Due: Friday, 3 May 2002, 12:00

Preliminary remark: All the following should be solved in Agda or Alfa. All solutions should be both type checked and a termination check should be carried out.

Question 1.

• Introduce the set \( \mathbb{N} \) of natural numbers.

• Introduce \texttt{zero}, the successor operation and elements \texttt{one}, \texttt{two}, \texttt{three} of \( \mathbb{N} \).

• Introduce addition and multiplication.

• Introduce, depending on \texttt{A:: Set} and \texttt{n::N} the set \texttt{Vec A n} of vectors of length \( n \) with elements in \( A \).

• Define the sum of two elements of \texttt{Vec N n}. \hspace{1cm} \textbf{[15 marks]}

Question 2.

• Introduce the equality on natural numbers.

• Prove that this equality is reflexive, ie.
  \begin{verbatim}
  define \texttt{refl} :: (n::N) -> \texttt{Eqnat n n}.
  \end{verbatim}
  Hint: Use induction (essentially case distinction).

• Prove that it is symmetric, ie. define

  \begin{verbatim}
  sym (n::N)
  (m::N)
  (p::\texttt{Eqnat n m})
  :: \texttt{Eqnat m n}
  \end{verbatim}

• Prove that this equality is transitive, ie.
  \begin{verbatim}
  define \texttt{trans} (n::N)
  (m::N)
  (k::N)
  (p::\texttt{Eqnat n m})
  (q::\texttt{Eqnat m k})
  ::\texttt{Eqnat n k}
  \end{verbatim}
  \begin{itemize}
  \item Hint: Use induction (essentially case distinction).
  \end{itemize}
If \( n \neq m \) or \( m \neq k \) you can use the empty case distinction on \( p \).
- You might need to prove auxiliary properties like \( 0 + n = n \).

- Prove that + is commutative, ie. \( n + m = m + n \).

HINTS:
- You might need some auxiliary lemma like \( (S \ n) + m = S(n + m) \).
- You might need to use reflexivity or transitivity.

[30 MARKS]

Question 3.
- Introduce the \( n \times m \) Matrices with elements in \( N \).
  - A matrix is a vector of columns, each column being a vector of elements of \( N \).
- Define the functions which you need later: the projection of a matrix with \( S \ n \) rows to its first row and to the other \( n \) rows.
- Now define the product of two matrices:
  - First define the result of multiplying one row with one column, ie.
    \[
    (a_1, \ldots, a_n) \cdot \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = a_1 \cdot b_1 + \cdots + a_n \cdot b_n
    \]
  - Now define the result of multiplying an \( n \times m \) matrix with one column:
    \[
    \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}
    \]
    * It’s probably easiest to work by recursion on \( n \).
    * You will probably need to make use of the projections defined above.
- Now define the product of matrices.

[30 MARKS]

Question 4. The goal of this question is to verify a half-bit adder.
- A binary circuit component is a function which has as input two Boolean values and returns one Boolean value.
  A unary circuit component is a function which has as input one Boolean value and returns one Boolean value.
  Introduce the set of binary and the set of unary components.
• Introduce the and-gate and the or-gate as binary components and the not gate as a unary gate.

• Introduce the identity function (ie. $\lambda x.x$) as a unary component.

• Introduce a function which takes three binary circuit components and returns a binary circuit component which is the result of composing the three components according to the following diagram:

• Introduce the function which takes two unary and one binary circuit component and returns a binary circuit component, which is the result of composing the three components according to the following diagram:

• Consider a half bit adder:

  - The output “sum” of the half-bit adder depending on the inputs is a binary circuit component, which one obtains by:
    * starting with an and-gate (the one to the right),
    * composing it with the unary circuit components identity and not,
    * then composing it again with an and- and an or-gate.
  - The output “carry” is obtained by taking an and gate.

Construct sum and carry as the result of composing the just mentioned circuit components.

• Now define a second version of sum and carry as a binary circuit components, by defining the corresponding functions $\text{Bool} \rightarrow \text{Bool} \rightarrow \text{Bool}$ directly using case distinction.

• Introduce an equality on $\text{Bool}$. 
• Now show that the “sum” and “carry” defined directly coincide with the versions constructed from logical gates, i.e., show that for all Boolean inputs the outputs of both versions coincide.

[25 marks]