

Compiladores

Prof. Marcus Ramos – Prova 1 – 06/05/2026

1. (1 ponto) De que maneira o advento das linguagens de alto nível impactou a programação de computadores e impulsionou a criação de compiladores?

Ao trazer mais produtividade, maior legibilidade, maior segurança e mais legibilidade, as linguagens de alto nível tornaram-se indispensáveis na programação de computadores. Como estas linguagens não são interpretadas diretamente pelo hardware, é necessário que as mesmas sejam traduzidas para a linguagem que o hardware entende, o que é feito por meio dos compiladores.

2. (1 ponto) Por que uma gramática com recursão à esquerda não é LL(k) para nenhum valor de k?

Porque os conjuntos *first* da regra recursiva nunca é disjunto em relação ao *first* das demais regras do mesmo símbolo não terminal

3. (1 ponto) Prove que a gramática abaixo é LL(1):

$$S \rightarrow aXbY\#$$
$$X \rightarrow aX \mid b$$
$$Y \rightarrow a \mid b \mid \varepsilon$$

Não terminal S:

$$first_1(aXbY\#) = \{a\}$$

Não terminal X:

$$first_1(aX) = \{a\}$$
$$first_1(b) = \{b\}$$

Não terminal Y:

$$first_1(a) = \{a\}$$
$$first_1(b) = \{b\}$$
$$follow_1(Y) = \{\#\}$$

4. (2 pontos) Obtenha o esboço de um analisador sintático usando o método recursivo descendente para a linguagem gerada pela gramática da questão 3:

```
void parseS () {
  if sc=="a" then {
    acceptIt ();
    parseX ();
    accept ("b");
    parseY ();
    accept ("#")
  }
  else ERRO
}
```

```

void parseX () {
if sc="a" then {
                acceptIt ();
                parseX ()
            }
else if sc="b" then acceptIt ()
else ERRO
}

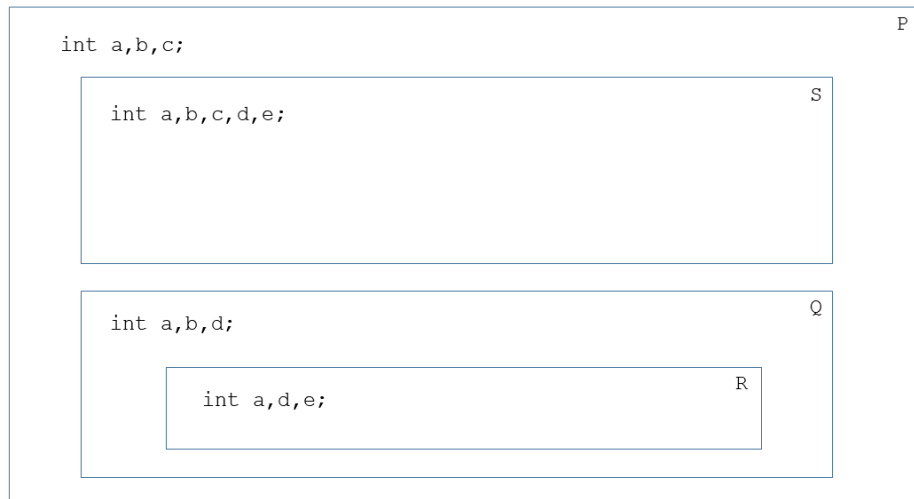
void parseY () {
if sc="a" then acceptIt ()
else if sc="b" then acceptIt ()
else if sc="#" then
else ERRO
}

```

5. (1 ponto) Por que os escopos de um mesmo nome em um programa são sempre disjuntos?

Porque deseja-se vincular cada uso de um nome com uma única declaração (no máximo).

6. (2 pontos) Considere o programa abaixo e mostre a situação da Tabela de Símbolos durante a compilação do bloco R. Suponha uma organização na forma de pilha (com ponteiro de topo de pilha) e mostre nome e nível das variáveis, deixando espaço para o ponteiro da declaração.



Nome	Ponteiro	Nível
A		0
b		0
c		0
a		1
b		1
d		1
a		2

d		2
e		2

← Topo da pilha

7. (1 ponto) Por que a verificação de índices em uma matriz é considerada uma forma de análise de contexto que só pode ser feita, no caso geral, em tempo de execução?

É uma regra de contexto pois o valor da expressão indexadora deve se situar dentro dos limites declarados. Uma expressão indexadora só pode, no caso geral, ser avaliada em tempo de execução, pois depende de informações que são estarão disponíveis durante a execução do programa. Caso ela possa ser avaliada em tempo de compilação, então a análise de contexto neste caso pode ser feita estaticamente.

8. (1 ponto) O que significa uma AST ser “decorada”?

Uma AST decorada é o resultado da fase de análise de contexto, e incorpora na AST criada pela análise sintática as seguintes informações: (i) identificação (vinculação entre usos e declarações dos nomes por meio de ponteiros, levando em conta as regras de escopo da linguagem) e (ii) avaliação dos tipos dos operandos e das operações, permitindo que todas as expressões tenham o seu tipo determinado estaticamente (em linguagens com tipos estáticos), garantindo o seu correto uso e o emprego adequado das operações. “Decoração” da AST é uma atividade da fase de análise de contexto que envolve os dois pontos acima.