CS 6V81-002: Quiz 6 Solutions

February 13, 2008

1. Dependent types can be thought of as... (circle all that apply)

   (a) functions from type variables to terms
   (b) functions from terms to types
   (c) a generalization of arrow types
   (d) a generalization of sum types

2. Consider two functions, `first1` and `first2`, where `first1` has type $\mathit{Vector} \rightarrow \mathit{data}$ and `first2` has type $\Pi n\!:\!\mathit{Nat}.\,\mathit{Vector}(n+1) \rightarrow \mathit{data}$. A difference between `first1` and `first2` is... (circle all that apply)

   (a) `first2` does not type-check when applied to an empty vector, but `first1` does
   (b) `first2` can only be passed a unique pointer to the vector, whereas `first1` accepts pointer aliases
   (c) `first2`'s type asserts that it returns the 1st element of the vector, whereas `first1`'s type only asserts that it returns some value of type *data*
   (d) `first2` expects more arguments than `first1`

3. The addition of dependent types into the typing system of DTAL improves upon TAL by... (circle all that apply)

   (a) supporting array bounds check elimination optimizations
   (b) reducing the TCB by reducing the number of typing rules
   (c) reducing the TCB by simplifying the proof of soundness
   (d) reducing the TCB by reducing the size of typing annotations

4. In addition to a type-checker, consumers of DTAL code must also implement... (circle one)

   (a) a "logical inference engine" (based on bayesian logic)
   (b) a "constraint solver" (based on linear integer programming)
   (c) a "type inference engine" (based on the Hindley-Milner algorithm)
   (d) a "just-in-time (JIT) compiler" (based on Xanadu)

5. A function of type $\Pi n\!:\!\mathit{Nat}\,.\,(\alpha)\mathit{list}(n) \rightarrow (\alpha)\mathit{list}(n)$ returns a list that is the same length as the input list. Write a dependent type for a function that that accepts two lists as input and returns the concatenation of the two lists.

$$\Pi m\!:\!\mathit{Nat}\,.\,\Pi n\!:\!\mathit{Nat}\,.\,(\alpha)\mathit{list}(m) \rightarrow (\alpha)\mathit{list}(n) \rightarrow (\alpha)\mathit{list}(m+n)$$