Functive

Henry Blanchette

Contents

1 Grammar

1.1 Formalization

$$
\begin{array}{rcl}
\textit{program} & ::= & \overline{s} \\
s & ::= & \texttt{Begin Module } n \texttt{ . } \overline{s} \texttt{ End Module } n \texttt{ .} \\
 & & \mid \texttt{ Definition } n \texttt{ : t := } e \texttt{ .} \\
 & & \mid \texttt{ Signature } n \texttt{ := t . } \mid \texttt{ Assumption } n \texttt{ : t .} \\
e & ::= & v \mid \texttt{(} e' \ e \texttt{)} \mid \texttt{(let } n \texttt{ := } e \texttt{ in } e \texttt{)} \\
 & & \mid \texttt{(fun } n \texttt{ => } e \texttt{)} \mid \texttt{(rec } n \texttt{ of } n \texttt{ => } e \texttt{)} \\
v & ::= & \textit{primitive-value} \mid n \\
t & ::= & \textit{primitive-type} \mid n \mid \texttt{(} \ t \ t \ \texttt{)} \mid \texttt{(} \ t \ e \ \texttt{)} \\
 & & \mid \texttt{(} \ t \texttt{ -> } t \ \texttt{)} \mid \texttt{(forall } n \texttt{ : } t \texttt{ , } t \texttt{)} \\
\textit{primitive-value} & ::= & z \in \mathbb{Z} \mid b \in \{\textit{true}, \textit{false}\} \mid \texttt{()} \\
\textit{primitive-type} & ::= & \texttt{integer} \mid \texttt{boolean} \mid \texttt{unit} \mid \texttt{type} \\
n & ::= & \textit{string} \mid \textit{n.string}
\end{array}
$$

1.2 Notation

- An expression with an overline may be repeated any natural number of times.