

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
ALUNO: THIAGO THEIRY DE OLIVEIRA
DISCIPLINA: COMPILADORES
DOCENTE: SAMUEL XAVIER DE SOUZA

Lista de Compiladores

Questões respondidas: **cap 1:** 1,2,3 ; **cap 2:** 1,2,5,9,14 ; **cap 3:** 3,8 ; **cap 4:** 3;

capítulo 1 -

1 - Considere um navegador Web simples, que tome como entrada uma string de texto em formato HTML e apresente a especificada notação gráfica na tela. O processo de exibição utiliza compilação ou interpretação?

R = Interpretadores e compiladores têm muito em comum, e executam muitas das mesmas tarefas. Dessa forma um navegador pode ser visto como um compilador que traduz uma página da web de uma linguagem textual para uma linguagem gráfica. Também pode ser visto como um interpretador porque analisa a entrada toda vez que é exibida. Assim podemos dizer que o processo pode utilizar tanto compilação, como interpretação. com ênfase em um exemplo citado no livro em que o programa de composição tipográfica que produz PostScript, ou seja, o código que transforma PostScript em pixels é normalmente um interpretador.

2 - Ao projetar um compilador, enfrentam-se muitos dilemas. Quais são as cinco qualidades que você, como usuário, considera mais importantes em um compilador que adquire? Essa lista muda quando você é o construtor do compilador? O que sua lista lhe diz, a respeito de um compilador, que você implementaria?

R= 1 - Como usuário desejaria ter um compilador que identificassem erros dos mais simples aos mais complexo

- 2 - alertasse a existência dos erros identificando o local do erro
- 3 - Um link para documentação sobre o erro ou aviso
- 4 - o código compilado esteja de fato correto
- 5 - reconhecesse mais de uma linguagem

A lista não deveria sofrer alteração, já que devemos produzir um produto que nós mesmo usamos, mas poderíamos acrescentar o fato de um compilador que fosse tão bom como simplificado, ou seja, de boa qualidade e fácil do usuário saber usar. Como por exemplo uma interface no geral e ações como de depurar ser relativamente fácil de utilização.

3 - Compiladores são usados em muitas circunstâncias diferentes. Que diferenças você poderia esperar nos compiladores projetados para as seguintes aplicações:

a. Um compilador Just-in-time usado para traduzir o código da interface de usuário baixado de uma rede?

R = A diferença que podemos esperar neste compilador é uma combinação de duas abordagens tradicionais de tradução para código de máquina: anterior à execução e durante a execução (interpretação), portanto combina algumas vantagens e desvantagens de ambas. A grosso modo, a compilação JIT combina a velocidade do código compilado à flexibilidade da interpretação. A compilação dinâmica permite otimizações que não são possíveis em compilações estáticas, levando em alguns casos a códigos mais rápidos.

b. Um compilador destinado ao processador embutido usado em um telefone celular?

R = Nessa situação a diferença que podemos esperar neste compilador é que tem problemas maiores que o de velocidade para focar. Um celular ou dispositivo móvel requer energia para seu funcionamento, então um compilador que gastasse essa energia não seria viável, logo o compilador que use o mínimo de energia possível para o processo de compilação é fundamental, ou seja que tenha uma eficiência energética e de preferência códigos pequenos.

c. Um compilador usado em um curso de programação introdutório no ensino médio?

R= nessa situação a diferença que podemos esperar neste compilador é que por se tratar da sua utilização em um curso introdutório, um compilador deve fornecer feedback detalhado sobre erros de sintaxe e otimizar o código escrito já que possivelmente os códigos feitos pelo usuário não terão a melhor performance.

d. Um compilador usado para criar simulações de túnel de vento executadas em um processador maciçamente paralelo (onde todos os processadores são idênticos)?

