

Homework # 10

due May 6, 13:00

Please turn in the answers to sections §3,4,5 by email to `scmalte@inf.ethz.ch`.

1 Reading

Please read Chapters 20 and 21 in your textbook.

2 Problems

Please do the following problems

- Exercise 20.1.2
- Exercise 20.2.1 (convert your code for the previous question)
- Exercise 21.2.2
- Exercise 21.3.4

Please check your answers against the solutions in the book. Do not turn in your answers.

3 Modeling recursion

Using the `fullisorec` checker, but *not* using the “fix” keyword or side-effects, implement a “plus” function that works on actual nats, not on church numerals. You will need to implement your own Y combinator. Put your result in file `myfix.f`. Demonstrate that it works by adding 2 and 3.

NB: The textbook (page 273) gives the wrong definition of `fix`: it will cause nontermination. (Furthermore it assumes equi-recursive types. Your code needs to work with iso-recursive types.) You need to use the call-by-value version on page 65 and add iso-recursion. A major part of this question is figuring out how to handle these aspects.

4 Proofs

Prove the type soundness of iso-recursive types with subtyping using SASyLF. Products rather than records are used to keep the complexity reasonable. Since subtyping is involved, the canonical forms lemma is a major aspect of the proof.

5 Mechanization

Continue your mechanization from Homework #8 and prove at least one non-trivial lemma or theorem in SASyLF for this system. If you didn’t start an acceptable mechanization for Homework #8, please request an alternate paper to mechanize. Please put comments in your SASyLF file that relate your mechanization to the formalism in the paper.