



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) O gabarito deve ser preenchido com caneta esferográfica azul ou preta. Em uma questão de múltiplas escolhas, múltiplas marcações anularão a referida questão.
- (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. (70 pontos) Assinale a alternativa correta.

1. Considere as afirmações abaixo.

- I. Um token é um símbolo não-terminal de uma gramática.
- II. Um mesmo token pode ser representado por lexemas distintos.

Podemos afirmar que

- (A) ambas afirmações estão corretas
- (B) ambas afirmações estão incorretas
- (C) a afirmativa I está correta e a afirmativa II está incorreta
- (X) a afirmativa I está incorreta e a afirmativa II está correta

2. Considere o seguinte trecho de código, escrito em linguagem C:

```
int 1 = x;
```

Pode-se afirmar que este trecho

- (A) contém um erro léxico
- (X) contém um erro sintático
- (C) contém um erro semântico
- (D) não contém erros

3. Considere três estratégias de tratamento da buferização:

- I. implementar a buferização explicitamente em linguagem de montagem.
- II. utilizar uma linguagem de programação e a buferização a ela atrelada.
- III. usar um gerador léxico, que será o responsável pela buferização.

Qual sequência de identificadores elenca estas três estratégias, da mais complexa para a mais simples?

- (X) I, II, III
- (B) I, III, II
- (C) II, I, III
- (D) III, II, I

4. Considere, no contexto da implementação da buferização, o uso de um par de buffers, com N caracteres cada. Assuma que cada caractere ocupe 1 byte de memória. Qual é a área ocupada por este par de buffers, se $N = 6$ e forem usados caracteres sentinelas para sinalizar o final de cada bloco?

- (A) 12
- (X) 14
- (C) 16
- (D) 18

5. Considere os conjuntos

$$A = \{ \heartsuit, \diamondsuit, \spadesuit, \clubsuit \}$$

e

$$B = \{ n \in \mathbb{N} \mid 2^{100} \text{ divide } n \}$$

É verdade que

- (A) ambos conjuntos são alfabetos
 - (X) apenas o conjunto A é um alfabeto
 - (C) apenas o conjunto B é um alfabeto
 - (D) nenhum dos dois conjuntos é um alfabeto
6. Seja s uma cadeia sobre um alfabeto Σ e t a cadeia obtida a partir da remoção de zero ou mais caracteres do fim de s . Então, em relação a s , a cadeia t é
- (X) prefixo
 - (B) sufixo
 - (C) subcadeia
 - (D) subsequência
7. Seja $s = \text{"abbaabab"}$. Qual é o menor natural n tal que existe uma cadeia t com $s = t^n$?
- (X) 1
 - (B) 2
 - (C) 3
 - (D) 4
8. Sejam L e M duas linguagens. Qual é a notação para linguagem abaixo?

$$N = \{ st \mid s \in L \wedge t \in M \}$$

- (A) $N = L \cup M$
- (X) $N = LM$
- (C) $N = L^*$
- (D) $N = L^+$

9. Seja $L = \{ a, b \}$ e $M = \{ x, y \}$. A cadeia $axyb$ pertence a qual linguagem?

- (A) $L^3 M$
- (B) ML^3
- (C) $L^2 M^2$
- (X) $LM^2 L$

10. Sejam r, s, t expressões regulares e ϵ a cadeia vazia. A expressão

$$r^{**} = r^*$$

corresponde a qual propriedade das expressões regulares?

- (A) associativa
- (B) distributiva
- (X) idempotente
- (D) elemento neutro

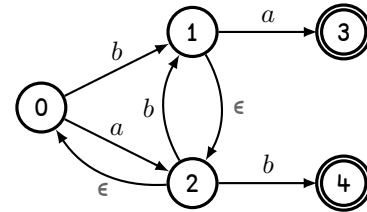
11. Dadas quatro cadeias

$$s_1 = \epsilon, s_2 = baab, s_3 = bbab \text{ e } s_4 = aaaa,$$

quantas dentre estas cadeias fazem parte da linguagem definida pela expressão regular $b^*(ab^+)^+$?

