## Question 2

In this report, the author discusses multiple aspects on different stages of the design of Programming Languages, including the design, documentation, and debugging.

In Section 3.5, the author discussed about the readability, and argued that in practice it's very unlikely that the output of a computer will be more readable than its input. While I agree that readability is important, and I personally enjoy using those languages with good code readability, there are indeed some examples where the output of the program is easier to interpret than the original code, such as LaTeX, or Postscript, where the output pdf is more intuitive than the source codes. Other examples include HTML/CSS and other UI related languages.

In Section 6, the author discussed the importance on the efficiency and expressiveness of arithmetic expressions. I've been using both Python and Julia for numerical computation tasks. While Python is very convenient based on the abundance of third-party libraries, I personally found that Julia to be easier to use in term of the language design, as it supports more mathematical operations on the syntactic level.

## Question 3

In order to implement the division operator, first the (non-negative) rational numbers should be incorporated as normal forms. The value of variables, and the states are now functions $L \to \mathbb{Q}$, or $\mathbb{Q}_+$. In addition, we need the following extra inference rule for division introduction, which has an assumption that certifies the divider is not zero.

$$\frac{\langle e_1,\ \sigma\rangle \Downarrow n_1 \qquad \langle e_2,\ \sigma\rangle \Downarrow n_2 \qquad \langle e_2 = 0,\ \sigma\rangle \Downarrow \texttt{false}}{\langle e_1/e_2,\ \sigma\rangle \Downarrow n_1/n_2}\ \texttt{divisionI}$$

## Question 4

One thing special about the `let` command is that while all other variables may be updated, $x$ itself must not be updated at the end. Therefore we have the introduction rule as follows:

$$\frac{\langle x,\ \sigma\rangle \Downarrow n_0 \qquad \langle e,\ \sigma\rangle \Downarrow n_1 \qquad \langle c,\ \sigma[x := n_1]\rangle \Downarrow \sigma'}{\langle \texttt{let}\ \ x = e\ \ \texttt{in}\ \ c,\ \sigma\rangle \Downarrow \sigma'[x := n_0]}\ \texttt{letI}$$

## Question 5

- New redex: `let` $x = n$ `in` $c$.
- Local reduction rule:

$$\langle \texttt{let}\ \ x = n\ \ \texttt{in}\ \ c,\ \sigma\rangle \longrightarrow \langle c;\ x := n,\ \sigma[x := n]\rangle$$

Question assigned to the following page: 5

- Context decomposition: Consider $\texttt{let}\ x = e\ \texttt{in}\ c$.
  - if $e$ is not $n$: $H =''\ \texttt{let}\ x = H_1\ \texttt{in}\ c''$, $e = H_1[r]$
  - if $e = n$: $H = \bullet$, $r =''\ \texttt{let}\ x = n\ \texttt{in}\ c''$