

# Linguagens de Programação

## Cálculo Lambda

1. Considere o termo- $\lambda$   $(\lambda x. \lambda y. x \ y)(\lambda x. x \ y)$

- (a) Identifique as diferentes variáveis que nele ocorrem.
- (b) Apresente um termo  $\alpha$ -equivalente em que todas as variáveis tenham nomes distintos.
- (c) Reduza-o até onde for possível.

2. Repita o exercício anterior para os termos

- (a)  $(\lambda x. \lambda y. y \ (y \ x)) \ y \ (\lambda x. (\lambda x. x) \ x)$
- (b)  $(\lambda a. a \ a)(\lambda a. \lambda b. a \ b)$

3. Considere o programa em C:

```
int f(int x) { return x * 2 + 1; }
int main() { return f(f(1)); }
```

- (a) Escreva um termo do cálculo- $\lambda$  que o represente.
- (b) Reduza-o.

4. Considere o programa em SML:

```
fun f x = x * x;
fun g x = f x + f x;
f 2 + g 2;
```

- (a) Escreva um termo do cálculo- $\lambda$  que o represente.
- (b) Reduza-o.

5. Repita o exercício anterior para o programa:

```
fun f x = x * x;
fun g x = f x + f x;
f (3 + g 2);
```

6. Considere o programa em SML:

```
fun f x = x * x;  
fun g x = f x + f x;  
g (f 2);
```

- (a) Escreva um termo do cálculo- $\lambda$  que o represente.
- (b) Reduza o termo da alínea anterior escolhendo sempre o redex mais exterior, mais à esquerda (estratégia *call-by-name*).  
Um redex é mais exterior se está no âmbito de menos abstrações que outro. Por exemplo, no termo  $(\lambda z.z) ((\lambda x.(\lambda y.y) x) w)$ , os redexes  $(\lambda z.z)((\lambda x.(\lambda y.y) x) w)$  e  $(\lambda x.(\lambda y.y) x) w$  são mais exteriores que  $(\lambda y.y) x$ .
- (c) Reduza o mesmo termo escolhendo sempre o redex mais interior, mais à esquerda (estratégia *call-by-value*).

7. Repita o exercício anterior para o programa:

```
fun f y = y * y;  
fun g x = f x;  
f (g 5);
```

8. Considere o programa em SML:

```
fun f x = x + x;  
fun g f x = f x + 1;  
val h = g f;  
h 3;
```

- (a) Escreva um termo do cálculo- $\lambda$  que o represente.
- (b) Reduza-o.

9. Repita o exercício anterior para o programa:

```
fun c f g x = f (g x);  
fun h x = x + 1;  
c h h y;
```