R= nessa situação a diferença que podemos esperar neste compilador é que por se tratar da sua utilização em túneis de vento, que são equipamentos que simulam os deslocamentos de ar e a carga de vento sobre estruturas para a análise do desempenho aerodinâmico, os compiladores concentraram nos problemas de desempenho para que se possa ter a melhor análise. por ser Composto de grande número de processadores idênticos, é necessário fazendo a melhor escolha de registradores e também de escalonamento de instruções.

e. Um compilador que visa programas numericamente intensivos para um grande número de máquinas diversas?

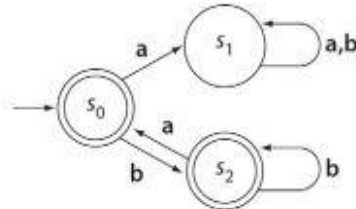
R= Nessa situação a diferença que podemos esperar neste compilador é que por se tratar da sua utilização em programas numericamente intensivos, é preciso fazer uma gestão muito bem da memória, para assim fazer cálculos e alocações mais eficientes.

capítulo 2 -

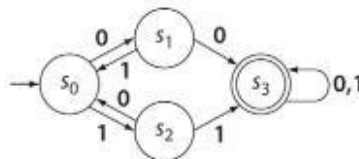
1 -

1. Descreva, informalmente, as linguagens aceitas pelos seguintes FAs:

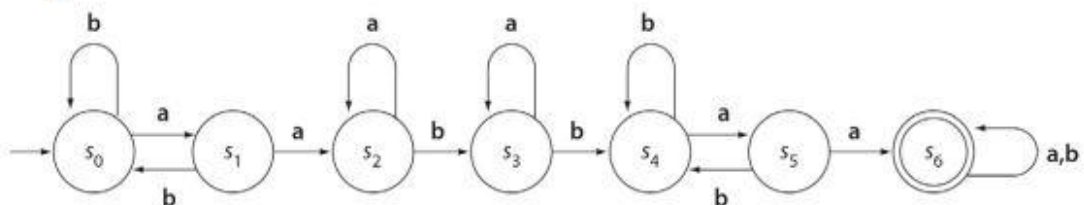
a.



b.



c.

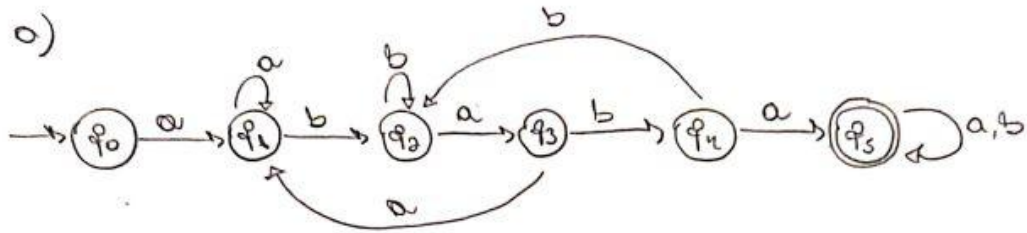


- Linguagem aceita a cadeia vazia, só pode começar com o caractere "b" e todo a tem um antecessor b
- Linguagem que começa com "00" ou "11" ou linguagem que tenha a subcadeia "000" ou "111"
- Linguagem que contém a subcadeia 'aab' e depois 'baa'. não precisam está concatenadas

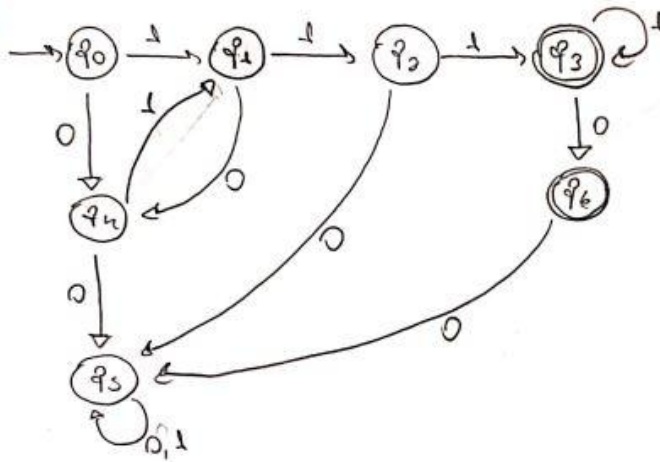
2 - Construa um FA aceitando cada uma das seguintes linguagens:

