Real Numbers

October 7, 2023

Contents

1 Expression **1**

2 Operations **2**

1 Expression

The expression for a real number is very similar to that of an integer, except instead of expecting a natural number as its second component, it expects a pair of them.

$$\lambda s \lambda n \lambda d . \upsilon_2 s (\upsilon_2 n d)$$

We now however need a much more robust reduction step for the actual structure generating function. We need all congruent fractional representations to simplify to the most atomic versions of themselves, and we also need there to be only 1 expression of

0.

$$
\begin{aligned}
\mathrm{reduce}_{\mathbb{R}} = \; & \lambda n.(\\
& \quad \lambda c.(\\
& \quad\quad (\lambda g.gn(cn)g) \\
& \quad\quad (\lambda m\lambda k\lambda f.\mathrm{eq}_{\mathbb{N}}k1_{\mathbb{N}}m(\\
& \quad\quad\quad (\lambda x.(fx)(cx)f) \\
& \quad\quad\quad (\upsilon_2(\mathrm{div}_{\mathbb{N}}(m\pi_1)k)(\mathrm{div}_{\mathbb{N}}(m\pi_2)k)) \\
& \quad\quad)) \\
& \quad) \\
& \quad (\lambda x.\mathrm{gcf}_{\mathbb{N}}(\mathrm{min}_{\mathbb{N}}(x\pi_1)(x\pi_2))(\mathrm{max}_{\mathbb{N}}(x\pi_1)(x\pi_2)) \\
&) \\
\mathrm{reduce0}_{\mathbb{R}} = \; & \lambda r.0?_{\mathbb{N}}(r\pi_2\pi_1)(\upsilon_2(\mathrm{T})(\upsilon_2(0_{\mathbb{N}})(0_{\mathbb{N}})))r \\
\mathbb{R} = \; & \lambda s\lambda n\lambda d.\mathrm{reduce0}_{\mathbb{R}}(\upsilon_2 s(\mathrm{reduce}_{\mathbb{R}}(\upsilon_2 nd)))
\end{aligned}
$$

2 Operations

We now need to make sure all standard mathemetic functions are defined for this type, as this will be our final numeric type for every day use in data structures and algorithms. We will begin by defining functions that operate on unsigned fractional quantities, so that the base manipulations are ready to go, making it easier to deal with behavior branching when signs are present. Fortunately these are pretty easy to write, as we can just split it up between numerator and denominator and then reduce the fraction.

$$
\begin{aligned}
\mathrm{add}^{+}_{\mathbb{R}} = \quad & \lambda n \lambda m.\mathrm{reduce}_{\mathbb{R}}(\upsilon_2(\\
& \quad \mathrm{add}_{\mathbb{N}}(\mathrm{mul}_{\mathbb{N}}(n\pi_1)(m\pi_2))(\mathrm{mul}_{\mathbb{N}}(m\pi_1)(n\pi_2)) \\
&)(\\
& \quad \mathrm{mul}_{\mathbb{N}}(n\pi_2)(m\pi_2) \\
&)) \\
\mathrm{sub}^{+}_{\mathbb{R}} = \quad & \lambda n \lambda m.\mathrm{reduce}_{\mathbb{R}}(\upsilon_2(\\
& \quad \mathrm{sub}_{\mathbb{N}}(\mathrm{mul}_{\mathbb{N}}(n\pi_1)(m\pi_2))(\mathrm{mul}_{\mathbb{N}}(m\pi_1)(n\pi_2)) \\
&)(\\
& \quad \mathrm{mul}_{\mathbb{N}}(n\pi_2)(m\pi_2) \\
&)) \\
\mathrm{mul}^{+}_{\mathbb{R}} = \quad & \lambda n \lambda m.\mathrm{reduce}_{\mathbb{R}}(\upsilon_2(\\
& \quad \mathrm{mul}_{\mathbb{N}}(n\pi_1)(m\pi_1) \\
&)(\\
& \quad \mathrm{mul}_{\mathbb{N}}(n\pi_2)(m\pi_2) \\
&)) \\
\mathrm{max}^{+}_{\mathbb{R}} = \quad & \lambda a \lambda b.\mathrm{gt}_{\mathbb{N}}(\mathrm{mul}_{\mathbb{N}}(a\pi_1)(b\pi_2))(\mathrm{mul}_{\mathbb{N}}(b\pi_1)(a\pi_2))ab \\
\mathrm{min}^{+}_{\mathbb{R}} = \quad & \lambda a \lambda b.\mathrm{gt}_{\mathbb{N}}(\mathrm{mul}_{\mathbb{N}}(a\pi_1)(b\pi_2))(\mathrm{mul}_{\mathbb{N}}(b\pi_1)(a\pi_2))ba
\end{aligned}
$$

