Homework # 8
due October 28

1 Reading

Please read Chapter 13 in your textbook.

2 Problems

Please do the following problem:

- Exercise 13.1.1, and include the effects of evaluating

  ```
  c = (lambda x: Nat . {ref x, ref x}) 0
  ```

  Explain your answer!

3 Discussion

- The textbook says (page 167) that we must include a well typing in the requirements for progress and preservation. Give two counter-examples, one for Progress (Theorem 13.5.7) and one for Preservation (Theorem 13.5.3) if they omit any mention of well typing.
- Where do store typings come from; how does one create a store typing? In particular, for a programmer wanting to know whether their program will execute without getting stuck, how can they use the progress and preservation theorems? The theorems require that we have a store typing, what store typing should one use?
- Definition 13.5.1 (and the SASyLF skeleton file) define store well-typing in a Γ context. When would this context ever be non-empty? What would it mean for a store μ to be well-typed only in a non-empty context?

4 Proofs

Complete the proof of soundness of references started in the SASyLF skeleton file. In particular

- Prove the indicated "effectiveness" lemmas needed for progress. (An *effectiveness* lemma, usually named "can-*operation*" indicates that an operation can indeed be carried out.)
- Prove the indicated helper lemmas needed for preservation, including Lemma 13.5.5 and the analogous one that handles allocation of new references.
- Complete the proof of preservation to handle the new syntactic forms. The proof already includes the cases for the simply-typed lambda calculus.