

Exame de Implementação de Linguagens

Departamento de Ciência de Computadores
Faculdade de Ciências – Universidade do Porto
23 de Junho de 2015
Duração: 2 horas

Parte I

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

```
let func = \x -> 2*x + 5 in func 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) For the following Haskell program:

```
sum :: [Integer] -> [Integer]  
sum [] = 0  
sum (x:xs) = x + sum xs
```

Consider the term language defined as argument of the compile function presented in appendix, extended with pairs and variants. Translate this Haskell program to that term language (hint: use pairs and variants to code lists).

(3) Consider the following Haskell program:

```
sqr = \x -> x*x  
main = sqr (sqr 3)
```

Write the main function as a graph (in reduction graph notation) and write the step-by-step execution of its graph reduction process (presenting all the intermediate graphs following each reduction step).

Parte II

- (4) Considere o programa Prolog que se segue:

```
reverse(empty, empty).  
reverse(list(H,T),L) :- reverse(T,L1), append(L1, list(H,empty), L).
```

- (a) Escreva o código compilado do programa no contexto da máquina \mathcal{M}_3 da WAM.
- (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 *ks/3* é um predicado tabelado com *mode-directed tabling*:

```
:- table ks(index,index,max).  
ks(0,0,0).  
ks(I,K,V):- I > 0, I1 is I-1, ks(I1,K,V).  
ks(I,K,V):- I > 0, item(I,F), K1 is K-F, K1 >= 0,  
                I1 is I-1, ks(I1,K1,V1), V is V1+1.
```

item(5,10).
item(4,20).
item(3,30).
item(2,40).
item(1,50).

Considere ainda a seguinte consulta:

```
?- ks(5,100,V), fail.
```

- (a) Indique quais as chamadas que seriam tabeladas pela execução da consulta e quais as respectivas soluções para cada chamada.
- (b) Considerando ainda a mesma consulta, descreva de forma sucinta o que aconteceria caso *ks/3* fosse apenas declarado como um predicado tabelado sem *mode-directed tabling*. Que implicações isso poderia ter num problema de maior dimensão (i.e., com um maior número de itens).
- (c) A técnica da tabulação pode ser vista como o resultado da composição de uma operação de *Generate()* com uma operação de *Aggregate()*. Justifique esta afirmação e, nesse contexto, explique como pode ser vista a extensão de *mode-directed tabling*.