, b) : \Sigma A B} \quad (\Sigma I)$$

- Elimination:

$$\frac{\Gamma \vdash p : \Sigma A B}{\Gamma \vdash \text{fst } p : A} \quad (\Sigma EL) \quad \frac{\Gamma \vdash p : \Sigma A B}{\Gamma \vdash \text{snd } p : B (\text{fst } p)} \quad (\Sigma ER)$$

**Notation.** We usually write  $\Sigma[x : A] B x$  for  $\Sigma A B$ .

**Exercise.** Let  $\Gamma := A : \mathcal{U}, B : \mathcal{U}, C : A \rightarrow B \rightarrow \mathcal{U}$ . Derive

$$\Gamma \vdash \_ : (\Sigma[p : A \times B] C (\text{fst } p) (\text{snd } p)) \rightarrow \Sigma[x : A] \Sigma[y : B] C x y$$

## Computation

Let  $\Gamma := A : \mathcal{U}, B : A \rightarrow \mathcal{U}, C : A \rightarrow \mathcal{U}$ . Try to derive

$$\Gamma \vdash \_ : (\Pi[p : \Sigma A B] C (\text{fst } p)) \rightarrow \Pi[x : A] B x \rightarrow C x$$

... and you should notice some problems.

So far we have been concentrating on the *statics* of type theory; here we need to formally invoke the *dynamics* of the theory.

## Equality judgements

We introduce a new kind of judgements stating that two terms should be regarded as the same during typechecking:

$$\Gamma \vdash t = u \in A$$

for which we also have a well-formedness requirement that  $A$  and everything appearing on the right of the colons in  $\Gamma$  are judged to be sets, and  $t$  and  $u$  are judged to be elements of  $A$ .

## Computation rules

For each set, (when applicable) we specify additional *computation rules* stating that eliminating an introductory term yields a component of the latter. This is the type-theoretic formulation of *Gentzen's inversion principle*.

For example, for product types we have two computation rules:

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B}{\Gamma \vdash \text{fst}(a, b) = a \in A} (\times\text{CL}) \quad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B}{\Gamma \vdash \text{snd}(a, b) = b \in B} (\times\text{CR})$$

## More computation rules

$$\frac{\Gamma, x : A \vdash t : B \quad \Gamma \vdash a : A}{\Gamma \vdash (\lambda x. t) a = t[a/x] \in A \rightarrow B} (\rightarrow C)$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash f : A \rightarrow C \quad \Gamma \vdash g : B \rightarrow C}{\Gamma \vdash \text{case}(\text{left } a) fg = fa \in C} (+CL)$$

$$\frac{\Gamma \vdash b : B \quad \Gamma \vdash f : A \rightarrow C \quad \Gamma \vdash g : B \rightarrow C}{\Gamma \vdash \text{case}(\text{right } b) fg = gb \in C} (+CR)$$



## More computation rules

$$\frac{\Gamma, x : A \vdash t : B \quad \Gamma \vdash a : A}{\Gamma \vdash (\lambda x. t) a = t[a/x] \in B a} \text{ (}\Pi\text{C)}$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B a}{\Gamma \vdash \text{fst}(a, b) = a \in A} \text{ (}\Sigma\text{CL)}$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B a}{\Gamma \vdash \text{snd}(a, b) = b \in B a} \text{ (}\Sigma\text{CR)}$$

## Equivalence rules

Judgemental equality is an equivalence relation.

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t = t \in A} \text{ (refl)}$$

$$\frac{\Gamma \vdash t = u \in A}{\Gamma \vdash u = t \in A} \text{ (sym)}$$

$$\frac{\Gamma \vdash t = u \in A \quad \Gamma \vdash u = v \in A}{\Gamma \vdash t = v \in A} \text{ (trans)}$$

## Congruence rules

We need a congruence rule for each constant we introduce:

$$\frac{\Gamma \vdash a = a' \in A \quad \Gamma \vdash b = b' \in B}{\Gamma \vdash (a, b) = (a', b') \in A \times B}$$

$$\frac{\Gamma \vdash p = p' \in A \times B}{\Gamma \vdash \text{fst } p = \text{fst } p' \in A} \quad \frac{\Gamma \vdash p = p' \in A \times B}{\Gamma \vdash \text{snd } p = \text{snd } p' \in B}$$

$$\frac{\Gamma, x : A \vdash t = t' \in B}{\Gamma \vdash \lambda x. t = \lambda x. t' \in A \rightarrow B}$$

$$\frac{\Gamma \vdash f = f' \in A \quad \Gamma \vdash a = a' \in A}{\Gamma \vdash f a = f' a' \in B}$$

... and similar rules for left, right, case, and absurd.

