

Written Assignment 5

Assigned: October 14

Due: March 21 at 1:15pm

Instructions: This assignment asks you to prepare written answers to questions on type inference. Each of the questions has a short answer. You may discuss this assignment with other students and work on the problems together. However, your write-up should be your own individual work.

Remember that written assignments are to be turned in at the start of lecture. Please write your name, your account name, your TA's name, and your section time on your homework! We need this information so that we can give you credit for the assignment and so that we can return it to you.

1. Show the full type derivation (as done in slide 53 in the lecture notes) for the following judgment:

$$\begin{array}{l} O[\text{Object}/a] \vdash \text{case } a \text{ of} \\ \quad x:\text{Int} \Rightarrow x + 5 * 4; \\ \quad z:\text{Bool} \Rightarrow \text{true}; \\ \text{esac} \end{array}$$

2. Suppose we extend the grammar for Cool with the type “**Long**” that is analogous to **Long** in Java. That is to say that a Long contains 64 bits as opposed to the standard Int that only hold 32.

Give sound typing rules that we can add to the Cool specification that relates Long to Int and Object.

3. Suppose we extend Cool with exceptions by adding two new constructs to the Cool language modeled after the Python equivalents.

$$\begin{array}{l} \text{expr} ::= \text{try expr except } ID : \text{Type} \Rightarrow \text{expr finally expr} \\ \quad | \quad \text{raise expr} \\ \quad | \quad \dots \end{array}$$

Here **try**, **except**, **finally**, and **raise** are three new terminals. “**raise expr**” returns *expr* to the closest enclosing except block that has the right type. Note that since **raise** expression returns control to a different location, we do not really care about the context in which raise is used. The type in except refers to the type of errors that can be caught. Finally refers to actions that should occur at the end of the try block or at the end of the except block if it was triggered.

Give a set of new sound typing rules that we can add to the Cool specification to accommodate these two new constructs (raise and try/except/finally). (Hint: Use a bottom type \perp which functions like $\perp \sqcup T = T$. For more details look at https://en.wikipedia.org/wiki/Bottom_type.)

4. Java includes arrays and allows inheritance. You are allowed to assign one type of an array to another so long as the type of the first array is a subtype of the second. This however, turns out to be unsound despite being able to pass Java's type checking rules. Give an example (that is code) that would pass Java's static type checking but would cause an error in execution. Furthermore describe why this error occurs.
5. Write Java code that would print “d” when ran in a dynamically scoped environment but “s” when ran in a statically scoped environment.