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 \texttt{let } x = M_1 \texttt{ in } M_2. \quad [6 \text{ marks}]

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

(c) What is meant by the type soundness property of a programming language and its type system? \quad [1 \text{ 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

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

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