1 Terms

$$\begin{array}{llll}
t, u & ::= & & \\
 & | & x & \text{(variable)} \\
 & | & \textbf{fun}\ (x : T) = t & \text{(abstraction)} \\
 & | & t\, u & \text{(function application)} \\
 & | & \textbf{let}\ x = t\ \textbf{in}\ u & \text{(let binding)} \\
 & | & \textbf{fun}\ [X] = t & \text{(type abstraction)} \\
 & | & t[T] & \text{(type application)} \\
 & | & (t : T) & \text{(type annotation)}
\end{array}$$

2 Types

$$\begin{array}{llll}
S, T & ::= & & \\
 & | & X & \text{(type variable)} \\
 & | & S \to T & \text{(function type)} \\
 & | & \forall X.T & \text{(polymorphic type)} \\
 & | & T_0 * ... * T_k & \text{(tuples)}
\end{array}$$