1 Syntax

1.1 Language E of Expressions

Language	Types	Expressions	Comments
E (Ch.4)	`num` `str`	x `num`$[n]$ `str`$[s]$ $e_1 + e_2$ $e_1 * e_2$ $e_1\hat{}\ e_2$ `let` x `be` e_1 `in` e_2	
ED (Ch.8.1)		`fun` $f(x:\tau_1):\tau_2\ =\ e_1$ `in` e_2 $e_1(e_2)$	Limited extension, superceded by next: First-order functions, with their names are from a different variable supply here.
EF (Ch.8.2)	$\tau_1 \to \tau_2$	$\lambda(x:\tau)e$ $e_1(e_2)$	Full functions as first-class citizens, with variable names.

1.2 Language T of Gödel total functions

♠**JB:** . . . TODO♠

1.3 Language family PCF of (general) recursive functions

Language	Types	Expressions	Comments
PCF (Ch.19)	`nat` $\tau_1 \to \tau_2$	x `z` `s`(e) `ifz` e $(e_0,\ x.e_1)$ $\lambda(x:\tau)e$ $e_1(e_2)$ `fix` $(x:\tau)$ `is` e	
FPC (Ch.20)	t `rec`t `is` τ	`fold`$_{t.\tau}(e)$ `unfold`(e)	Full functions as first-class citizens, with variable names.

2 Typing

Language	Rules	Comments
E (Ch.4)	$\dfrac{}{\Gamma, x : \tau \vdash x : \tau}$ $\quad$ $\dfrac{}{\Gamma \vdash \texttt{str}[s] : \texttt{str}}$ $\quad$ $\dfrac{}{\Gamma \vdash \texttt{num}[n] : \texttt{num}}$	Typing axiom and atoms
	$\dfrac{\Gamma \vdash e_1 : \texttt{num} \quad \Gamma \vdash e_2 : \texttt{num}}{\Gamma \vdash \texttt{plus}(e_1, e_2) : \texttt{num}}$ $\quad$ $\dfrac{\Gamma \vdash e_1 : \texttt{num} \quad \Gamma \vdash e_2 : \texttt{num}}{\Gamma \vdash \texttt{times}(e_1, e_2) : \texttt{num}}$	`num` operations
	$\dfrac{\Gamma \vdash e_1 : \texttt{str} \quad \Gamma \vdash e_2 : \texttt{str}}{\Gamma \vdash \texttt{cat}(e_1, e_2) : \texttt{str}}$ $\quad$ $\dfrac{\Gamma \vdash e : \texttt{str}}{\Gamma \vdash \texttt{len}(e) : \texttt{num}}$	conversions
	$\dfrac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma, x : \tau_1 \vdash e_2 : \tau_2}{\Gamma \vdash \texttt{let}\ x\ \texttt{be}\ e_1\ \texttt{in}\ e_2 : \tau_2}$	local binding
ED (Ch.8.1)	...	♠**JB:** TODO♠