

EA876A - Primeira Prova - 29/04/2009

SEM Consulta - Duração: 120 minutos

Questões de igual valor - Utilize uma única folha de papel almaço

Nome: _____ RA: _____

Questão 1: Considere a gramática regular $L (L | D | _)^*$ que representa identificadores válidos em uma dada linguagem de programação (L significa “qualquer letra”, D “qualquer dígito” e $_$ é um caractere). Estabeleça um autômato finito determinístico passo a passo para esta expressão regular (não é necessário minimizar o autômato).

Resposta:

Autômato não determinístico: estado 1 inicial e demais finais

| | | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| L | 2 | 3 | 3 | 3 | 3 |
| D | | 4 | 4 | 4 | 4 |
| _ | | 5 | 5 | 4 | 5 |

Autômato mínimo (não exigido na questão): estado 1 inicial e 2 final

| | | |
|---|---|---|
| | 1 | 2 |
| L | 2 | 2 |
| D | | 2 |
| _ | | 2 |

Questão 2: Seja a seguinte gramática livre de contexto: $G = \{V_t, V_n, P, S\}$, onde $V_t = \{\text{while, cte, var, stmt}\}$, $V_n = \{S, \text{TEST}, \text{CMD}\}$, P dado abaixo:

$S \rightarrow \text{while TEST CMD}$

$\text{TEST} \rightarrow \text{cte}$

$\text{TEST} \rightarrow \text{var}$

$\text{CMD} \rightarrow \text{stmt CMD}$

$\text{CMD} \rightarrow \text{stmt}$

(a) Descreva sucintamente como a tabela de deslocamento e redução (DR) abaixo foi gerada (não é necessário gerar a tabela).

| | while | cte | var | stmt | \$ |
|-------|-------|-----|-----|------|----|
| S | | | | | |
| TEST | | | | D | |
| CMD | | | | | R |
| while | | D | D | | |
| cte | | | | R | |
| var | | | | R | |
| stmt | | | | D | R |
| \$ | D | | | | |

Resposta:

É realizada uma análise para se estabelecer as relações de precedência entre os símbolos terminais e não terminais. Com estas relações de precedência estabelece-se as relações de Wirth-Weber e, a partir destas a tabela de deslçamento e redução.

(b) Utilize o processo DR (passo a passo) para analisar a validade da sentença

while k a=0 b=0

Dica: k é uma variável, a=0 e b=0 são comandos (stmt).

Resposta:

Estado da pilha (topo mais a direita)

\$

\$ while

\$ while cte

\$ while TEST (produção 2)

\$ while TEST stmt

\$ while TEST stmt stmt

\$ while TEST stmt CMD (produção 5)

\$ while TEST CMD (produção 4)

\$ (produção 1, sentença válida)

Questão 3: Escreva uma especificação Lex/Flex para a gramática da questão 1 e uma especificação Yacc/Bison para as produções da gramática da questão 2.

Resposta:

Lex:

DIGIT [0-9]

LETRA [a-zA-Z]

UNDER “ ”

—

Yacc:

S : while TEST CMD;

TEST : cte

| var;

CMD : stmt CMD

| stmt;

Questão 4:

(a) Cite, por meio de exemplos, duas situações onde é possível a otimização do código gerado por um compilador.

Resposta:

Vide exemplos na seção 6.2 do livro (pg. 200)

(b) Utilize uma linguagem simbólica para exemplificar como um compilador C geraria o código abaixo:

```
i = 0
while(i < 10) {
    x[i] = 0;
    i = i + 1;
}
```

Resposta:

```
i := 0
_L1:  if i >= 10 goto _L2
a[i] := 0
i := i + 1
goto _L1
_L2: ...
```

Questão 5:

(a) Qual o processo que o carregador de ligação direta utiliza para determinar a Tabela de Símbolos Externos Globais (GEST)?

Resposta:

1. Seja $L = IPLA$
2. Para cada segmento S_i , seja L_i seu tamanho (registro SD)
 - Para os símbolos X presente em registros ESD (tipo 0) em S_i , compute sua posição no segmento S_i (P_i);
 - Adicione $\langle X, L + P_i \rangle$ na tabela GEST
3. Faça $L = L + L_i$ e vá para 1

(b) Descreva as vantagens e desvantagens do carregamento e ligação dinâmicos.

Resposta:

Vantagens: evita replicação de módulos em diferentes programas; otimiza o uso da memória (módulos não referenciados não ocupam memória)

Desvantagens: esquema mais complexo; programa é suspenso até o carregamento do módulo seja realizado.