



Compiladores – Prova #2

Nome:

Matrícula:

Data:

Observações:

- (a) A prova é individual e sem consulta, sendo vedado o uso de calculadoras e de telefones celulares.
 - (b) A interpretação dos comandos das questões faz parte da avaliação.
 - (c) A nota da prova será igual a 0 (zero) caso o estudante consulte algum material durante a prova, ou receba ou ofereça qualquer ajuda a outro estudante durante a prova.
 - (d) As questões podem ser resolvidas a lápis ou à caneta. Entretanto, a resposta final deve ser **destacada** de forma clara (circulada, sublinhada, reforçada, indicada, etc...) para que a questão seja devidamente corrigida.
 - (e) O grampo da prova **não deve ser removido**. Caso o grampo seja removido, a nota da prova será igual a 0 (zero).
-

Parte A

1. (7 pontos) Complete a sentença: “Uma gramática livre de contexto é composta por terminais, que são símbolos básicos para a formação de cadeias, e por não-terminais, os quais impõem uma estrutura hierárquica na linguagem; um deles é designado símbolo de partida, e o conjunto de cadeias geradas por ele é a linguagem definida pela gramática. As relações entre estes dois componentes fundamentais da gramática são dadas por meio de produções.”
2. (5 pontos) Considere a gramática abaixo:

$$S \rightarrow SaS \mid bSbS \mid a \mid b$$

Encontre uma derivação mais à direita da cadeia *babbab* com, no máximo, 5 passos.

$$\begin{aligned} S &\Rightarrow bSbS \Rightarrow bSb(SaS) \Rightarrow bSb(Sab) \\ &\Rightarrow bSb(bab) \Rightarrow bab(bab) \end{aligned}$$

3. (8 pontos) **Assinale a alternativa correta.** Quantas são as subcadeias de uma cadeia de tamanho 5?
(A) (A) 5
(B) (B) 6
(C) (X) 16
(D) (D) 32
4. (10 pontos) **Julgue os itens abaixo.** Em cada item, preencha os parêntesis com V (verdadeiro) ou F (falso).

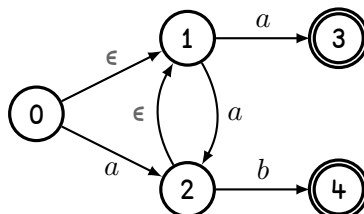
- (F) Um prefixo de *s* é uma cadeia obtida a partir da remoção de zero ou mais caracteres do início de *s*.
- (V) Uma subsequência de *s* é uma cadeia obtida pela remoção de zero ou mais símbolos de *s*, não necessariamente contíguos.
- (V) Dada uma linguagem *L*, a cadeia *ε* pertence a linguagem *L*^{*}.

(V) Um autômato finito não-determinístico admite transições- ϵ .

(F) Erros léxicos surgem da inconformidade da disposição dos tokens em relação às regras gramaticais.

Parte B

5. Considere o grafo de transições do AFN abaixo.



(i.) (10 pontos) Converta este AFN para uma gramática livre de contexto G .

$A_0 \rightarrow A_1 \mid aA_2$
 $A_1 \rightarrow aA_2 \mid aA_3$
 $A_2 \rightarrow A_1 \mid bA_4$
 $A_3 \rightarrow \epsilon$
 $A_4 \rightarrow \epsilon$

(ii.) (5 pontos) Descreva os elementos da linguagem gerada pela gramática G .

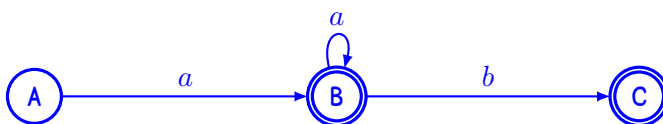
Todas as cadeias não-vazias de a 's, seguidas ou não por um único b .

6. (15 pontos) Determine uma expressão regular que defina o conjunto de todas as cadeias formadas pelos caracteres a e b tais que símbolos adjacentes sejam distintos, incluindo a cadeia vazia. Exemplos de cadeias válidas: ϵ , aba , $baba$, $ababababab$.

