Notes on Using Reynolds' Notation in Haskell

Paul Hudak

October 8, 2003

Reynold's	**Haskell**
$P_0 \times P_1 \times \cdots \times P_n$	`(P0, P1, ..., Pn)`
$\langle x_0,\ x_1,\ \ldots,\ x_n \rangle$	`(x0, x1, ..., xn)`
π_i^n	Use `fst` and `snd` for pairs (i.e. when n=2). Otherwise, define use pattern-matching; for example π_2^4 is: `pi42 (x0,x1,x2,x3) = x2`

$f_0 \otimes \cdots \otimes f_n$	`papn (f0, ..., fn)`
	where `papn` is a family of functions indexed by `n` and defined by:
	`papn (f0, ..., fn) x = (f0 x, ..., fn x)`

$f_0 \times \cdots \times f_n$	`paptn (f0, ..., fn)`
	where `paptn` is a family of functions indexed by `n` and defined by:
	`paptn (f0, ..., fn) (x0, ..., xn) = (f0 x0, ..., fn xn)`

$(P_0 + P_1 + \cdots + P_n)_\perp$	`Qn P0 P1` $\cdots$ `Pn`
	where `Qn` is a family of polymorphic types indexed by `n` and defined by:
	`data Qn a0 ...  an = In0 a0 | In1 a1 | ...  | Inn an`

ι_i^n	`Ini`

$f_0 \oplus \cdots \oplus f_n$	`sapn (f0, ..., fn)`
	where `sapn` is a family of functions indexed by `n` and defined by:
	`sapn (f0, ..., fn) x = case x of`
	`In0 y -> f0 y`
	`...`
	`Inn y -> fn y`

$f_0 + \cdots + f_n$	`sapin (f0, ..., fn)`
	where `sapin` is a family of functions indexed by `n` and defined by:
	`sapin (f0, ..., fn) x = case x of`
	`In0 y -> In0 (f0 y)`
	`...`
	`Inn y -> Inn (fn y)`