- $\{w \in \{a, b\}^* \mid w \text{ começa com 'a' e contém 'baba' como uma substring}\}$
- $\{w \in \{0, 1\}^* \mid w \text{ contém '111' como uma substring e não contém '00' como uma substring}\}$
- $\{w \in \{a, b, c\}^* \mid \text{em } w \text{ o número de 'a's no módulo 2 é igual ao de 'b's no módulo 3}\}$

2. a)

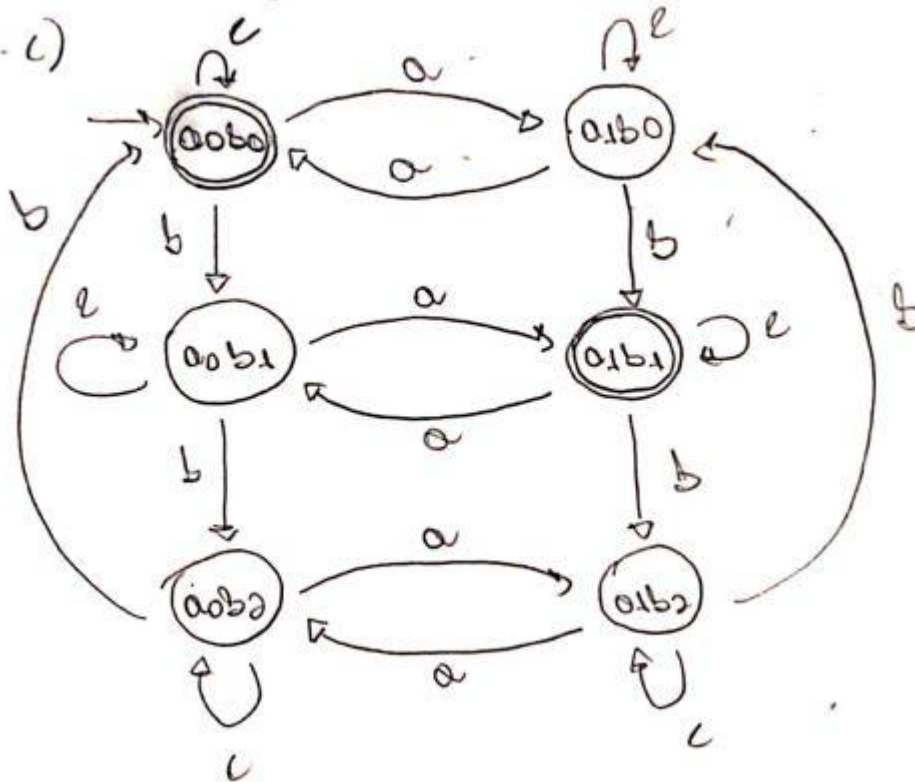


b)



Ativar o Windows
Acesse Configurações pa

2. c)



5. Escreva uma expressão regular para cada uma das seguintes linguagens:

a. Dado um alfabeto $O = \{0, 1\}$, L é o conjunto de todas as strings de pares de 0s e pares de 1s alternados

$R = (1100 (1100)^* (11)) \mid (0011 (0011)^* (00))$.

b. Dado um alfabeto $O = \{0, 1\}$, L é o conjunto de todas as strings de 0s e 1s que contêm um número par de 0s ou de 1s.

$R = (1^*01^*01^*)^+ \mid (0^*10^*10^*)^+$

c. Dado o alfabeto inglês minúsculo, L é o conjunto de todas as strings em que as letras aparecem em ordem lexicográfica crescente.

