14 Types (AMP)

(*a*) Give the Mini-ML typing rules for variables, boolean values and conditional expressions, function abstraction and application, and for expressions of the form `let` $x = M_1$ `in` M_2. [6 marks]

(*b*) Midi-ML is obtained from Mini-ML by adding a unit type *unit*, reference types τ *ref* and associated expressions for the unit value (), reference creation `ref` M, dereferencing $!M$ and assignment $M_1 := M_2$. Give the Midi-ML typing rules for these forms of expression. [4 marks]

(*c*) What is meant by the *type soundness* property of a programming language and its type system? [1 mark]

(*d*) Explain why combining the typing rules in part (*a*) with those in part (*b*) leads to an unsound type system and how Midi-ML modifies the typing rule for let-expressions in order to ensure the type soundness property. Illustrate your answer using the expression

$$\texttt{let}\, r = \texttt{ref}\, \lambda x(x)\, \texttt{in}\, (\lambda y(!r()))(r := \lambda x(\texttt{if}\, x\, \texttt{then}\, \texttt{false}\, \texttt{else}\, \texttt{true})).$$

(You need not give a formal definition of the operational semantics of Midi-ML.) [9 marks]