

Higher Order Calculus – homework

Piotr Krzemiński

June 17, 2014

There are two variants of the homework: theoretical and practical. You have to pick at least one of them, solve it, and deliver the solution to pio.krzeminski@gmail.com. Deadline is 28th June 2014.

1 Practical Variant

Your task is to provide implementation of type-checker for a normal system of higher order calculus presented at the lecture.

- define representation for terms and types in higher order calculus
- provide translation of types to a normal form
- implement a type-checker for terms using inclusion rules of a normal system
- pretty printer of terms and types would be nice, but it's not necessary

Encode examples from the book connected with object-oriented encoding of binary trees (*Bin*, *UBin*, *leaf*, two leaves with joining root).

Notice that in these examples λ -abstraction & application on terms appears. Add them explicitly to the calculus (providing rules for typing them) or define appropriate translation in terms of objects.

The implementation has to be done in functional language (Haskell, OCaml, Scala, F#, etc.).

2 Theoretical Variant

Prove Lemma 20.5-2 from the book.

Lemma (Equivalence to a normal form). *If $E \vdash A :: K$, then $E \vdash A \leftrightarrow A^{nf} :: K$.*

The proof has to be in PDF file or handwritten and scanned.