COMP3610/6361 Principles of Programming Languages

Assignment 1

ver 1.0

Submission Guidelines

- Due time: Aug 31, 2023, 11am (Canberra Time)
- Submit a pdf via Wattle.
- Scans of hand-written text are fine, as long as they are readable and neat.
- Please read and sign the declaration on the last page and attach a copy to your submission.
- **No late submission, deadline is strict**

Exercise 1 (Type Preservation) (10 Marks)

Does *Type Preservation* hold for the variant language with rules (assign1') and (seq1') from page 47 of the slides (see also below) instead of (assign1) and (seq1)?

$$\text{(assign1')} \qquad \langle l := n, s\rangle \rightarrow \langle n, s + \{l \mapsto n\}\rangle$$

$$\text{(SEQ1')}\ \langle v; E_2, s\rangle \rightarrow \langle E_2, s\rangle$$

If not, give an example, and explain which changes to the typing rules would be needed to get the property back; if yes, sketch a proof.

Exercise 2 (Structural Induction) (20 Marks)

A BTree is defined by the following grammar.

$$T ::= \text{Id} \mid \text{One}\ T \mid \text{Two}\ T\ T$$

That means that leaves are labelled 'Id', and inner nodes can have One or Two children.

Question 1 Define a function *leaves* that determines the number of leaves for a given tree.

Question 2 Define a function *two_succ* that determines the number of nodes with two children.

Question 3 Derive an induction principle for BTree

Question 4 Prove that any tree B with n leaves has exactly $n-1$ nodes with two children:

$$\mathit{two\_succ}\ B = (\mathit{leaves}\ B) - 1$$

Exercise 3 (Functions) (20 Marks)

Question 5 Calculate the free variables of the following expressions:

1. $x + (\mathbf{fn}\ y : \mathrm{int} \Rightarrow z)$
2. $(\mathbf{fn}\ y : \mathrm{int} \Rightarrow (\mathbf{fn}\ y : \mathrm{int} \Rightarrow (\mathbf{fn}\ y : \mathrm{int} \Rightarrow y)))$
3. $\mathbf{while}\ !l_0 \geq y\ \mathbf{do}\ l_0 := x$

Draw also their abstract syntax trees (up to alpha equivalence).

Question 6 Perform the following substitutions:

1. $\{y\ z/x\}\,(\mathbf{fn}\ x : \mathrm{int} \Rightarrow y\ x)$
2. $\{z\ x/x\}\,(\mathbf{fn}\ y : \mathrm{int} \Rightarrow y\ x)$
3. $\{z\ x/x\}\,(\mathbf{fn}\ z : \mathrm{int} \Rightarrow (\mathbf{fn}\ x : \mathrm{int} \Rightarrow y\ x)\ x\ z)$

Exercise 4 (Nested substitution) (20 Marks)

Question 7 For the language variant featuring while, if-statement and functions (no recursion), show the following statement.

$$\{E_3/y\}\,\{E_2/x\}\,E_1 \quad = \quad \{(\{E_3/y\}\,E_2)/x\}\,\{E_3/y\}\,E_1$$

where x and y are variable with $x \neq y$, and E_1, E_2 and E_3 expressions with $x \notin \mathrm{fv}(E_3)$. Clearly state your proof strategy and mark the places where the assumptions are used.

Exercise 5 (Exception Handling) (10 Marks)

When using error handling in its simplest form (see lecture), the progress property does not hold.

Question 8 Explain why the property is broken and give an example.

Question 9 Fix the progress theorem in a way that it still captures the essence of progress. The theorem should hold for languages with error handling. Other properties such as type preservation and type safety should stay valid.

Exercise 6 (Subtyping) (20 Marks)

Question 10 For each of the two bogus T ref subtype rules

$$\frac{T <: T'}{T\ \mathrm{ref} <: T'\,\mathrm{ref}} \qquad\qquad \frac{T' <: T}{T\ \mathrm{ref} <: T'\,\mathrm{ref}}$$

give an example program that is typable with that rule but gets stuck at runtime.

Question 11 We have introduced the following rule for subtyping products

$$\text{(s-pair)} \quad \frac{T_1 <: T_1' \qquad T_2 <: T_2'}{T_1 * T_2 <: T_1' * T_2'}$$

Would it be a good idea to add a subtyping rule such as $T_1 * T_2 <: T_1$? Justify your answer.

Academic Integrity

I declare that this work upholds the principles of academic integrity, as defined in the University Academic Misconduct Rule; is entirely my own work, with only the exceptions listed; is produced for the purposes of this assessment task and has not been submitted for assessment in any other context, except where authorised in writing by the course convener; gives appropriate acknowledgement of the ideas, scholarship and intellectual property of others insofar as these have been used; in no part involves copying, cheating, collusion, fabrication, plagiarism or recycling.

____________________ Date

____________________ Signature