Exercise 1F-2

In his paper, Hoare argues that efficient object code remains an objective criterion for good language design. He mentions that while arguments about computer hardware becoming more and more fast and efficient that efficacy loss in language design is becoming tolerable do hold some ground in certain cases, most of the times the efficacy loss is way too severe even for better hardware. From my experience with programming for the past few years, I have come to the same conclusion as him on this matter. Hoare may have written this article back in the 70s, but the fact that the tasks we need our computers to undertake are becoming more and more advanced still holds true, and recent talks about Moore's Law not holding true anymore only makes the situation worse. An efficient language that can optimize the program code well is crucial in this case.

Hoare also argues that simplicity is a crucial principle that is often replaced by some others with the principle of modularity, which he sees as inferior. He argues that the principle of modularity, where a programmer who does not understand the entirety of the language can still get by with only understanding a small part of it, only works if the program written actually works. If it doesn't, he claims, and the programmer invoked a function of the language he is not familiar with, the consequences will be dire. I don't necessarily think this is the case any longer. The advanced technology of today allows us to easily look for help online if we find something that we do not understand while coding, like in online help sites like StackExchange. Even if we do not understand the entirety of the language, it is easy to find someone else who does and is willing to help.

Hoare's paper was written almost half a century ago. The parts of his arguments that still hold true today do so by touching on subjects that are timeless, like the need for faster and more efficient programming. However, the world has changed dramatically since 50 years ago, and many discoveries have been made that make the other parts of his arguments obsolete.

Exercise 1F-3

The most important thing about division is that the denominator cannot equal to zero. Thus the second expression may not evaluate to 0. Furthermore, IMP language only uses integers, so we need to implement integer division, which means throwing out the remainder. Thus we need to update the rules of inference to include this:

$$\frac{\langle e_1, \sigma\rangle \Downarrow n_1 \quad \langle e_2, \sigma\rangle \Downarrow n_2 \neq 0}{\left\langle \frac{e_1}{e_2}, \sigma \right\rangle \Downarrow \left\lfloor \frac{n_1}{n_2} \right\rfloor}$$

Exercise 1F-4

$$\frac{\langle e, \sigma\rangle \Downarrow n \quad \langle c, \sigma[x := n]\rangle \Downarrow \sigma'}{\langle let\ x = e\ in\ c,\ \sigma\rangle \Downarrow \sigma'\,[x := \sigma(x)]}$$

Peer Review ID: 63184849 — enter this when you fill out your peer evaluation via gradescope

Exercise 1F-5

$$\begin{aligned}
r ::= {} & x \\
& \mid n_1 + n_2 \\
& \mid x := n \\
& \mid skip;\ c \\
& \mid if\ true\ then\ c_1\ else\ c_2 \\
& \mid if\ false\ then\ c_1\ else\ c_2 \\
& \mid while\ b\ do\ c \\
& \mid let\ x = e\ in\ c
\end{aligned}$$

$$\begin{aligned}
&\langle x, \sigma\rangle \to \langle \sigma(x), \sigma\rangle \\
&\langle n_1 + n_2, \sigma\rangle \to \langle n, \sigma\rangle \quad where\ n = n_1\ plus\ n_2 \\
&\langle n_1 = n_2, \sigma\rangle \to \langle true, \sigma\rangle \quad if\ n_1 = n_2 \\
&\langle x := n, \sigma\rangle \to \langle skip, \sigma[x := n]\rangle \\
&\langle skip;\ c, \sigma\rangle \to \langle c, \sigma\rangle \\
&\langle if\ true\ then\ c_1\ else\ c_2, \sigma\rangle \to \langle c_1, \sigma\rangle \\
&\langle if\ false\ then\ c_1\ else\ c_2, \sigma\rangle \to \langle c_2, \sigma\rangle \\
&\langle while\ b\ do\ c, \sigma\rangle \to \langle if\ b\ then\ c;\ while\ b\ do\ c\ else\ skip, \sigma\rangle \\
&\langle let\ x = e\ in\ c, \sigma[x := n]\rangle \to \langle c; x := n, \sigma[x := e]\rangle
\end{aligned}$$

$$\begin{aligned}
H ::= {} & \bullet \mid n + H \\
& \mid H + e \\
& \mid x := H \\
& \mid if\ H\ then\ c_1\ else\ c_2 \\
& \mid H;\ c \\
& \mid let\ x = H\ in\ c
\end{aligned}$$

Peer Review ID: 63184849 — enter this when you fill out your peer evaluation via gradescope

1 HW1 (select all pages: your first page has your name and bookkeeping, and all others are anonymous))

- 0 pts Correct

Peer Review ID: 63184849 — enter this when you fill out your peer evaluation via gradescope

gradescope