System F_{ω}^{ρ}

Albert ten Napel

1 Syntax

l taken from an infinite set of labels

$s ::=$	(sorts)
$\quad Kind$	(sort of kinds)
$\quad s \to s$	(sort arrow)
$k ::=$	(kinds)
$\quad \kappa, \kappa_1, \kappa_2, \kappa_3$	(kind variable)
$\quad Type$	(kind of types)
$\quad Row$	(kind of rows)
$\quad Label$	(kind of labels)
$\quad k \to k$	(kind arrow)
$\quad k\ k$	(kind application)
$\quad \lambda(\kappa : s).k$	(kind abstraction)
$\quad \forall(\kappa : s).k$	(universally quantified kind)
$t ::=$	(types)
$\quad 'l$	(type labels)
$\quad \tau, \tau_1, \tau_2, \tau_3$	(type variables)
$\quad Lab$	(label constructor)
$\quad Rec$	(record constructor)
$\quad Var$	(variant constructor)
$\quad t \to t$	(type arrows)
$\quad t\ t$	(type application)
$\quad \lambda(\tau : k).t$	(type abstraction)
$\quad \forall(\tau : k).t$	(universally quantified type)
$\quad \Lambda(\kappa : s).t$	(type kind abstraction)
$\quad t[k]$	(type kind application)
$\quad \{\}$	(empty row)
$\quad \{l : t \mid t\}$	(row extension)
$\quad rowelim$	(row elimination)
$e ::=$	(terms)
$\quad \nu, \nu_1, \nu_2, \nu_3$	(variables)
$\quad 'l$	(labels)
$\quad e\ e$	(application)
$\quad \lambda(\nu : t).e$	(abstraction)
$\quad \Lambda(\tau : k).e$	(term type abstraction)
$\quad e[t]$	(term type application)
$\quad \{\}$	(empty record)

(1)

$\mathrm{id} : \forall(\tau : Type).\tau \to \tau$
$\mathrm{id} = \Lambda(\tau : Type).\lambda(\nu : \tau).\nu$

$\mathrm{const} : \forall(\tau_1 : Type).\forall(\tau_2 : Type).\tau_1 \to \tau_2 \to \tau_1$
$\mathrm{const} = \Lambda(\tau_1 : Type).\Lambda(\tau_2 : Type).\lambda(\nu_1 : \tau_1).\lambda(\nu_2 : \tau_2).\nu_1$

$\mathrm{kind\ rowelim} : \forall(\kappa_1 : Kind).\forall(\kappa_2 : Kind).\kappa_2 \to (Label \to \kappa_1 \to \kappa_2 \to \kappa_2) \to Row\ \kappa_1 \to \kappa_2$
$\mathrm{kind\ rowmap} : \forall(\kappa_1 : Kind).\forall(\kappa_2 : Kind).(\kappa_1 \to \kappa_2) \to Row\ \kappa_1 \to Row\ \kappa_2$
$\mathrm{type\ rowmap} = \Lambda(\kappa_1 : Kind).\Lambda(\kappa_2 : Kind).\lambda(\tau_1 : \kappa_1 \to \kappa_2).$
$rowelim[\kappa_1][Row\ \kappa_2]\ \{\}\ (\lambda(\tau_2 : Label).\lambda(\tau_3 : \kappa_1).\lambda(\tau_4 : Row\ \kappa_2).\{\tau_2 : \tau_1\ \tau_3 \mid \tau_4\})$