

# Comp 411 Notes

## System F

Scribe: Roumen Kaiabachev

April 25, 2005

### 1 Polymorphic lists

$$t ::= X \mid t \rightarrow t \mid \forall X.t$$

$$\text{Bool} = \text{True of unit} \mid \text{False of unit} \quad \text{Bool} = \forall X.X \rightarrow X \rightarrow X$$

$$\text{Nat} = \text{Succ of nat} \mid \text{Zero of unit} \quad \text{CNat} = \forall X.(X \rightarrow X) \rightarrow X \rightarrow X$$

What is the correspondence between the types of the above two? *Succ of nat* is of type  $\text{nat} \rightarrow \text{nat}$  and *Zero of unit* is of type  $\text{nat}$ . So  $(X \rightarrow X)$  in the typing for *Nat* corresponds to the first  $X$  in the typing for *Bool*.

Now let's give typing for an integer list in System F:

$$\begin{aligned} \text{intList} &= \text{Cons of int} * \text{intList} \mid \text{Nil} \\ \text{CintList} &= \forall X.(\text{int} \rightarrow X \rightarrow X) \rightarrow X \rightarrow X \end{aligned}$$

How about polymorphic lists?

$$\begin{aligned} \text{nil} &= \lambda X.\lambda f : \text{int} \rightarrow X \rightarrow X.\lambda z : X.z \\ \text{cons} &: \text{int} \rightarrow \text{CintList} \rightarrow \text{CintList} \\ \text{cons} &= \lambda i : \text{int}.\lambda l.\text{CintList}.\lambda X.\lambda f : \text{int} \rightarrow X \rightarrow X.\lambda z : X.f\ i(l[X]\ f\ z) \end{aligned}$$

In the book this is given as

$$\text{cons} = \lambda \text{hd} : \text{int}.\lambda \text{tl}.\text{CintList}.\lambda X.\lambda \text{cons} : \text{int} \rightarrow X \rightarrow X.\lambda \text{nil} : X.\text{cons}\ \text{hd}(\text{tl}[X]\ \text{cons}\ \text{nil})$$

What would happen if we wanted to encode function spaces:

$$\begin{aligned} \text{val} &= \text{Int of int} \mid \text{Fun of val} \rightarrow \text{val} \\ \text{Cval} &= \forall X.(\text{int} \rightarrow X) \rightarrow ((X \rightarrow X) \rightarrow X) \rightarrow X \\ \text{int} &: \text{int} \rightarrow \text{val} \\ \text{int} &= \lambda i : \text{int}.\lambda X.\lambda f_{\text{int}} : (\text{int} \rightarrow X).\lambda f_{\text{fun}} : (X \rightarrow X) \rightarrow X.f_{\text{int}}\ i \end{aligned}$$

$$\begin{aligned} \text{fun} &: (\text{val} \rightarrow \text{val}) \rightarrow \text{val} \\ \text{fun} &= \lambda f : (\text{val} \rightarrow \text{val}).\lambda X.\lambda f_{\text{int}} : (\text{int} \rightarrow X).\lambda f_{\text{fun}} : (X \rightarrow X) \rightarrow X.\text{SOMETHING}. \end{aligned}$$

HW: Write a fold operator in place of **SOMETHING**.