Semantics of Programming Languages

The following grammar specifies the types and expressions of a simple functional programming language.

Types: \( T ::= \text{int} | T \rightarrow T' \)

Expressions: \( e ::= n | x | e + e' | \text{fn}(x : T) \Rightarrow e | ee' \)

where \( n \) ranges over all integers, and \( x \) ranges over variables.

(a) Give a reasonable semantics for this language, by specifying a type system and a reduction relation. Use the call-by-name evaluation order. \([9 \text{ marks}]\)

(b) Write down all the reduction steps of the following expression. You do not need to give their derivations.

\[
(fn(x : \text{int}) \Rightarrow (fn(x : \text{int}) \Rightarrow x + x)) (1 + 2) (3 + 4)
\]

\([3 \text{ marks}]\)

(c) Prove the following property of substitution. [Hint: Use rule induction for \( \Gamma, x : T \vdash e' : T' \).]

\[
\text{if } \Gamma \vdash e : T \text{ and } \Gamma, x : T \vdash e' : T' \text{ with } x \not\in \text{dom}(\Gamma) \text{ then } \Gamma \vdash \{e/x\}e' : T'
\]

\([8 \text{ marks}]\)