

Exame de Implementação de Linguagens

Departamento de Ciência de Computadores
Faculdade de Ciências – Universidade do Porto
1 de Julho de 2013
Duração: 2 horas

Parte I

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

```
p(done, []).  
p(do(A,B),C) :- task(A,do(D),E), p(B,F), join(E,F,C).
```

- (a) No contexto da máquina \mathcal{M}_3 da WAM, escreva o código compilado que correspondente ao programa 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 anterior de forma a tirar partido dessas otimizações (indique de forma clara onde cada otimização se encontra no novo código).

- (2) Considere os seguintes predicados Prolog:

```
p1(A,B) :- e(A,B).                p2(A,B) :- e(A,B).                e(a,b).  
p1(A,B) :- e(A,X), p1(X,B).       p2(A,B) :- p2(A,X), e(X,B).       e(b,c).  
  
p3(A,B,D) :- e(A,X,D1), p3(X,B,D2), D is D1+D2.    e(a,b,1).  
p3(A,B,D) :- e(A,B,D).           e(b,a,2).  
  
output([A]) :- !, write(A), nl.  
output([H|T]) :- write(H), write(' '), output(T).
```

- (a) Tendo por base as *queries*

```
Q1) ?- p1(X,Y), output([X,Y]), fail.  
Q2) ?- p2(X,X), output([X,X]), fail.  
Q3) ?- p3(X,Y,Z), output([X,Y,Z]), fail.  
Q4) ?- p3(X,X,Z), output([X,X,Z]), fail.
```

identifique, para cada *query*, quais os problemas (se algum) 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, para cada *query*, qual o output obtido pela execução segundo esse modelo de execução.

Parte II

(3) Considere a seguinte função em Haskell que calcula o n -ésimo número de Fibonacci por recursão:

```
nfib :: Int -> Int
nfib n = if n>=2 then nfib (n-1) + nfib (n-2) else n
```

- (a) A linguagem FUN apresentada nas aulas tem apenas uma expressão condicional “ifzero e ” que testa $e = 0$. Pretende-se estender a linguagem com expressões condicionais da forma “if $e_1 \geq e_2$ then ... else ...”.

Sugestão: acrescente um operador \geq à linguagem cujo resultado será zero ou diferente de zero conforme o resultado comparação dos seus argumentos. Defina a semântica operacional deste operador e extenda o esquema de compilação da FUN para código da máquina SECD.

- (b) Traduza esta definição para a linguagem FUN apresentada nas aulas e o operador \geq sugerido na alínea anterior.
- (c) Usando o esquema de compilação da FUN para a SECD e a extensão da alínea anterior, traduza a definição de `nfib` para código da máquina SECD.

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

```
foldr :: (a -> b -> b) -> b -> [a] -> b
foldr f z [] = z
foldr f z (x:xs) = f x (foldr f z xs)
```

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