CSCI 7000-11 Spring 23

Return to types

* Let's revisit STLC
* Add extensions: Sum types, Lists, Records, Subtyping
* Polymorphism (System F)

* types $A, B ::= \text{Unit} \mid A \to B \mid A \times B$

* terms $e_1, e_2 ::= e_1\ e_2 \mid \langle e_1, e_2 \rangle \mid \text{fst}\ e_1 \mid \text{snd}\ e_2 \mid () \mid \lambda(x:A).\ e \mid x \mid \text{fix}\ e$

$\Omega = (\lambda x.\ x\ x)\ (\lambda x.\ x\ x)$

$= (\lambda(x:?).\ x\ x)\ (\lambda x.\ x\ x)$

$(T_1 \to T_2)$

$(T_1 \to T_2) \to T_3$

$T_{11} \to T_{12} \to T_2$

$$\boxed{\Gamma \vdash e : T}$$

Typing judgement

$$\frac{\Gamma \vdash e_1 : A \to T \qquad \Gamma \vdash e_2 : A}{\Gamma \vdash e_1\ e_2 : T}$$

$(\lambda(x : \text{int}).\ x + 1)\ 3$

$\text{int} \to \text{int}$ $\qquad$ int

$$\frac{(x : A) :: \Gamma \vdash e : B}{\Gamma \vdash (\lambda(x : A).\ e) : A \to B}$$

$(\lambda(x : A).\ e)$: Function

$A \to B$: function type

$$\frac{x, int :: [] \vdash x+1 : int}{[] \vdash (\lambda (x : int).\ x+1) : int \to int}$$

Γ : (string * type) list

$$\frac{\Gamma(x) = T}{\Gamma \vdash x : T}$$

int
bool

$\lambda f.\ e$

$$\frac{\Gamma \vdash e : (A \to A) \to A \to A}{\Gamma \vdash (fix\ e) : A \to A}$$

$$fact = \lambda n.\ \text{if } n = 0 \text{ then } 1 \text{ else } n * fact\ (n-1)$$

$(int \to int) \to int \Rightarrow int$

$$\text{fix}\ (\lambda f.\ \lambda n.\ \text{if}\ n = 0\ \text{then}\ 1\ \text{else}\ n * f\ (n-1))$$

$$\text{int} \to \text{int}$$

$$e_1, e_2 ::= \ldots$$

$$\mid \text{let}\ x = e_1\ \text{in}\ e_2$$

$$\frac{\Gamma \vdash e_1 : A \qquad (x, A) :: \Gamma \vdash e_2 : B}{\Gamma \vdash \text{let}\ x = e_1\ \text{in}\ e_2 : B}$$

$$\{\ \text{let}\ x = 0\ \text{in}\ \text{if}\ x = 0\ \text{then}\ 2\ \text{else}\ 2\ \}$$

$$\text{int}$$

$$\frac{\Gamma \vdash e_1 : A \qquad (f, A) :: \Gamma \vdash e_2 : B}{\Gamma \vdash \text{let rec } f = e_1 \text{ in } e_2 : B}$$

$$\frac{(f, A \to A) :: \Gamma \vdash e_1 : A \to A \qquad (f, A \to A) :: \Gamma \vdash e_2 : B}{\Gamma \vdash \text{let rec } f = e_1 \text{ in } e_2 : B}$$

let rec f = $\lambda(x : int).$ ~~~~

in ~~~~

(or)

let rec (f : A → B) = $\lambda(x : A)$ ...

in ....

Extensions

$e_1, e_2 ::= \ldots\ldots$

| left [T_2] e
| right [T_1] e
| (match e with
  | Left x → e_1
  | Right y → e_2)

(left 2) — get right

inl
inr

OCAML

```
type ('a, 'b) Either =
  | Left of 'a
  | Right of 'b
```

('a + 'b)

```
let x = Left (String) 2 in
```

Left [String] 2

Type Annotation

$e_1 ::= \text{Nil}[T]$

$\quad\quad | \ \text{Cons}\ e_1\ e_2$

let $x = \text{Nil}[\text{int}]$