CS 565 Sample Final

Your name: Ben Delaware

April 24, 2022

Problem 1. Give two examples of values in the lambda calculus extended with natural numbers.

5

$\lambda x.\ x+1$ $\lambda x.\ 2$

Problem 2. Give two examples of stuck expressions in the lambda calculus extended with natural numbers.

there's a space here!

$5\ 5$

$5 + (\lambda x. x)$

Problem 3. Give two examples of reducible expressions in the lambda calculus extended with natural numbers, and show the result of taking a single step using the call-by-value reduction strategy.

$(\lambda x.\ x+1)\ (3+5) \to (\lambda x. x+1)\ 8 \to 8+1$

$((\lambda x. x)\ (\lambda x. x+1))\ 5 \to (\lambda x. x+1)\ 5 \to 5+1 \to 6$

$(3+5) \to 8$

Problem 4. For an arbitrary postcondition and IMP program, is there *always* a precondition that would result in a valid Hoare triple? In other words, for a concrete Q and c is there always a way to fill in the first part of the following triple such that it is provable?

$$|-\ \{\ \}\ c\ \{Q\}$$

If your answer is "yes", is this assertion always <u>unique</u>? If your answer is "no", give an example of a Q and c for which there is no such completion.

Yes, it is the assertion Produced by wlp(c,Q).
Recall that we showed $\vdash \{wlp(c,Q)\}\ c\ \{Q\}$ is always provable as part of the proof of relative completeness.
This assertion is not syntactically unique: we can define many different, but equivalent assertions: $5=5 \wedge wlp(c,Q)$, for example. It is always semantically unique however: there is no weaker precondition that will make it true!

Problem 5. Provide typing contexts and annotations that make each of the following be well-typed simply typed lambda calculus with natural number (STLC+Nat) expressions, or state that no such typing context and annotations exist. Give the type for the entire term if it is well-typed.

- $\vdash (\lambda\, w : nat \to nat \to nat\ .\ \lambda\, z : nat\ .\ (w\ z\ z) + 1)\ (\lambda\, y : nat\ .\ \lambda\, x : nat\ .\ y + x)\ 2 : nat$

- $\vdash \lambda\, x : nat \to nat\ .\ \lambda\, w : nat\ .\ x\ (1 + w) : (nat \to nat) \to nat \to nat$

Problem 6. Give an example of a either a reduction or typing rule that, when added to STLC+Nat, would break progress. Provide a concrete counterexample that demonstrates the violation.

$$\frac{\Gamma \vdash t_1 : Nat \quad \Gamma \vdash t_2 : T}{\Gamma \vdash t_1\ t_2 : Nat}$$

$$\frac{\vdash 0 : Nat \quad \vdash 1 : Nat}{\vdash 0\ 1 : Nat}$$, but

$0\ 1$ is a stuck term.

Problem 7. Give an example of a either a reduction or typing rule that, when added to STLC+Nat, would break preservation. Provide a concrete counterexample that demonstrates the violation.

$$\frac{\Gamma \vdash t_1 : Nat \quad \Gamma \vdash t_2 : Nat \quad \Gamma \vdash t_3 : T}{\Gamma \vdash (t_1 + t_2)\ t_3 : Nat}$$

Note: $\overline{\vdash (0+1)\ 3 : Nat}$ and $(0+1)\ 3 \longrightarrow 1\ 3$, (space)

but $\nvdash 1\ 3$

Problem 8. Give the most general unifiers for the following sets of constraints, or state that no solution exists.

- {Z=Y→ Q, Q=Nat}

[Q↦Nat, Z↦Y→Nat]

- {W=Q→ Q, Q=Nat, W=U, U=Nat}

No Solution!

- {}(the empty set of constraints)

[ ] (the empty substitution)

Problem 9. Suppose we have types Q, U, and W with Q <: U and U <: W. Which of the following subtyping assertions are then true? Write true or false after each one.

- (U→ Q)→ U <: (Q→ U)→ W.

  false

  Need: U <: W and (Q → U) <: (U → Q), but U ≮: Q

- {q:Q, y:Q} <: {y:U, q:W}.

  false

- {f: U → U, y:U} <: {g: Q → W, y:W}. true

Problem 10. Give the definition of polymorphic function composition ∘ in System F. In Coq, the definition of this function is:

```
Definition comp {A B C} (f : A → B) (g : B → C) : A → C :=
  fun a ⇒ g (f a).
```

$\Lambda A. \Lambda B. \Lambda C. \lambda f{:}A \to B. \lambda g : B \to C . \lambda a. g(f a)$

Problem 11. Give an example of a System F term which has no analogue in the simply typed lambda calculus.

$\Lambda B. \lambda f : (\forall A. A \to A). f [\forall A. A \to A] f B : \forall B. (\forall A. A \to A). B \to B$