

Exame Linguagens de Programação 2

12 Janeiro 2004

1. Considere a linguagem COMBO definida pela seguinte gramática (as palavras reservadas estão em **bold**, literais e identificadores em *itálico*).

```
Program ::= Decl* Comm*
Decl ::= var Ident | const Ident = Expr
Comm ::= Ident := Expr | while Expr do Comm
        | Comm ; Comm | print Expr
Expr ::= Expr + Expr | Expr - Expr | Number
        | Ident | read Ident
```

- (a) Defina a sintaxe abstracta da linguagem COMBO usando um conjunto de classes Java. Assuma que os valores manipulados pelos programas escritos nesta linguagem são apenas inteiros.
- (b) Defina uma semântica operacional para a linguagem COMBO, introduzindo em cada uma das classes um método declarado como

```
int eval(Env E);
```

Pretende-se que a chamada do método **eval** num objecto Java que represente um programa completo tenha o efeito de o interpretar. Justifique as decisões que tomar, mas assumida uma classe Java **Env** implementando a funcionalidade usual de um ambiente.

2. Considere de novo a linguagem COMBO.
 - (a) Identifique que erros podem ocorrer durante a execução de um programa COMBO. Para cada erro que identifique, apresente um programa COMBO que nele incorra.
 - (b) Defina um sistema de tipos que garanta que programas COMBO não incorram em nenhum dos erros que identificou no ponto anterior. (Sugestão: os únicos tipos que precisa de considerar são **int**, **ok**, e **var**). Defina o seu sistema de tipos usando regras de tipificação sobre asserções da forma:

$$\Delta \vdash E : T$$

onde Δ representa o contexto, E é um programa, comando ou expressão, e T é o tipo atribuído a E .

3. Considera ainda de novo a linguagem COMBO! Pretende-se agora adicionar procedimentos à linguagem, obtendo-se a linguagem COMBO+.

```
Program ::= Decl* Comm*
Decl ::= var Ident | const Ident = Expr
        | proc Ident(Ident) = Program
Comm ::= Ident := Expr | while Expr do Comm
        | Comm ; Comm | print Expr | call Ident(Ident)
Expr ::= Expr + Expr | Expr - Expr | Number
        | Ident | read Ident
```

Como pode observar, para simplificar assumimos que os procedimentos podem ter apenas um argumento, que é passado por referência.

Por exemplo, o programa COMBO+ seguinte

```
var a
var b
proc inc(x) = x := x + 1
b := 2;
inc(b);
print b
```

imprime o valor 3.

O seu objectivo agora é definir a semântica da linguagem COMBO+ por tradução na linguagem CALCREG (CALCREG foi estudada nas aulas práticas, e usada para implementar a linguagem OhSimple).

- (a) Explique claramente como traduziria cada uma das construções da sintaxe abstracta da linguagem COMBO+ na linguagem CALCREG.
- (b) A linguagem COMBO+ que definiu na alinea anterior por “compilação” em CALCREG implementa normalmente a regra de resolução de nomes estática ou dinâmica? Justifique.
4. Discuta a validade da seguinte afirmação: o interpretador da linguagem CALCREG realizado nas aulas práticas efectua uma gestão de memória automática usando o “garbage collector” da linguagem Java apenas por razões de simplicidade, porque é sempre possível determinar estáticamente quando é que um registo ou célula de memória pode ser eliminado.