## Conversion rule

Once we establish that two sets are judgementally equal, we can transfer terms between the two sets.

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash A = B \in \mathcal{U}}{\Gamma \vdash t : B} \text{ (conv)}$$

**Exercise.** Finish deriving

$$\Gamma \vdash \_ : (\Pi[p : \Sigma A B] C(\text{fst } p)) \rightarrow \Pi[x : A] B x \rightarrow C x$$

(where  $\Gamma := A : \mathcal{U}, B : A \rightarrow \mathcal{U}, C : A \rightarrow \mathcal{U}$ ).

## More congruence rules

We can state congruence rules for dependent products and sums only after we have the conversion rule. Why?

$$\frac{\Gamma \vdash a = a' \in A \quad \Gamma \vdash b = b' \in B a}{\Gamma \vdash (a, b) = (a', b') \in \Sigma A B}$$

$$\frac{\Gamma \vdash p = p' \in \Sigma A B}{\Gamma \vdash \text{fst } p = \text{fst } p' \in A} \quad \frac{\Gamma \vdash p = p' \in \Sigma A B}{\Gamma \vdash \text{snd } p = \text{snd } p' \in B(\text{fst } p)}$$

$$\frac{\Gamma, x : A \vdash t = t' \in B x}{\Gamma \vdash \lambda x. t = \lambda x. t' \in \Pi A B}$$

$$\frac{\Gamma \vdash f = f' \in A \quad \Gamma \vdash a = a' \in A}{\Gamma \vdash f a = f' a' \in B a}$$

# Universes

In our current theory, to form types like  $A \rightarrow \mathcal{U}$  where  $A : \mathcal{U}$ , we need to assume  $\mathcal{U} : \mathcal{U}$ . However, this assumption — called *impredicativity* — was shown to lead to inconsistency by Jean-Yves Girard. (This result is commonly referred to as *Girard's paradox*.)

We thus need to introduce a *predicative* hierarchy of universes  $\mathcal{U}_0, \mathcal{U}_1, \dots$ , up to infinity.

- Smaller universes are elements of larger universes.

$$\frac{}{\Gamma \vdash \mathcal{U}_i : \mathcal{U}_{i+1}} (\mathcal{U}\text{F})$$

- Elements of smaller universes are also elements of larger universes. This is called *cumulativity*.

$$\frac{\Gamma \vdash A : \mathcal{U}_i}{\Gamma \vdash A : \mathcal{U}_{i+1}} (\text{cum})$$

## Universe polymorphism

Formation rules and elimination rules have to be revised to work across the universe hierarchy.

For example, the formation rule for coproducts becomes

$$\frac{\Gamma \vdash A : \mathcal{U}_i \quad \Gamma \vdash B : \mathcal{U}_j}{\Gamma \vdash A + B : \mathcal{U}_{\max i j}} (+F)$$

The elimination rule for coproducts is

$$\frac{\Gamma \vdash q : A + B \quad \Gamma \vdash f : A \rightarrow C \quad \Gamma \vdash g : B \rightarrow C}{\Gamma \vdash \text{case } q f g : C} (+E)$$

for which we implicitly assume the following judgements:

$$\Gamma \vdash A : \mathcal{U}_i \quad \Gamma \vdash B : \mathcal{U}_j \quad \Gamma \vdash C : \mathcal{U}_k$$

In practice, however, we can drop the indices (as if assuming  $\mathcal{U} : \mathcal{U}$ ) because they can be inferred by the Harper–Pollack algorithm.

## Exercises

- Let  $\Gamma := A : \mathcal{U}, B : \mathcal{U}, R : A \rightarrow B \rightarrow \mathcal{U}$ . Derive the *axiom of choice*:

$$\Gamma \vdash \_ : (\Pi[x:A] \Sigma[y:B] R x y) \rightarrow \Sigma[f:A \rightarrow B] \Pi[x:A] R x (f x)$$