2005 Paper 6 Question 10

Foundations of Functional Programming

(*a*) Give reduction rules associated with the standard combinators `S`, `K` and `I`. [4 marks]

(*b*) Explain as simple a procedure as possible that converts lambda expressions into terms that use only these combinators. At this stage you are not expected to be concerned with efficiency, but you should make it clear why your combinator forms relate to the lambda expressions that you start with. [5 marks]

(*c*) Illustrate the scheme you have given above by applying it to the lambda expression $\lambda x.\lambda y.((y\ x)\ y)$ and explain why in practical reduction to combinators at least additional symbols `B` and `C` are normally introduced. [5 marks]

(*d*) Suppose that you have primitive functions available that can represent integers, booleans, arithmetic and an "if ... then" operation (and so on, as necessary). Show how to convert the following ML-like code first into raw lambda calculus and then into combinator form:

```
fun f n = if (iszero n) 1
             (double (f (subtract1from n)));
```

Note that the code has been presented in a way intended to show the functions used to perform arithmetic. You may assume the availability of a `Y` operator if that helps. [6 marks]