Appendix A

LTR Syntax

The full LTR syntax is as follows:

Variables	$x,\ y$	$\in$	[a-z][a-z_]*
Positive types	$P,\ Q,\ R$	::=	$1 \mid (P \times Q) \mid (P + Q) \mid {\downarrow}N$
			$\mid \ \mu F \supset \alpha \Rightarrow t \mid \exists a : \tau \,.\, P \mid (P \wedge [\varphi])$
Negative types	N	::=	$(P \to N) \mid {\uparrow}P \mid \forall a : \tau \,.\, N \mid [\varphi] \supset N$
Functors	F	::=	$(F \oplus F) \mid \hat{P}$
Product functors	$\hat{P}$	::=	$(\hat{B} \otimes \hat{P}) \mid \mathrm{I}$
Base functors	$\hat{B}$	::=	$[P] \mid \mathrm{Id}$
Sorts	τ	::=	$\mathbb{B} \mid \mathbb{N} \mid \mathbb{Z} \mid (\tau, \tau)$
Numbers	n	$\in$	[0-9]+
Index variables	$a,\ b$	$\in$	[a-z][a-z_]*
Index terms	t	::=	$a \mid n \mid +n \mid -n \mid (t + t) \mid (t - t)$
			$\mid \ (t * t) \mid (t \,/\, t) \mid (t \,\%\, t) \mid (t, t)$
			$\mid \ \pi_1\, t \mid \pi_2\, t \mid \varphi$
Propositions	φ	::=	$(t = t) \mid (t \neq t) \mid (t < t) \mid (t \leq t)$
			$\mid \ (t > t) \mid (t \geq t) \mid (t \wedge t) \mid (t \vee t)$
			$\mid \ \neg\varphi \mid \mathrm{T} \mid \mathrm{F}$
Algebras	α	::=	$(p_1 \Rightarrow t_1 \parallel p_2 \Rightarrow t_2 \parallel \cdots)$
Sum algebra patterns	p	::=	$\mathrm{inj}_1\, p \mid \mathrm{inj}_2\, p \mid q$
Product algebra patterns	q	::=	$() \mid (o, q)$
Base algebra patterns	o	::=	$\_ \mid a \mid \mathrm{pack}(a, o)$
Values	$v,\ w$	::=	$x \mid \langle\rangle \mid \langle v, w \rangle \mid \mathrm{inj}_1\, v \mid \mathrm{inj}_2\, v$
			$\mid \ \mathrm{into}(v) \mid \{e\}$
Expressions	e	::=	$\mathrm{return}\ v \mid \mathrm{let}\ x = g;\ e \mid \mathrm{match}\ h\ \{r\}$
			$\mid \ \lambda x \,.\, e \mid \mathrm{rec}\, x : N \,.\, e \mid \mathrm{unreachable}$
Heads	h	::=	$x \mid [v : P]$
Bound expressions	g	::=	$h(v, w, \dots) \mid (e : {\uparrow}P)$

Match patterns	r	$::=$	$\langle\rangle \;\Rightarrow\; e \;\mid\; \langle x,y\rangle \;\Rightarrow\; e$
		$\mid$	$\mathrm{inj}_1\; x \;\Rightarrow\; e \;\parallel\; \mathrm{inj}_2\; x \;\Rightarrow\; e$
		$\mid$	$\mathrm{into}(x) \;\Rightarrow\; e$
REPL commands	R	$::=$	$\mathrm{let}\; x = g \;\mid\; \mathrm{let}\; x : P = v \;\mid\; \mathrm{def}\; x : N = e$
		$\mid$	$\mathrm{rec}\; x : N = e \;\mid\; \mathrm{alg}\; a = \alpha \;\mid\; \mathrm{type}\; p = P$
		$\mid$	$\mathrm{type}\; p(a : \tau) = P$

Additionally, the main executable supports including a file by typing `!!path/to/file` and line comments starting with `--` or `#`.

To support non-Unicode input devices, the following aliases are defined:

`inj`$_1$	`::=`	`inj1 | inl`		$\rightarrow$	`::=`	`~ | ->`
`inj`$_2$	`::=`	`inj2 | inr`		$\Rightarrow$	`::=`	`' | =>`
π_1	`::=`	`π1 | L`		$\parallel$	`::=`	`||`
π_2	`::=`	`π2 | R`		$\wedge$	`::=`	`&`
`Id`	`::=`	`id`		$\vee$	`::=`	`|`
$\times$	`::=`	`X`		$\neg$	`::=`	`!`
$\oplus$	`::=`	`(+)`		$\forall$	`::=`	`A`
$\otimes$	`::=`	`(×) | (X)`		$\exists$	`::=`	`E`
$\neq$	`::=`	`!=`		$\supset$	`::=`	`S`
$\leq$	`::=`	`<=`		$\top$	`::=`	`T`
$\geq$	`::=`	`>=`		$\bot$	`::=`	`F`
λ	`::=`	`fun`		$\mathbb{B}$	`::=`	`B`
μ	`::=`	`fix`		$\mathbb{N}$	`::=`	`N`
$\uparrow$	`::=`	`^`		$\mathbb{Z}$	`::=`	`Z`
$\downarrow$	`::=`	`V`				