9. Simply Typed Lambda-Calculus (Types and Programming Languages)

Kwanghoon Choi

Software Languages and Systems Laboratory
Chonnam National University

Week 6 - Appendix

The syntax & operational semantics of λ-calculus

Syntax:

Term `t`	::=	`x`	variable
	|	λ`x.t`	abstraction
	|	`t t`	application
Value `v`	::=	λ`x.t`	

Evaluation:

$$\frac{\texttt{t1} \rightarrow \texttt{t1'}}{\texttt{t1 t2} \rightarrow \texttt{t1' t2}} \quad \text{(E-App1)}$$

$$\frac{\texttt{t2} \rightarrow \texttt{t2'}}{\texttt{v1 t2} \rightarrow \texttt{v1 t2'}} \quad \text{(E-App2)}$$

$$(\lambda\texttt{x. t})\ \texttt{v} \rightarrow [\texttt{x} \mapsto \texttt{v}]\ \texttt{t} \quad \text{(E-AppAbs)}$$

The type system for λ-calculus

Pure simply typed lambda-calculus ($\lambda_{\to}$)

Types	T	::=	Nat	| Bool	|	T $\to$ T
Terms	t	::=	x	| λ x:T.t	|	t t
Values	v	::=	λ x:T.t			

Typing rules

$$\frac{\Gamma(x)=T}{\Gamma \vdash x : T} \quad \text{(T-Var)}$$

$$\frac{\Gamma,\ x{:}T1 \vdash t : T2}{\Gamma \vdash \lambda x.t : T1 \to T2} \quad \text{(T-Abs)}$$

$$\frac{\Gamma \vdash t1 : T1 \to T2 \qquad \Gamma \vdash t2 : T1}{\Gamma \vdash t1\ t2 : T2} \quad \text{(T-App)}$$