PRIFYSGOL CYMRU; UNIVERSITY OF WALES

DEGREE EXAMINATIONS MAY/JUNE 2003

SWANSEA

Computer Science

CS 321 Functional Programming II

Attempt 2 questions out of 3

Time allowed: 2 hours

Students are permitted to use the dictionaries provided by the University

Students are NOT permitted to use calculators

CS_321
FUNCTIONAL PROGRAMMING 2

(Attempt 2 questions out of 3)

Question 1

a. A functional language can be implemented by translating it into an extended version of λ-calculus. Give a typical translation of the following pieces of Gofer code into λ-expressions.

i)

```
sumsquares x y = (square x) + (square y)
                 where square z = z * z
```

ii)

```
sum n = if n == 1 then 1 else n + sum (n-1)
```

[5 marks]

b. Assume that the following constants have the following types:-

```
+:     int -> int -> int
-:     int -> int -> int
FIX:   ((a->b)->(a->b))->(a->b)
COND:  bool -> c -> c -> c
EQ:    d -> d -> bool
1:     int
```

Carefully derive the type of `sum` by assigning appropriate types to each component of the λ-translation you gave in part **(a)**. Indicate how, and where, type equivalences arise. Explain how type checking can be considered an example of term unification.

[6 marks]

c. Suggest some suitable Haskell datatypes to represent the translated form of code in **(a)**, and type expressions that could be used in a program to carry out type checking. How could the definitions for type expressions be extended to represent type schemes and why might this be necessary?

[8 marks]

d. In part **(b)** we assumed that the arithmetic operators + and – operated on integers. Explain how the Class system of Haskell allows us to define overloaded functions which can be defined to operate on any numeric type.

[6 marks]

Question 2

a. It is desired to generate a stream of integer values in ascending order such that each entry in the stream is a product of powers of 2, 3 and 5 (*ie* the stream of values $2^m3^n5^p$). Show how functions defined on streams can be used to generate this sequence of values. **[4 marks]**

b. What is the general type of a program which can operate on files, and on standard input and output in Haskell or Gofer? What is the assumed type of the operating system? Give brief details of the sorts of values that can be used as elements of the input and output types. **[5 marks]**

c. This type involves lists and the usual approach to defining functions on lists is to use pattern matching. Why might this be inappropriate in the context of an interactive program, and how can irrefutable patterns be used? **[4 marks]**

d. An alternative approach to writing interactive Haskell programs is to use the concept of a monad. Explain how this approach works. **[5 marks]**

e. Arrays can be modelled by lists, but this can lead to large amounts of data being replicated. Explain why this is the case and how the monad approach can reduce the problem. **[7 marks]**

Question 3

For this question you should answer ONE of the TWO options (i) OR (ii) in part (a) AND ALL of part (b).

a. EITHER

i)

(1) Translate the following expression into a suitable form for evaluation on an SECD machine

```
times  2  3   where  times a b = a * b
```

[3 marks]

(2) State what the four stacks of the SECD machine are used for and outline how the translation of the expression in **(1)** would proceed. You do not need to give a complete evaluation, but you should outline the steps involved and clearly indicate how each stack is used in the evaluation and how it is determined that the evaluation has finished. **[7 marks]**

(3) Is the evaluation process you carried out in **(2)** strict or lazy? Discuss how you might adapt the approach to make it strict if what lazy or vice versa, or why this is not possible. **[3 marks]**

OR

ii)

(1) The following is the result of translating an expression into SKI combinator form.

```
S ( S ( K + ) I ) (K 2) 3
```

Show how the above expression can be represented in graphical form and indicate how it would be evaluated. A complete evaluation is not required, but you should outline all the steps necessary in the evaluation process.

[6 marks]

(2) Expressions of the form `S E F` can often be simplified when either `E` and/or `F` are of the form `K E` or `I` when two additional combinators (given below) may be introduced to reduce the complexity of such translations. Briefly explain what optimisations are possible and optimise the expression given in **(1)**.

```
B = λfxy.f(xy)
C = λfxy.fyx
```

[7 marks]

b. ***Answer ALL of this part***

i) Consider the following function and suggest how it might be represented using a data flow graph.

```
sum n = if n == 1 then 1 else n + sum (n-1)
```

[4 marks]

ii) Outline how the graph given in **(i)** could be translated into a form suitable for evaluation on a parallel computer. Indicate a suitable abstract architecture for the computer and outline a execution cycle for both data-driven and demand-driven evaluation.

[8 marks]