

Linguagens de Programação

Exceções

1. [Mitchell 8.1] Considere as seguintes funções escritas em ML:

```
exception Excpt of int;  
fun twice(f,x)=f(f(x)) handle Excpt(x) => x;  
fun pred(x)=if x=0 then raise Excpt(x) else x-1;  
fun dumb(x)=raise Excpt(x);  
fun smart(x)=1+pred(x) handle Excpt(x)=>1;
```

Qual o resultado da avaliação de cada uma das seguintes expressões. Descreva que exceções são levantadas e onde.

- (a) `twice(pred,1)`
- (b) `twice(dumb,1)`
- (c) `twice(smart,0)`

2. [Mitchell 8.4] A seguinte função ML utiliza a exceção `Odd` (a expressão $\sim n$ é $-n$, o valor negativo do inteiro n em ML)

```
fun f(0)=1  
| f(1) = raise Odd  
| f(3) = f(3-2)  
| f(i) = (f(i-2) handle Odd => ~i)
```

Quando `f(11)` é executado, são realizados os seguintes passos:

```
call f(11)  
call f(9)  
call f(7)  
...
```

Assuma que se `f` chama `g` e `g` levanta uma exceção que `f` não trata, então registo de activação de `f` é retirado do *stack* sem o controlo de retorno para a função `f` e escreva os passos restantes que serão executados. Inclua apenas os seguintes tipos de passos:

- chamada de função (com argumento)
- retorno de função (com resultado)
- levantamento de excepção
- *pop* do registo de activação do *stack* da função sem o controlo de retorno para a função
- tratamento de excepção