$R = a * b * c * d * e * f * g * h * i * j * k * l * m * n * o * p * q * r * s * t * u * v * w * x * y * z *$

d. Dado um alfabeto $O = \{a, b, c, d\}$, L é o conjunto de strings $xyzwy$, onde x e w são strings de um ou mais caracteres em O ; y é qualquer caractere único de em O ; e z é o caractere z , tomado de fora do alfabeto. (Cada string $xyzwy$ contém duas palavras xy e wy criadas a partir das letras em O . As palavras terminam na mesma letra, y . E são separadas por z .)

$R = x = (a \mid b \mid c \mid d)^* \quad w = (a \mid b \mid c \mid d)^* \quad y = (a \mid b \mid c \mid d) \quad z = z$

$((a \mid b \mid c \mid d)^* a z (a \mid b \mid c \mid d)^* a) \mid$
 $((a \mid b \mid c \mid d)^* b z (a \mid b \mid c \mid d)^* b) \mid$
 $((a \mid b \mid c \mid d)^* c z (a \mid b \mid c \mid d)^* c) \mid$
 $((a \mid b \mid c \mid d)^* d z (a \mid b \mid c \mid d)^* d)$

e. Dado um alfabeto $O = \{+, -, \times, \div, (,), id\}$, L é o conjunto de expressões algébricas usando adição, subtração, multiplicação, divisão e parênteses sobre ids.

$R =$ Esta linguagem não pode ser descrita em uma expressão regular

9. Em alguns casos, dois estados conectados por um ϵ -movimento podem ser combinados.

a. Sob que conjunto de condições dois estados conectados por um ϵ -movimento podem ser combinados?

$R =$ As condições em que os estados podem ser combinados são os que os estados resultantes continuem respeitando as transições de saída. Ou seja, o acréscimo da transição não interferirá e nem causará uma falha lógica. Transições ϵ fornecem uma maneira conveniente de modelagem de sistemas cujos estados atuais não são conhecidos com precisão. Dessa forma, Transições ϵ não acrescentam qualquer capacidade extra de reconhecimento de linguagens formais.

b. Dê um algoritmo para eliminar ϵ -movimentos.

- Considere-se um estado qualquer $q_i \in Q$. Se houver uma transição em vazio de q_i para q_j , deve-se eliminá-la, copiando-se para a linha que representa o estado q_i todas as transições que partem dos estados q_j para os quais é feita a transição em vazio.
- Esse procedimento corresponde, em notação tabular, à realização de uma fusão (“merge”) entre a linha do estado q_i que contém a transição em vazio para o estado-destino q_j e a própria linha do estado q_j , armazenando-se o resultado novamente na linha correspondente ao estado q_i .
- Havendo mais de uma transição em vazio indicada, deve-se repetir cumulativamente o procedimento para todas elas.

c. Como seu algoritmo se relaciona com a função de ϵ -fechamento usada para implementar a construção de subconjunto?

R= a função ϵ -fechamento depende de certa forma das transições ϵ . Dado um sistema S , o ϵ -fechamento são todos os estados que podem ser alcançados com a transição ϵ . Dessa forma se eliminarmos todas as transições ϵ , não terá estados para ser alcançado com essa transição, logo não terá o ϵ -fechamento.

14. Este capítulo descreveu diversos estilos de implementações de DFA. Uma alternativa usaria funções mutuamente recursivas para implementar um scanner. Discuta as vantagens e desvantagens desta implementação.

R= A principal vantagem é que a implementação é conceitualmente simples, assim tornando, de certa forma, um código mais simples, deixando-o mais enxuto, legível e evitando muitos laços de repetição.

A principal desvantagem é que não é eficiente, pois sua execução poderia ocupar um grande espaço de memória, já que um scanner tradicional ele aloca um registro de ativação, numa função recursiva, poderá gerar várias alocações de registros de ativação, tornando o código ineficiente

capítulo 3 -

3. Quando perguntados sobre a definição de uma gramática livre de contexto não ambígua em um exame, dois alunos deram respostas diferentes. O primeiro a definiu como “uma gramática na qual cada sentença tem uma única árvore sintática por derivação mais à esquerda”. O segundo, “uma gramática na qual cada sentença tem uma única árvore sintática por qualquer derivação”. Qual deles está correto?

