Run Haskell Anywhere with Java and Jython

Zayd Hammoudeh (zayd.hammoudeh@sjsu.edu)

February 25, 2016

1 Running in the Java Virtual Machine

C is one of the most commonly used languages when it comes to maximizing execution performance. However, C/C++'s "write once compile anywhere" paradigm makes it less portable. In contrast, the near ubiquity of the Java Virtual Machine (JVM) allows it to be "write once run anywhere."

On many occasions, the global development community has leveraged the Java environment to allow other languages to take advantage of Java's "run anywhere" capability. Examples include: JRuby for the Ruby programming language [jru], Scala [sca], Renjin for the R programming language [ren], and Jython for the Python programming language [jyt].

Currently, there is no full implementation of Haskell in the JVM. One Haskell dialect that is runnable in Java is Frege [Wec].

In this project, I will implement a stripped down version of Haskell in the Java Virtual Machine.

References

[jru] Home jruby.org. `http://jruby.org/`. (Accessed on 02/24/2016).

[jyt] Platform specific notes. `http://www.jython.org/archive/21/platform.html`. (Accessed on 02/25/2016).

[ren] Renjin.org — about. `http://www.renjin.org/about.html`. (Accessed on 02/25/2016).

[sca] The scala programming language. http://www.scala-lang.org/. (Accessed on 02/25/2016).

[Wec] "Ingo Wechsung". Frege/frege: Frege is a haskell for the jvm. it brings purely functional programing to the java platform. https://github.com/Frege/frege. (Accessed on 02/25/2016).