Período 2012.2
1) Diferencie Token, Padrão e Lexema

2) Na etapa de análise lexica, um identificador também pode ser matcheado como uma palavra chave. Cite dois modos que fazem com que o analisador léxico não dê conflito para o caso acima. (algo assim).
Resposta: 1.Cadastro de palavras reservadas na tabela de símbolos e consultas subseqüentes
2.Diagramas de transição separados para cada palavra reservada

Exercícios (Livro)

3.7. Escreve as definições regulares para as seguintes linguagens:
a) Todas as cadeias de letras que contenham as cinco vogais ordenadas.
alfabeto = [a-z]

consoantes = {bcdfghjklmnpqrstvxyz}

alfabeto* a (consoante|a)* e (consoante|e)* i (consoante|i)* o (consoante|o)* u alfabeto*

b) Todas as cadeias de letras nas quais as letras estão em ordem lexicográfica ascendente.

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*

c) Comentários consistindo em uma cadeia envolvida por /* e */ sem */ intervenientes a menos que figurem entre aspas.

simbolo = [^(*/)]

barrafake = (“*/”)

/* (barrafake | simbolo)* */

d) Todas as cadeias de zero e uns com um número par de zeros e ímpar de uns.

AB*, where
A = 1 | 0 (00 | 11)*(10 | 01)
B = 00 | 11 | (01 | 10)(00 | 11)*(01 | 10)

3.16. Construa um autômato finito não-determinístico para as seguintes expressões regulares. Mostre a sequência de movimentos feita por cada uma no processamento da cadeia de entrada ababbab. (dá pra usar todos os algoritmos pra testar)
a) (a | b)*
b) (a* | b*)*
c) ((E | a) b*)*
d) (a | b)* abb (a | b)*

Slides

1. Descreva as linguagens denotadas pelas seguintes
expressões regulares
a) a(a|b)*a
b) ((ϵ|a)b*)*
c) (a|b)*a(a|b)(a|b)
d) A*ba*ba*ba*
e) (aa|bb)((ab|ba)(aa|bb)*(ab|ba)(aa|bb)*)*

2. Escreva definições regulares para as seguintes linguagens:
a) Todas as cadeias de letras minúsculas que contêm as cinco vogais em ordem
b) Comentários, consistindo em uma cadeia cercada por /* e */, sem um */ intercalado
c) Todas as cadeias de as e bs com um número par de as e um número ímpar de bs

3. Faça diagramas de transição para reconhecer as mesmas
linguagens como cada uma das seguintes expressões
regulares:
a) a(a|b)*a
b) ((ϵ|a)b*)*
c) (a|b)*a(a|b)(a|b)

4. Suponha que tenhamos os seguintes tokens: (1) a palavrachave
while; a palavra-chave when; e identificadores
consistindo em sequencias de letras e dígitos, começando
com uma letra. Mostre:
a) O NFA para esses tokens;
b) O DFA para esses tokens.

Resposição

Questão 01 – Qual a diferença entre compiladores convencionais Java e Compiladores JITs Java? Explique do ponto de vista arquitetural de um compilador. (1,0 ponto)

Em sistemas computacionais baseados em máquinas virtuais, como Java e .NET, o processo de compilação traduz o código fonte para uma representação intermediária conhecida pelo termo bytecode. Esta representação intermediária não está associada a nenhum código de máquina específico e pode ser transportada para várias arquiteturas de computador distintas. Em cada arquitetura específica, essa representação intermediária é interpretada - ou executada em uma máquina virtual.

Nos ambientes que oferecem o recurso de JIT, a máquina virtual responsável pela execução dos bytecodes resultantes da compilação do programa fonte realiza a tradução desse bytecode para código de máquina nativo enquanto o executa. No caso mais comum, cada trecho de código é traduzido no instante em que está para ser executado pela primeira vez, daí derivando o nome "just-in-time".

