

CIS 500 Software Foundations

Homework Assignment 7

References and Exceptions

Due: Monday, November 14, 2005, by noon

Submission instructions:

You must submit your solutions electronically (in ascii, postscript, or PDF format). Electronic solutions should be submitted following the same instructions as last time; these can be found at <http://www.seas.upenn.edu/~cis500/homework.html>. Do **not** email your solutions to us.

1 Exercise Is the language of Chapter 13 (see Figure 13-1) normalizing on well-typed terms? In other words, is it the case that for any τ and μ such that $\Gamma \mid \Sigma \vdash \tau : T$ and $\Gamma \mid \Sigma \vdash \mu$ (for some Γ and Σ), there exists a τ' and μ' with $\tau \mid \mu \longrightarrow^* \tau' \mid \mu'$ and $\tau' \mid \mu' \not\rightarrow$? If so, prove this. If not, then: (1) write a well-typed function, in this language extended with natural numbers and booleans, that takes as its argument a natural number n and returns the sum of all numbers from 0 to n , and (2) write this same function in OCaml without using `let rec` or `fold` functions.

2 Exercise In Exercise 3.5.16 from TAPL (from Homework 1), we added the term `wrong` to the Arith language along with some evaluation rules. Chapter 14 of TAPL adds error handling to the simply-typed λ -calculus (see Figure 14-2). What is the difference between adding `wrong`, and adding error handling (besides the trivial difference that we're starting with different languages in each case)? Is there a difference?

We would like you to think about this carefully. One thing you may consider, for example, is if we added these features to solve the same problem with our languages. Keep your answer to about one paragraph (possibly two).

3 Exercise For each new extension to the simply-typed lambda calculus (STLC) with error (see Figure 14-1), we must add new rules to propagate the error. For example, in the language STLC + error + booleans, we need the rule

`if error then t2 else t3 → error`

in order to show the progress theorem. What new rules do we need for STLC + error + unit + references, i.e., the language of Figure 13-1 extended with error? (Reminder: Don't forget about the store!)

4 Exercise State the canonical forms, progress, and preservation lemmas for STLC + error handling (see Figure 14-2). Also state the type soundness theorem for this language.

5 Exercise Suppose we extend the language STLC + unit + references (Figure 13-1) with the following new syntax, typing rules, and evaluation rules:

$t ::= \dots$ terms:
`free t` deallocation

$$\frac{\Gamma \mid \Sigma \vdash t : \text{Ref } T}{\Gamma \mid \Sigma \vdash \text{free } t : \text{Unit}} \text{T-FREE}$$

$$\frac{l \in \text{dom}(\mu)}{\text{free } l \mid \mu \longrightarrow \text{unit} \mid (\mu \setminus \{l\})} \text{E-FREEREF} \qquad \frac{\tau \mid \mu \longrightarrow \tau' \mid \mu'}{\text{free } \tau \mid \mu \longrightarrow \text{free } \tau' \mid \mu'} \text{E-FREE}$$

where we use the notation $\mu \setminus \{l\}$ to indicate the store μ without any binding for the location l . Some questions about this language:

1. Which of these new evaluation rules are computation rules?
2. Which of these new evaluation rules are congruence rules?
3. Does progress (Theorem 13.5.7) hold for this language? If so, give the new case(s) in the proofs. If not, give a counterexample.
4. Does preservation (Theorem 13.5.3) hold for this language? If so, give the new case(s) in the proofs. If not, give a counterexample.
5. Based on your answers to the previous two questions, can you give a program \mathfrak{t} in this language with $\emptyset \mid \emptyset \vdash \mathfrak{t} : \mathsf{T}$ and $\mathfrak{t} \mid \emptyset \longrightarrow^* \mathfrak{t}' \mid \mu' \not\rightarrow$ such that \mathfrak{t}' is not a value? If not, why not? If so, give us such a program. To make your program easier to read, you may use syntactic sugar.

6 Debriefing

1. How many hours did each person in your group spend on this assignment, including time taken to read TAPL?
2. Would you rate it as easy, moderate, or difficult?
3. Did everyone in your study group participate?
4. How deeply do you feel you understand the material it covers (0%–100%)?

If you have any other comments, we would like to hear them; please send them to cis500@cis.upenn.edu.