Foundations of Functional Programming

(a) Write a pure lambda-expression that will act as a fixed-point operator $Y$ such that the identity $Yf = f(Yf)$ will hold. [6 marks]

(b) Write pure lambda-expressions that define functions $P$, $A$ and $D$ such that $A(Px y) = x$ and $D(Px y) = y$. Observe that $P$ can be thought of as creating a 2-tuple and $A$ and $D$ then act as selectors that can retrieve the two components. [7 marks]

(c) Using the two above lambda-expressions it is possible to express mutual recursion between two functions, say $f$ and $g$. This can be done by using $Y$ to help find the value of $(Pfg)$ the tuple whose elements are $f$ and $g$. Using the artificial and rather silly example [the example will never terminate since it has no stopping condition!]

\[
\begin{align*}
    f\ x &= g (f (g\ x)) \\
    \text{AND} \ g\ x &= g (f\ x)
\end{align*}
\]

show how to construct a pure lambda expression that would evaluate

\[
(f\ g)
\]

[7 marks]