R= Os dois estão corretos, pois a definição de uma gramática ambígua é que se alguma sentença tiver mais de uma derivação seja à esquerda ou à direita ela é ambígua. Dessa forma, falar que existe uma única derivação, tanto a esquerda ou qualquer derivação está correto em relação a uma gramática não ambígua.

8. Os parsers top-down e bottom-up criam árvores sintáticas em diferentes ordens. Escreva um par de programas, TopDown e BottomUp, que tomam uma árvore sintática como entrada e imprimem os nós em ordem de construção. TopDown deverá exibir a ordem para um parser top-down, enquanto BottomUp, a ordem para um parser bottom-up.

TopDown:

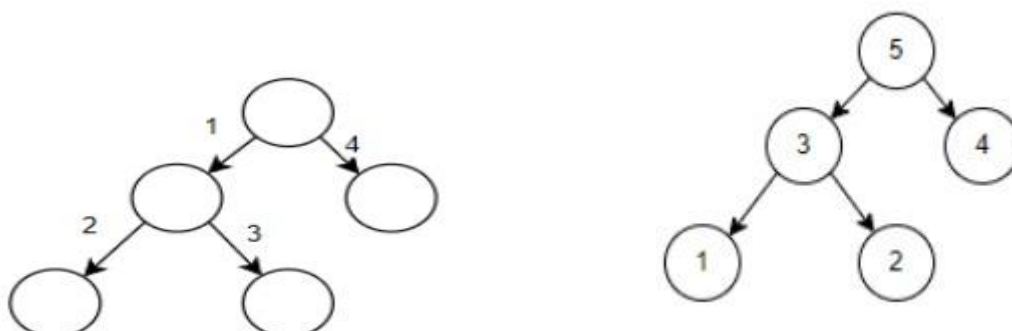
```
Passo 1: Repita os Passos 2 a 4 enquanto TREE != NULL
Passo 2: Escreva DADOS DA ÁRVORE
Passo 3: PREORDER(TREE LEFT)
Passo 4: PREORDER(TREE RIGHT) [END OF LOOP]
Passo 5: END
```

A ordem de percurso é feita indo no nó raiz primeiro, a subárvore esquerda em seguida e, em seguida, a subárvore direita.

BottomUp:

```
Passo 1: Repita os passos 2 a 4 enquanto TREE != NULL
Passo 2: POSTORDER(TREE LEFT)
Passo 3: POSTORDER(TREE RIGHT)
Passo 4: Escreva DADOS DA ÁRVORE [END OF LOOP]
Passo 5: END
```

A ordem de percurso é feita indo na subárvore esquerda primeiro, a subárvore direita em seguida e, finalmente, o nó raiz.



capítulo 4 -

3. Com base nas regras de avaliação a seguir, desenhe uma árvore sintática anotada que mostre como a árvore sintática para $a - (b + c)$ é construída.

Produção	Regras de avaliação
$E_0 \rightarrow E_1 + T$	$\{E_0 \leftarrow \text{.nptr} \leftarrow \text{mknode}(+, E_1.\text{nptr}, T.\text{nptr})\}$
$E_0 \rightarrow E_1 - T$	$\{E_0 \leftarrow \text{.nptr} \leftarrow \text{mknode}(-, E_1.\text{nptr}, T.\text{nptr})\}$
$E_0 \rightarrow T$	$\{E_0 \leftarrow \text{.nptr} \leftarrow T.\text{nptr}\}$
$T \rightarrow (E)$	$\{T.\text{nptr} \leftarrow E.\text{nptr}\}$
$T \rightarrow \text{id}$	$\{T.\text{nptr} \leftarrow \text{mkleaf}(\text{id}, \text{id.entry})\}$

