

Assignment #3

Name: _____ ID: _____

This assignment has 4 questions, for a total of **25** marks.

Question 1: **SOS for STLC** 5 marks

Write the reduction rules for structural operational semantics, small step, call by name for STLC.

Question 2: **Progress cases for pairs** 5 marks
Write the proof for the progress theorem for the following cases related to pairs: $t \equiv \langle t_1, t_2 \rangle$ [3], $t \equiv t_1.1$ [2].

Question 3: **Progress cases for sums** 4 marks
Write the proof for the progress theorem for the following cases related to sums: $t \equiv \text{inl } t_1$ [2], $t \equiv \text{case } t_1 \text{ of } \text{inl } x_1 \mapsto t_2 \mid \text{inr } x_2 \mapsto t_3$ [2].

Question 4: **Named functions** 11 marks

Extend STLC to have named function, call this language STLCN. A program is no longer a term, but a collection of named functions. A named function defines a function name, a parameter of a certain type and the function body. A function body is a term. Terms now must include new constructs to call other functions.

Define the new syntactic category of programs P , of named functions F and of terms [1]. Define the COS judgements for STLCN [1]. Define the primitive reduction rules for STLCN as well as evaluation contexts [2]. Define the typing judgements for STLCN, starting from how to determine when a program is well typed [3]. Define the typing rules for STLCN [4].

Some primitive reductions and typing rules from STLCN will be similar to those of STLC. To avoid too much duplicates, name all the STLC rules that have analogous ones and show how to convert two of them to the new forms only.

STLCN should still be safe, i.e., it should not get stuck trying to call a function with a parameter of the wrong type, or calling a function that does not exist. However, STLCN is not normalising: there may be functions that mutually call each other and thus diverge.