COMP4161 T3/2023
Advanced Topics in Software Verification

Assignment 1

This assignment starts on Friday 22nd September 2023 and is due on Friday 29th September 2023 23:59:59. We will accept plain text (.txt) files, PDF (.pdf) files, and Isabelle theory (.thy) files. You are allowed to make late submissions up to five days (120 hours) after the deadline, but at a cost: -5 marks per day.

The assignment is take-home. This does NOT mean you can work in groups. Each submission is personal. For more information, see the plagiarism policy: https://student.unsw.edu.au/plagiarism

Submit using `give` on a CSE machine:

```
give cs4161 a1 files ...
```

For example:

```
give cs4161 a1 a1.thy a1.pdf
```

1 λ-Calculus (16 marks)

(a) Simplify the term $(xy)(\lambda x.(\lambda y.(\lambda z.(z(xy)))))$ syntactically by applying the syntactic conventions and rules. Justify your answer. (2 marks)

(b) Restore the omitted parentheses in the term $x(\lambda xy.x(yz)(xy))(\lambda y.yz)$ (make sure you don't change the term structure). (2 marks)

(c) Find the normal form of $(\lambda f.\lambda x.f(f(fx)))(\lambda g.\lambda y.g(gy))$. Justify your answer by showing the reduction sequence. Each step in the reduction sequence should be a single β-reduction step. Underline the redex being reduced for each step. (6 marks)

(d) Recall the encoding of natural numbers in lambda calculus (Church Numerals) seen in the lecture:

$0 \equiv \lambda f\ x.\ x$
$1 \equiv \lambda f\ x.\ f\ x$
$2 \equiv \lambda f\ x.\ f\ (f\ x)$
$3 \equiv \lambda f\ x.\ f\ (f\ (f\ x))$ ...

Define `exp` where `exp m n` beta-reduces to the Church Numeral representing m^n. Provide a justification of your answer. (6 marks)

2 Types (20 marks)

(a) Provide the most general type for the term $\lambda ab.a(cb)b$. Show a type derivation tree to justify your answer. Each node of the tree should correspond to the application of a *single* typing rule, and be labeled with the typing rule used. Under which contexts is the term type correct? (5 marks)

(b) Find a closed lambda term that has the following type:

`('a ⇒ 'b) ⇒ 'a ⇒ ('a ⇒ 'b ⇒ 'c) ⇒ 'c`

(You don't need to provide a type derivation, just the term). (4 marks)

(c) Explain why $\lambda x.xx$ is not typable. (3 marks)

(d) Find the normal form of $(\lambda xy.y)(\lambda z.zz)$ and give it a type. (3 marks)

(e) Is $(\lambda xy.y)(\lambda z.zz)$ typable? Compare this situation with the Subject Reduction that you learned in the lecture. (5 marks)

3 Propositional Logic (29 marks)

Prove each of the following statements, using only the proof methods: `rule`, `erule`, `assumption`, `cases`, `frule`, `drule`, `rule_tac`, `erule_tac`, `frule_tac`, `drule_tac`, `rename_tac`, and `case_tac`; and using only the proof rules: `impI`, `impE`, `conjI`, `conjE`, `disjI1`, `disjI2`, `disjE`, `notI`, `notE`, `iffI`, `iffE`, `iffD1`, `iffD2`, `ccontr`, `classical`, `FalseE`, `TrueI`, `conjunct1`, `conjunct2`, and `mp`. You do not need to use all of these methods and rules.

(a) `A ⟶ ¬ ¬ A` (3 marks)

(b) `¬ ¬ ¬ A ⟶ ¬ A` (3 marks)

(c) `¬ ¬ A ⟶ A` (4 marks)

(d) `¬ (A ∧ B) ⟶ ¬ A ∨ ¬ B` (4 marks)

(e) `(A ⟶ B) ⟶ ¬ A ∨ B` (4 marks)

(f) `(¬ A ∧ ¬ B) = (¬ (A ∨ B))` (6 marks)

(g) `(A ⟶ B) ⟶ ((B ⟶ C) ⟶ A) ⟶ B` (5 marks)

4 Higher-Order Logic (35 marks)

Prove each of the following statements, using only the proof methods and proof rules stated in the previous question, plus any of the following proof rules: `allI`, `allE`, `exI`, and `exE`. You do not need to use all of these methods and rules. You may use rules proved in earlier parts of the question when proving later parts.

(a) `(∃x. P x ⟶ Q) ⟶ (∀x. P x) ⟶ Q` (4 marks)

(b) `((∃x. P x) ⟶ Q) = (∀x. P x ⟶ Q)` (6 marks)

(c) `(∀x. P x) = (∄x. ¬ P x)` (6 marks)

(d) `(∀x. P x ∧ Q x) ⟶ (∀x. P x) ∧ (∀x. Q x)` (6 marks)

(e) `(∃x. P x ∨ Q x) ⟶ (∃x. P x) ∨ (∃x. Q x)` (6 marks)

(f) `(∀x y. A y ∨ B x) ⟶ (∀x. B x) ∨ (∀y. A y)` (7 marks)