

Exame de Implementação de Linguagens

Departamento de Ciência de Computadores
Faculdade de Ciências – Universidade do Porto
11 de Junho de 2013
Duração: 2 horas

Parte I

- (1) No contexto da máquina \mathcal{M}_3 da WAM, considere o código WAM que se segue:

```
L2: trust_me                                p/2: try_me_else L2
    allocate 4                               get_structure end/0, A1
    get_structure task/2, A1                 get_structure []/0, A2
    unify_variable X4                       proceed
    unify_variable Y1
    get_variable Y2, A2
    put_value X4, A1
    put_structure task/1, A2
    set_variable X5
    put_variable Y3, A3
    call do/3
    put_value Y1, A1
    put_variable Y4, A2
    call p/2
    put_value Y3, A1
    put_value Y4, A2
    put_value Y2, A3
    call append/3
    deallocate
```

- (a) Escreva o programa Prolog correspondente ao código WAM apresentado.
- (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*. Diga sucintamente no que cada uma consiste e reescreva o código WAM anterior de forma a tirar partido dessas otimizações.

- (2) Considere os seguintes predicados Prolog:

```
p(X,Z) :- p(X,Y), e(Y,Z).                e(a,b).
p(X,Z) :- e(X,Z).                        e(b,a).

p(X,Z,D) :- p(X,Y,D1), e(Y,Z,D2), D is D1+D2.  e(a,b,1).
p(X,Z,D) :- e(X,Z,D).                      e(b,a,2).
```

- (a) Tendo por base as *queries*

Q1) ?- p(X,Z).
Q2) ?- p(X,Z,D).

identifique, para cada *query*, quais os problemas que a resolução SLD não consegue tratar devidamente e indique que tipo de técnicas poderiam ser utilizadas para os solucionar.

- (b) Descreva de forma sucinta no que consiste o modelo básico de execução dessas técnicas e indique quais as principais extensões de suporte à sua implementação numa máquina WAM.

Parte II

(3) Considere a seguinte função em Haskell que calcula o máximo divisor comum usando o algoritmo de Euclides:

```
mdc :: Int -> Int -> Int
mdc a b = if b==0 then a else mdc b (a`mod`b)
```

- (a) Traduza esta definição para a linguagem FUN apresentada nas aulas. Deve assumir que existe um operador aritmético extra % para calcular o resto da divisão.
- (b) Considere agora uma nova instrução MOD para a máquina SECD que calcula o resto da divisão. Escreva a regra de transição da semântica operacional para esta instrução e o esquema de compilação do operador %.
- (c) Usando o esquema de compilação da FUN para a SECD e a extensão da alínea anterior traduza a definição da função mdc para código da máquina SECD.

(4) Considere a seguinte definição da função filter em Haskell:

```
filter :: (a -> Bool) -> [a] -> [a]
filter f [] = []
filter f (x:xs) = if f x then x : filter f xs
                  else filter f xs
```

Traduza esta definição para a linguagem da máquina STG. Justifique a sua resposta.