The two programs below compute the inner (scalar) product of two vectors. The first program is written in FP; the second one is written in Haskell.

1. \{ip !+ @ &* @ trans\}

2. \text{ip2 = foldr (+) 0 . map mult2a . zip2, where mult2a and zip2 are uncurried versions of (*) and zip, respectively.}

What are the symbols for composition in FP and Haskell, respectively?
\textbf{Answer:} @ and .

What are the symbols for apply-to-all in FP and Haskell, respectively?
\textbf{Answer:} & and \textsf{map}

Why does FP’s insert (!) have only one argument, while the corresponding higher-order function in Haskell, \textsf{foldr}, has two arguments?
\textbf{Answer:} Because !f is not defined for empty sequences. \textsf{foldr} instead works for lists of any length. In particular, \textsf{foldr f v \{\}} = v. Note that \textsf{foldr f v \{x\}} = f x v, while !f: \textsf{\langle x\rangle} = x, so v should be the identity element.