- (A) 0
- (B) 1
- (C) 2
- (X) 3

12. Considere o grafo de transições do AFN abaixo.



No grafo de transições do AFD equivalente, obtido por meio do algoritmo de conversão de AFN para AFD apresentado em aula, quantas são as arestas rotuladas por b que partem de e apontam para um mesmo estado?

- (X) 1
- (B) 2
- (C) 3
- (D) 4

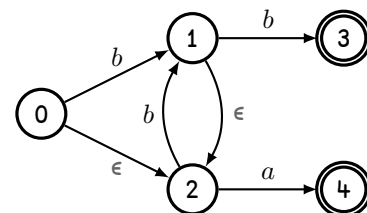
13. Dada a gramática livre de contexto

$$\begin{aligned} S &\rightarrow aSb \mid bTa \\ T &\rightarrow aT \mid bS \mid \epsilon, \end{aligned}$$

quantos são os passos da derivação, a partir de S , da cadeia $abab$?

- (X) 3
- (B) 4
- (C) 5
- (D) 6

14. Considere o diagrama de transições do AFN abaixo:

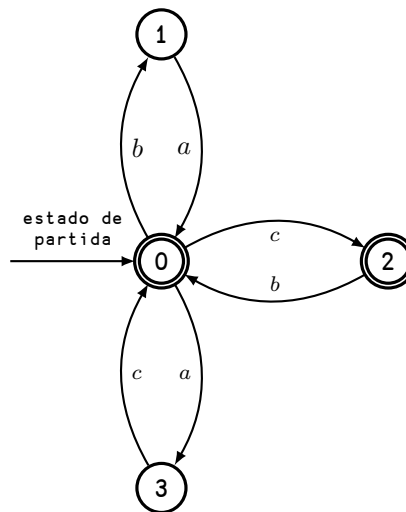


Seja $X = \{ 0 \}$. Qual é o Fechamento- $\epsilon(X)$?

- (A) $\{ 4 \}$
- (X) $\{ 0, 2 \}$
- (C) $\{ 1, 2 \}$
- (D) $\{ 1, 2, 3 \}$

Parte B. (30 pontos) Resolva a questão a seguir.

15. Considere o AFD abaixo:



Implemente, em C, C++ ou Python, a função `is_valid()`, que recebe uma string `s` como parâmetro e que retorna verdadeiro, se todos os caracteres de `s` compõem uma cadeia válida da linguagem definida pelo AFD; 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.

Seguem abaixo alguns testes unitários para esta função.

```
assert(is_valid("bacbac"));
assert(is_valid("ac"));
assert(is_valid("cbc"));
assert(not is_valid("abc"));
assert(is_valid(""));
```

Importante: Escreva o código com letra legível, de forma organizada e clara, usando as folhas pautadas do final da prova. O código não deve exceder 50 linhas.

```
1 int in_valid(const char *s)
2 {
3     int state = 0;
4     char *p = NULL;
5
6     for (p = s; *p; ++p)
7     {
8         if (*p != 'a' && *p != 'b' && *p != 'c')
9             return 0;
10
11         switch (state) {
12             case 0:
13                 if (*p == 'a')
14                     state = 3;
15                 else if (*p == 'b')
16                     state = 1;
17                 else
18                     state = 2;
19                 break;
20
21             case 1:
22                 if (*p == 'a')
23                     state = 0;
24                 else
```

```

25         return 0;
26
27     case 2:
28         if (*p == 'b')
29             state = 0;
30         else
31             return 0;
32
33     case 3:
34         if (*p == 'c')
35             state = 0;
36         else
37             return 0;
38     }
39 }
40
41 return state == 0 || state == 2;
42 }

```

```

1 def is_valid(s):
2
3     if len(s) % 2 == 1 and s[-1] != 'c':
4         return False
5
6     for i in range(0, len(s) // 2):
7         t = s[2*i:2*i + 2]
8
9         if t not in [ 'ba', 'cb', 'ac' ]:
10            return False
11
12 return True

```