Division is just a reciprocal multiplication, so we can do that when we already have signs. We can now move on to defining our final all encompassing functions for numeric manupulation. These are of course not the most efficient possible implementations, however for computational theory we dont care about performance, especially when every operation can be theoretically parallelized. We're just showing that it can be done at all.

$$
\begin{aligned}
\mathrm{add}_{\mathbb{R}} &= \lambda a \lambda b.\mathrm{and}_b(a\pi_1)(b\pi_1)(\\
&\quad\ \upsilon_2(n\pi_1)(\mathrm{add}^+_{\mathbb{R}} nm) \\
&)(\\
&\quad\ \upsilon_2(\mathrm{gt}_{\mathbb{N}}(\mathrm{mul}_{\mathbb{N}}(a\pi_2\pi_1)(b\pi_2\pi_2))(\mathrm{mul}_{\mathbb{N}}(b\pi_2\pi_1)(a\pi_2\pi_2)) \\
&\quad\quad\ (a\pi_1)(b\pi_1) \\
&\quad\ )(\\
&\quad\quad\ \mathrm{sub}^+_{\mathbb{R}}(\mathrm{max}^+_{\mathbb{R}}(a\pi_2)(b\pi_2))(\mathrm{min}^+_{\mathbb{R}}(a\pi_2)(b\pi_2)) \\
&\quad\ ) \\
&) \\
\mathrm{sub}_{\mathbb{R}} &= \lambda a \lambda b.\mathrm{add}_{\mathbb{R}} a(\upsilon_2(\mathrm{not}_b(b\pi_1))(b\pi_2)) \\
\mathrm{mul}_{\mathbb{R}} &= \lambda a \lambda b.\upsilon_2(\mathrm{eq}_b(a\pi_1)(b\pi_1))(\mathrm{mul}^+_{\mathbb{R}}(a\pi_2)(b\pi_2)) \\
\mathrm{div}_{\mathbb{R}} &= \lambda a \lambda b.\mathrm{mul}_{\mathbb{R}} a(\upsilon_2(b\pi_1)(\upsilon_2(b\pi_2\pi_2)(b\pi_2\pi_1))) \\
\mathbb{N} \to \mathbb{R} &= \lambda n.\upsilon_2 \mathrm{T}(\upsilon_2 n 1_{\mathbb{N}}) \\
\mathbb{Z} \to \mathbb{R} &= \lambda n.\upsilon_2(z\pi_1)(\upsilon_2(z\pi_2)1_{\mathbb{N}}) \\
0?_{\mathbb{R}} &= \lambda r.0?_{\mathbb{N}}(r\pi_2\pi_1) \\
\mathrm{lte}_{\mathbb{R}} &= \lambda a \lambda b.0?_{\mathbb{R}}(\mathrm{sub}^+_{\mathbb{R}}(a\pi_2)(b\pi_2)) \\
\mathrm{eq}_{\mathbb{R}} &= \lambda a \lambda b.\mathrm{and}_b(\mathrm{lte}_{\mathbb{R}} ab)(\mathrm{lte}_{\mathbb{R}} ba) \\
\mathrm{lt}_{\mathbb{R}} &= \lambda a \lambda b.\mathrm{and}_b(\mathrm{lte}_{\mathbb{R}} ab)(\mathrm{not}_b(\mathrm{eq}_{\mathbb{R}} ab)) \\
\mathrm{gt}_{\mathbb{R}} &= \lambda a \lambda b.\mathrm{not}_b(\mathrm{lte}_{\mathbb{R}} ab) \\
\mathrm{gte}_{\mathbb{R}} &= \lambda a \lambda b.\mathrm{not}_b(\mathrm{lt}_{\mathbb{R}} ab) \\
\mathrm{max}_{\mathbb{R}} &= \lambda a \lambda b.\mathrm{gt}_{\mathbb{R}} abab \\
\mathrm{min}_{\mathbb{R}} &= \lambda a \lambda b.\mathrm{gt}_{\mathbb{R}} abba \\
\mathrm{abs}_{\mathbb{R}} &= \lambda r.\mathbb{R}T(r\pi_2) \\
\mathrm{neg}_{\mathbb{R}} &= \lambda r.\mathbb{R}(\mathrm{not}_b(r\pi_1))(r\pi_2)
\end{aligned}
$$