

単純型付きラムダ計算のまとめ Church 流である。

#### 項と型の定義

$\langle \text{type} \rangle$	$::=$	$T$
		$\langle \text{base-type} \rangle$ $X$
		$\text{Arr } \langle \text{type} \rangle \langle \text{type} \rangle$
$\langle \text{term} \rangle$	$::=$	$t$
		$\text{Var } \langle \text{term-variable} \rangle$ $x$
		$\text{Fun } \langle \text{term-variable} \rangle \langle \text{type} \rangle \langle \text{term} \rangle$
		$\text{App } \langle \text{term} \rangle \langle \text{term} \rangle$
$\langle \text{context} \rangle$	$::=$	$G$
		$\langle \text{context} \rangle, (\langle \text{term-variable} \rangle, \langle \text{type} \rangle)$

$\langle \text{variable} \rangle$  に対しては形無しラムダ計算と同様の性質が要求されるが、単純型付きラムダ計算では  $\langle \text{base-type} \rangle$  に対しても「等しさが決定できること」などが要求される。これがあれば  $\langle \text{type} \rangle$  の元について合成が定義できる。これらに対して次のものが型なしラムダ計算と同様にして定義される。

- 自由変数・束縛変数・単純な代入・捕縛を回避した代入
- $\alpha$  同値関係・ $\alpha$  同値決定
- $\beta$  変換関係・正規形・正規形決定・ $\beta$  変換（評価戦略による）
- $\beta$  簡約関係・ $\beta$  簡約（評価戦略による）
- $\beta$  同値関係・ $\beta$  同値計算

これらに加えて次のものが定義される。

- 型判断 (type-judgement) :  $\text{context} \rightarrow \text{term} \rightarrow \text{type} \rightarrow \text{Prop}$   
 $:=$  以下のもので生成

#### 型判断の定義

$$\begin{array}{c}
 \frac{\Gamma \vdash t : A}{\Gamma, \_ \vdash t : A} \text{ weak} \\
 \\
 \frac{}{\Gamma, (x, A) \vdash x : A} \text{ Var} \\
 \\
 \frac{\Gamma, (x : A) \vdash M : B}{\Gamma \vdash \text{Fun } (x : A) M : A \rightarrow B} \text{ Fun} \\
 \\
 \frac{\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash N : A}{\Gamma \vdash \text{App } M N : B} \text{ App}
 \end{array}$$

- 型を推論する関数  $\text{typeOf} : \text{context} \rightarrow \text{term} \rightarrow \text{option}\langle \text{type} \rangle$   
 $:= \text{typeOf } G t := t$  について
  - $\text{Var } x$  のとき  $G(x)$
  - $\text{Fun } (x : A) M$  のとき  $A \rightarrow \text{typeOf } (G, (x : A)) M$

- $\text{App } M N$  のとき  $\text{typeOf } G M$  と  $\text{typeOf } G N$  の合成。
- !  $G \vdash x : \text{typeOf } x$
- ! 型は存在すれば一意
- ! 型が存在すれば強正規化可能
- 型判断を決定する関数  $\text{isTypable} : \text{context} \rightarrow \text{term} \rightarrow \text{type} \rightarrow \text{Bool}$   
 $\text{:=}$  型を推論する関数を使う。
- † 型判断を決定する関数を普通に定義しようとすると、 $\text{App}$  の部分で型推論が必要になりつまるので基本的に型判断を決定する関数は推論する関数を用いる。