

Apéndice: Traducción del Formalismo Básico a Haskell y Precedencia

Niveles de Precedencia

$E(x := a), .$	sustitución y evaluación
$\sqrt{\cdot}, (\cdot)^2$	raíces y potencias
$*, /$	producto y división
máx, mín	máximo y mínimo
$+, -$	suma y resta
$=, \leq, \geq$	operadores de comparación
\neg	negación
$\vee \wedge$	disyunción y conjunción
$\Rightarrow \Leftarrow$	implicación y consecuencia
$\equiv \neq$	equivalencia y discrepancia

Operadores en haskell

x^2	<code>x^2</code>
x^n	<code>x^n</code> con n entero
x^p	<code>x**p</code>
\sqrt{x}	<code>sqrt x</code>
$\sqrt[r]{x}$	<code>x**(1/r)</code>
$x \text{ máx } y$	<code>x 'max' y</code> o bien <code>max x y</code>
$x \text{ mín } y$	<code>x 'min' y</code> o bien <code>min x y</code>
$x \geq y$	<code>x >= y</code>
$x \leq y$	<code>x <= y</code>
$x = y$	<code>x == y</code>
$x \neq y$	<code>x /= y</code>
$\neg p$	<code>not p</code>
$p \wedge q$	<code>p && q</code>
$p \vee q$	<code>p q</code>
$\text{head}.xs$	<code>head xs</code>
$\text{tail}.xs$	<code>tail xs</code>
$x \triangleright xs$	<code>x:xs</code>
$xs \triangleleft x$	<code>xs ++ [x]</code>
$xs \uparrow n$	<code>take n xs</code>
$xs \downarrow n$	<code>drop n xs</code>
$xs \# ys$	<code>xs ++ ys</code>
$\#xs$	<code>length xs</code>
$xs ! n$	<code>xs !! n</code>

Tipos de los operadores

$-x$	<code>- : Num -> Num</code>
x^y	<code>(^) : Num -> Num -> Num</code>
\sqrt{x}	<code>\sqrt{\cdot} : Num -> Num -> Num</code>
$x * y$	<code>(*) : Num -> Num -> Num</code>
x / y	<code>(/) : Num -> Num -> Num</code>
$x \text{ máx } y$	<code>máx : Num -> Num -> Num</code>
$x \text{ mín } y$	<code>mín : Num -> Num -> Num</code>
$x + y$	<code>(+) : Num -> Num -> Num</code>
$x - y$	<code>- : Num -> Num -> Num</code>
$x > y$	<code>(>) : Num -> Num -> Bool</code>
$x \geq y$	<code>(>=) : Num -> Num -> Bool</code>
$x < y$	<code>(<) : Num -> Num -> Bool</code>
$x \leq y$	<code>(<=) : Num -> Num -> Bool</code>
$x = y$	<code>(=) : A -> A -> Bool</code>
$\neg p$	<code>\neg : Bool -> Bool</code>
$p \wedge q$	<code>(\wedge) : Bool -> Bool -> Bool</code>
$p \vee q$	<code>(\vee) : Bool -> Bool -> Bool</code>
$\text{head}.xs$	<code>head : [A] -> A</code>
$\text{tail}.xs$	<code>tail : [A] -> [A]</code>
$x \triangleright xs$	<code>(\triangleright) : A -> [A] -> [A]</code>
$xs \triangleleft x$	<code>(\triangleleft) : [A] -> A -> [A]</code>
$xs \uparrow n$	<code>(\uparrow) : [A] -> Int -> [A]</code>
$xs \downarrow n$	<code>(\downarrow) : [A] -> Int -> [A]</code>
$xs \# ys$	<code>(\#) : [A] -> [A] -> [A]</code>
$\#xs$	<code>\# : [A] -> Int</code>
$xs ! n$	<code>(!) : [A] -> Int -> A</code>