$(ab)^*a? \mid (ba)^*b?$

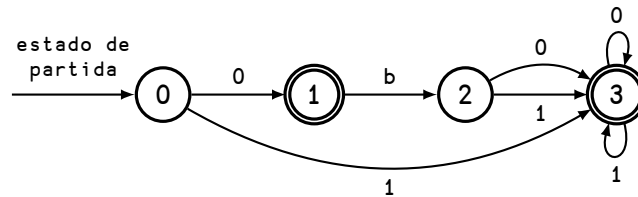
7. (20 pontos) Usando o algoritmo apresentado em aula, converta o AFN da Questão 5 para um AFD. **Atenção:** cada estado do AFD corresponde a um conjunto **não-vazio** de estados do AFN.

Fechamento- $\epsilon(0) = \{0, 1\} = A$
 Fechamento- $\epsilon(\text{Movimento}(A, a)) = \text{Fechamento-}\epsilon(\{2, 3\}) = \{1, 2, 3\} = B$
 Fechamento- $\epsilon(\text{Movimento}(A, b)) = \text{Fechamento-}\epsilon(\{\}) = \{\}$
 Fechamento- $\epsilon(\text{Movimento}(B, a)) = \text{Fechamento-}\epsilon(\{2, 3\}) = \{1, 2, 3\} = B$
 Fechamento- $\epsilon(\text{Movimento}(B, b)) = \text{Fechamento-}\epsilon(\{4\}) = \{4\} = C$
 Fechamento- $\epsilon(\text{Movimento}(C, a)) = \text{Fechamento-}\epsilon(\{\}) = \{\}$
 Fechamento- $\epsilon(\text{Movimento}(C, b)) = \text{Fechamento-}\epsilon(\{\}) = \{\}$



Parte C

8. Considere o diagrama de transições abaixo:



- (i.) (6 pontos) Determine a sequência de pares (estado, rótulo) que identificam o caminho que cadeias abaixo seguirão no diagrama.
- (a) 101
(0, 1) → (3, 0) → (3, 1)
- (b) 0b001
(0, 0) → (1, b) → (2, 0) → (3, 0) → (3, 1)
- (c) 10110
(0, 1) → (3, 0) → (3, 1) → (3, 1) → (3, 0)
- (ii.) (24 pontos) Implemente, em C, C++ ou Python, uma função que receba uma string s de tamanho N como parâmetro e que retorne verdadeiro, se todos os N caracteres de s compõem uma cadeia válida da linguagem definida pelo diagrama; ou falso, caso contrário. Use apenas elementos básicos da linguagem, sem recorrer a bibliotecas externas ou expressões regulares. Use o diagrama como guia: a função deve aceitar todas as cadeias que o diagrama aceita, e recusar todas as cadeias que o diagrama recusa.

Importante: Escreva o código com letra legível, de forma organizada e clara, numerando as linhas. O código não deve exceder 50 linhas.

Solução em C:

```
1 int is_valid(int N, const char *s)
2 {
3     int state = 0;
4
5     for (int i = 0; i < N; ++i)
6     {
7         char c = s[i];
8
9         switch (state) {
10            case 0:
11                if (c == '0')
12                    state = 1;
13                else if (c == '1')
14                    state = 3;
15                else
16                    return 0;
17
18            case 1:
19                if (c == 'b')
20                    state = 2;
21                else
22                    return 0;
23
24            case 2:
25                if (c == '0' || c == '1')
26                    state = 3;
```

```

27         else
28             return 0;
29
30     case 3:
31         if (c != '0' || c != '1')
32             return 0;
33     }
34 }
35
36 return state == 3;
37 }

```

Solução em Python:

```

1 def isValid(s):
2
3     if s == '0':
4         return True
5
6     if s[0] == '1':
7         s = s[1:]
8     elif s[:2] == '0b':
9         s = s[2:]
10    else:
11        return False
12
13    xs = [c in '01' for c in s]
14
15    return all(xs)

```