Name:

CS 421 — Type Semantics Rules (Monotype Version)

The Rules

The Language

L	::=	$\lambda x.L$	abstractions
	|	$L\ L$	applications
	|	`let` $x = L$ `in` L	Let expressions
	|	`if` L `then` L `else` L	If expressions
	|	E	expressions
E	::=	x	variables
	|	n	integers
	|	b	booleans
	|	$E \oplus E$	integer operations
	|	$E \sim E$	integer comparisons
	|	$E \,\&\&\, E$	boolean and
	|	$E \,\|\|\, E$	boolean or

The Type Rules

Constants

$$\frac{}{\Gamma \vdash n : \texttt{int}}\ \textsc{Const}, \text{ when } n \text{ is an integer.}$$

Similarly for `True` and `False`.

Variables

$$\frac{}{\Gamma \vdash x : \alpha}\ \textsc{Var}, \text{ when } x : \alpha \in \Gamma$$

Binary Arithmetic

$$\frac{\Gamma \vdash e_1 : \texttt{int} \qquad \Gamma \vdash e_2 : \texttt{int}}{\Gamma \vdash e_1 \oplus e_2 : \texttt{int}}\ \textsc{BinOp}$$

Integer Relations

$$\frac{\Gamma \vdash e_1 : \texttt{int} \qquad \Gamma \vdash e_2 : \texttt{int}}{\Gamma \vdash e_1 \sim e_2 : \texttt{bool}}\ \textsc{RelOp}$$

If

$$\frac{\Gamma \vdash e_1 : \texttt{bool} \qquad \Gamma \vdash e_2 : \alpha \qquad \Gamma \vdash e_3 : \alpha}{\Gamma \vdash \texttt{if}\ e_1\ \texttt{then}\ e_2\ \texttt{else}\ e_3 : \alpha}\ \textsc{If}$$

Application

$$\frac{\Gamma \vdash e_1 : \alpha_2 \to \alpha \qquad \Gamma \vdash e_2 : \alpha_2}{\Gamma \vdash e_1\ e_2\ : \alpha}\ \textsc{App}$$

Abstraction

$$\frac{\Gamma \cup \{x : \alpha_1\} \vdash e : \alpha_2}{\Gamma \vdash \lambda x.e : \alpha_1 \to \alpha_2}\ \textsc{Abs}$$

Let

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \cup [x : \tau_1] \vdash e_2 : \tau_2}{\Gamma \vdash \texttt{let x} = e_1\ \texttt{in}\ e_2 : \tau_2}\ \textsc{Let}$$

Let Rec

$$\frac{\Gamma \cup [x : \tau_1] \vdash e_1 : \tau_1 \qquad \Gamma \cup [x : \tau_1] \vdash e_2 : \tau_2}{\Gamma \vdash \texttt{let x} = e_1\ \texttt{in}\ e_2 : \tau_2}\ \textsc{LetRec}$$