Typify types

Oleg Grenrus

November 21, 2014

Parser and tools for type signatures of typify. Essentially the language is dependent type theory, with omitted lambda constructor.

1 Formal syntax definition

Terminals:

$$\frac{}{\top : \mathsf{type}}\ \textsc{Any} \qquad \frac{}{\bot : \mathsf{type}}\ \textsc{Nothing} \qquad \frac{}{\mathbb{1} : \mathsf{type}}\ \textsc{Unit}$$

$$\frac{}{\texttt{number} : \mathsf{type}}\ \textsc{Number} \qquad \frac{}{\texttt{string} : \mathsf{type}}\ \textsc{String} \qquad \frac{}{\texttt{bool} : \mathsf{type}}\ \textsc{Boolean}$$

Identifiers:

$$\frac{n : \mathsf{name}}{n : \mathsf{type}}\ \textsc{Identifier}$$

Row types:

$$\frac{a : \mathsf{type} \quad b : \mathsf{type}}{a \wedge b : \mathsf{type}}\ \textsc{Conjunction} \qquad \frac{a : \mathsf{type} \quad b : \mathsf{type}}{a \vee b : \mathsf{type}}\ \textsc{Disjunction} \qquad \frac{a : \mathsf{type}}{a? : \mathsf{type}}\ \textsc{Optional}$$

Application:

$$\frac{a : \mathsf{type} \qquad b : \mathsf{type}}{a\ b : \mathsf{type}}\ \textsc{Application}$$

Special modifiers:

$$\frac{a : \mathsf{type} \quad b : \mathsf{type}}{a \times b : \mathsf{type}}\ \textsc{Product} \qquad \frac{a : \mathsf{type}}{a \ldots : \mathsf{type}}\ \textsc{Variadic} \qquad \frac{n : \mathsf{type} \quad p : \mathsf{type}}{n : p : \mathsf{type}}\ \textsc{Named}$$

Brackets:

$$\frac{p : \mathsf{type}}{[p] : \mathsf{type}}\ \textsc{Square} \qquad \frac{p : \mathsf{type}}{\{p\} : \mathsf{type}}\ \textsc{Curly}$$

Functions:

$$\frac{a : \mathsf{type} \quad b : \mathsf{type}}{a \to b : \mathsf{type}}\ \textsc{Function}$$

Figure 1: Typify syntax rules

level	name		associativity	example
8	optional	?	postfix	$a?$
7	application	$a\ b$	left associative	$a\ b\ c \equiv (a\ b)\ c$
6	conjunction	$\wedge$	associative	$(a \wedge b) \wedge c \equiv a \wedge (b \wedge c)$
5	disjunction	$\vee$	associative	$(a \vee b) \vee c \equiv a \vee (b \vee c)$
4	ellipsis	...	postfix	$a \ldots$
3	type	:	left associative	$x : a : b \equiv (x : a) : b$
2	product	$\times$	associative	$(a \times b) \times c \equiv a \times (b \times c)$
1	function	$\rightarrow$	right associative	$a \rightarrow b \rightarrow c \equiv a \rightarrow (b \rightarrow c)$

Table 1: Operator precedence

name		code variants
any	$\top$	$\top$ *
nothing	$\bot$	$\bot$ _|_
unit	$\mathbb{1}$	$\mathbb{1}$ ()
optional	?	?
conjunction	$\wedge$	$\wedge$ |
disjunction	$\vee$	$\vee$ &
ellipsis	...	. . . (either three dots, or unicode ellipsis)
type	:	:
product	$\times$	$\times$,
function	$\rightarrow$	$\rightarrow$ ->

Table 2: Special character code representations

$$
\begin{aligned}
&a \times ys : b \ldots \times c \rightarrow d \rightarrow e \\
&\quad \equiv x : a \times ys : (b \ldots) \times c \rightarrow d \rightarrow e \\
&\quad \equiv (x : a) \times (ys : (b \ldots)) \times c \rightarrow d \rightarrow e \\
&\quad \equiv ((x : a) \times \big(ys : (b \ldots)\big) \times c) \rightarrow d \rightarrow e \\
&\quad \equiv ((x : a) \times \big(ys : (b \ldots)\big) \times c) \rightarrow (d \rightarrow e) \\
&a \times y : b \rightarrow c \\
&\quad \equiv a \times (y : b) \rightarrow c \\
&\quad \equiv (a \times (y : b)) \rightarrow c
\end{aligned}
$$

$$
\begin{aligned}
&a \vee b? \wedge c\ d \ldots \rightarrow e \\
&\quad \equiv a \vee (b?) \wedge (c\ d) \ldots \rightarrow e \\
&\quad \equiv a \vee ((b?) \wedge (c\ d)) \ldots \rightarrow e \\
&\quad \equiv (a \vee ((b?) \wedge (c\ d))) \ldots \rightarrow e \\
&\quad \equiv ((a \vee ((b?) \wedge (c\ d))) \ldots) \rightarrow e
\end{aligned}
$$

Figure 2: Examples of operator precedence

$$Either\ a\ b \equiv \{type : \texttt{"left"}, value : a\} \vee \{type : \texttt{"right"}, value : b\}$$
$$flatMap : (@ : Observable\ A \times f : A \rightarrow Observable\ B \vee Event\ B \vee B) \rightarrow EventStream\ B$$

Figure 3: Examples of real world types