Exame de Implementação de Linguagens

Departamento de Ciência de Computadores
Faculdade de Ciências – Universidade do Porto
14 de Julho de 2015
Duração: 2 horas

Parte I

(1) Let f be the following expression (in Haskell syntax):

```
let f = (let y = 2 in \x -> x + y) in f 3
```

Based on the compiler presented in appendix, compile this expression to SECD code and then present the step-by-step execution of the resulting SECD code (for each step present the execution stacks resulting from running the SECD machine).

(2) Modify the abstract syntax of the term language presented in appendix to add boolean expressions (*And*, *Or*). For these new boolean expressions define an operational semantics. Finally describe how would you compile the boolean expressions to the SECD machine.

(3) Compile the expression:

```
(\x -> (\y -> y * x) x)3
```

to a set of super-combinators.

Parte II

(4) Considere o programa Prolog que se segue:

```
p(end(X),[]).
p(next(X),[H|T]) :- do(p(next(X),H),Y), p(Y,T).
```

(a) Escreva o código compilado do programa no contexto da máquina $\mathcal{M}_3$ da WAM (note que o termo `[]` é uma constante).

(b) Considere as seguintes otimizações à WAM: (i) tratamento de constantes; (ii) tratamento de variáveis anónimas; (iii) melhor alocação de registos; (iv) *last call optimization*. Reescreva o código anterior de forma a tirar partido dessas otimizações (indique de forma clara onde cada otimização se encontra no novo código).

(5) Considere o programa Prolog que se segue em que *p/2* é um predicado tabelado:

```
:- table p/2.                          e(x,y).
p(A,B) :- p(A,C), e(C,B).              e(y,x).
p(A,B) :- e(A,B).                      e(y,z).

output(A,B) :- write(A), write('-'), write(B), nl.
```

Considere ainda a seguinte consulta:

```
?- p(X,Y), output(X,Y), fail.
```

(a) Indique quais as linhas de *output* (independente da ordem) obtidas pela execução da consulta.

(b) Justifique qual seria a primeira linha de *output* obtida pela execução da consulta caso *p/2* não fosse declarado como um predicado tabelado.

(c) Na *tabulação por suspensão da computação*, a execução é vista como uma sequência de sub-computações que podem ser suspensas e mais tarde recuperadas até se atingir um ponto-fixo. Descreva de forma sucinta que mecanismos podem ser utilizados para permitir suspender e recuperar o estado da computação e que condições são necessárias para garantir que se atingiu um ponto-fixo.