Analisador léxico é um importante módulo de um compilador. Ele é responsável por ler os caracteres de entrada de um programa fonte, agrupá-los em lexemas e produzir como saída uma sequência de tokens para cada lexema no programa fonte. Todos os lexemas válidos em um programa fonte e o seu reconhecimento é dado pelo uso de três notações que desempenham um papel fundamental em um analisador léxico: (i) expressões regulares; (ii) Autômatos Finitos Não-Deterministas (NFA); e (iii) Autômatos Finitos Deterministas (DFA). A implementação de um analisador léxico descrito através de expressões regulares requer a simulação de um NFA ou um DFA. Nas questões a seguir, você vai fazer uso destas três notações e de algoritmos construídos para simular autômatos (NFAs ou DFAs) a partir de expressões regulares, ou seja, algoritmos que simulam o reconhecimento dos padrões de lexemas do programa fonte.

Questão 02 – Sendo Σ = {a,b} o alfabeto considerado, escreva a definição regular para a linguagem que descreve todas as cadeias de letras minúsculas (vogais e consoantes) que contêm exatamente cinco vogais diferentes e em ordem, além das consoantes, que podem ocorrer em qualquer lugar, repetidamente ou não, e em qualquer ordem. (1,0 ponto)

Questão 03 – Converta a expressão regular (a|b)*aa(a|b)* (Sendo Σ = {a,b} o alfabeto considerado) para um NFA, usando o algoritmo McNaughton-Yamada-Thompson. Mostre o passo a passo da construção do NFA. (4,0 pontos)

Questão 04 – Converta a expressão regular (a|b)*aa(a|b)* (Sendo Σ = {a,b} o alfabeto considerado) diretamente em um DFA usando as funções nullable, firstpos, lastpos e followpos. Indique o cálculo de cada uma destas funções e sua aplicação para construir o DFA. (4,0 pontos)

(MINI-PROVA 1 2010.2)

1) Cite 3 caracteríticas que diferencia o mecanismo de funcionamneto do LEX dos outros geradores léxicos.

2) Diferencie compiladores JIT dos Java normais.

3) Faça definições regulares para uma cadeia que tenha as 5 vogais (somente) e em orddem. Pode ter consoantes.

4) Faça o NFA da expressão (a|b)*aa(a|b)* usando os algoritmos aprendidos em sala.

5) Calcule firstpos, lastpos, followpos e nullable para (a|b)*aa(a|b)* e a partir disso construa seu DFA.

(MINI-PROVA 1 2012.1)

1)

a) Explicar a importancia dos DFA's e NFA's no processo de compilação e qual a relação entre as expressões regulares que descrevem a linguagem e os automatos. (1,5)
*(Fábio) Uma importante estrutura de dados para o compilador é a **tabela de símbolos**. Nela são guardadas as informações sobre as **entidades** do programa fonte. Para popular a tabela de símbolos é necessário que seja feita uma varredura no código do programa-fonte, **identificando entidades como identificadores, operadores, etc.** Essas entidades são descritas por **padrões**, que são representados por **expressões***

***regulares**. Uma expressão regular pode ser representada por um DFA ou NFA.*

b) Explicar as 3 seções de um programa LEX.(1,5)
(Fábio)
Declarações: Declaração de variáveis, constantes e definições regulares.
Regras de tradução: Define as ações a serem tomadas para cada padrão identificado.
Funções auxiliares: Funções adicionais utilizadas nas ações.

2) Escreva uma ou mais definições regulares sobre o alfabeto SIGMA={0,1} que descreva todas as cadeias de múltiplas de 4 no alfabeto binário (1,5)
(Fábio)
elemento_canonico-> 0000|0001|0010|0011| … | 1111
L(r) -> (elemento_canonico)+

3) Explicar as regras base e indução do algorimto McNaughlan-Yamada-Thompson. E explicar a importância desse algoritmo no compilador. (2,0)
(Fábio)
Tem que ter imagens, verificar slides de 88 a 94.

4) Converta a expressão regular bba* (a|b) diretamente para um DFA. Calculando Firstpos, Lastpos, followpos e nullable assim como a tabela de transição. (3,5)