OO Code Samples

• Selections from solution to Assignment 2
  – Class hierarchy for Binding union
  – Sample visitor method code

• Discuss some OO design tradeoffs
  – Use of `instanceof`
  – With composite, visitor implementation of methods is not always mandated. Good idea to “build in” some core operations of a composite using the interpreter pattern. Why? Leaner (in terms of lines of code). Easier to read.
Good Commenting Conventions

• Write a javadoc description for every class, field, and non-trivial method.
• Method descriptions are informal contracts.
• Contracts should be as precise as possible. In some cases (e.g., GUI libraries), complete precision may not be feasible.
• The class assignment solutions could be better commented but I subscribe to the dictum that “brevity is a virtue”.
How to Eliminate \textit{lambda}

Goal: devise a few combinators (functions expressed in \(\lambda\)-notation with no free variables) that enable us to express all \(\lambda\)-expressions without explicitly using \(\lambda\).

Notation: let \(\lambda^*x.M\) denote \(\lambda x.M\) converted to an equivalent form that eliminates the starred \(\lambda\). Then

• \(\lambda^*x.x \rightarrow I\) (where \(I = \lambda x.x\))
• \(\lambda^*x.y \rightarrow Ky\) (where \(K = \lambda y.\lambda x.y\))
• \(\lambda^*x.(M N) \rightarrow S(\lambda^*x.M)(\lambda^*x.N)\)

\(S = \lambda x.\lambda y.\lambda z.((x z)(y z))\)

Strategy: eliminate \(\lambda\)-abstractions from inside out, one-at-a-time. Any order works. Transformation can cause exponential blow-up.

\textit{Note}: \(I\) is technically unnecessary since \(SKK = I\)
Checking the App case

\[ S \ (\lambda x. M) \ (\lambda x. N) \]
\[ = \ (\lambda x. \lambda y. \lambda z \ (x \ z) \ (y \ z)) \ (\lambda x. M) \ (\lambda x. N) \]
\[ = \ (\lambda y. \lambda z \ (\lambda x. M) \ z) \ (y \ z) ) \ (\lambda x. N) \]
\[ = \ (\lambda y. \lambda z \ (M_{x\leftarrow z}) \ (y \ z)) \ (\lambda x. N) \]
\[ = \ (\lambda z. \ (M_{x\leftarrow z}) \ ((\lambda x. N) \ z) ) \]
\[ = \ (\lambda z. \ (M_{x\leftarrow z}) \ (N_{x\leftarrow z})) = \lambda x. \ (M \ N) \quad (by \ \alpha\text{-conversion}) \]