EDITAL PROPEG Nº 26/2019 – PROVA DE CONHECIMENTOS ESPECÍFICOS

Chave de Respostas

Questão 1

As definições para automato finitos, solicitadas na questão 1, são apresentadas nas definições 1, 2 e 3. A definição para expressões regulares é feita na definição 4. As definições para gramáticas regulares são apresentadas nas definições 5 e 6.

Definição 1 *Um* **autômato finito determinístico** *(AFD) é uma quíntupla* $M = (Q, \Sigma, \delta, q_0, F)$*, onde:*

- Q *é um conjunto finito de* **estados internos***;*
- Σ *é um conjunto finito de símbolos, chamado de* **alfabeto de entrada***;*
- $\delta : Q \times \Sigma \rightarrow Q$ *é uma função, chamada de* **função de transição** *entre os estados;*
- $q_0 \in Q$ *é o* **estado inicial***;*
- $F \subseteq Q$ *é o conjunto de* **estados finais***, ou* **estados de aceitação***.*

Definição 2 *Um* **autômato finito não determinístico** *(AFN) é uma quíntupla* $M = (Q, \Sigma, \delta, q_0, F)$*, onde:*

- Q *é um conjunto finito de* **estados internos***;*
- Σ *é um conjunto finito de símbolos, chamado de* **alfabeto de entrada***;*
- $\delta : Q \times \Sigma \rightarrow 2^Q$ *é a* **função de transição** *entre os estados, cujo resultado, diferentemente de um AFD, não é um único elemento de* Q *e sim um subconjunto dele;*
- $q_0 \in Q$ *é o* **estado inicial***;*
- $F \subseteq Q$ *é o conjunto de* **estados finais***, ou* **estados de aceitação***.*

Definição 3 *Um* **autômato finito não determinístico com movimentos vazios** *(AFN-*λ*) é uma quíntupla* $M = (Q, \Sigma, \delta, q_0, F)$*, onde:*

- Q *é um conjunto finito de* **estados internos***;*
- Σ *é um conjunto finito de símbolos, chamado de* **alfabeto de entrada***;*
- $\delta : Q \times \Sigma \cup \{\lambda\} \rightarrow 2^Q$*, cujo resultado a partir de uma entrada vazia também é não determinístico;*
- $q_0 \in Q$ *é o* **estado inicial***;*
- $F \subseteq Q$ *é o conjunto de* **estados finais***, ou* **estados de aceitação***.*

Definição 4 *Seja Σ um alfabeto. Então*

1. *$\emptyset$, λ e x, para todo $x \in \Sigma$, são expressões regulares, neste caso, chamadas de* expressões regulares primitivas*;*
2. *Se r_1 e r_2 são expressões regulares, então $r_1 + r_2$, $r_1 \cdot r_2$, r_1^* e (r_2) são expressões regulares;*
3. *Uma cadeia é uma expressão regular se, e somente se, ela pode ser derivada de expressões regulares primitivas, aplicando-se a regra 2 um número finito de vezes.*

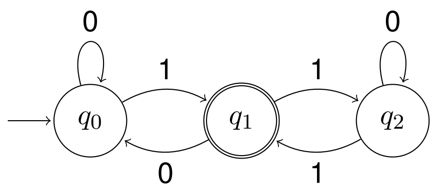
Definição 5 *Seja $G = (V, T, P, S)$ uma gramática. Sejam A e B elementos de V e w uma cadeia de T^*. Então G é uma* **Gramática Linear** *se todas as suas produções encontram-se em uma, e somente uma, das seguintes formas:*

1. *Gramática Linear à Direita (GLD): $A \rightarrow wB$ ou $A \rightarrow w$*
2. *Gramática Linear à Esquerda (GLE): $A \rightarrow Bw|w$*
3. *Gramática Linear Unitária à Direita (GLUD): é uma GLD com $\mathcal{N}(w) \leq 1$*
4. *Gramática Linear Unitário à Esquerda (GLUE): é uma GLE com $\mathcal{N}(w) \leq 1$*

Definição 6 *Seja G uma gramática, ela é dita* **Gramática Regular** *(GR), se G é uma gramática linear.*

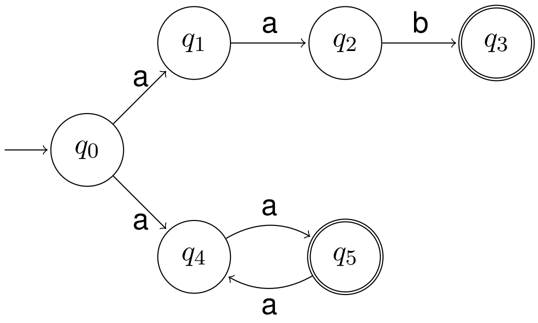
Exemplos para automato finitos, solicitados na questão 1, são apresentadas nas figuras 1, 2 e 3. A definição para expressões regulares é feita na definição 4. As definições para gramáticas regulares são apresentadas nas definições 5 e 6.

Figura 1 – Diagrama de automato finito determinístico.



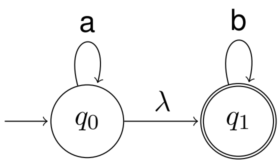
Fonte: Elaboração própria.

Figura 2 – Diagrama de um de automato finito não determinístico.



Fonte: Elaboração própria.

Figura 3 – Diagrama de um de automato finito não determinístico com movimento vazio.



Fonte: Elaboração própria.

Um exemplo para expressão regular, solicitado na questão 1, pode ser o seguinte: considerando $\Sigma = \{a, b, c\}$, podemos verificar que a cadeia $(a + b \cdot c)$ é uma expressão regular, com base na definição 4, aplicada a seguir:

1. Tomando $r_1 = b$ e $r_2 = c$ como ER, temos que $r_3 = r_1 \cdot r_2$ é uma ER;
2. Tomando $r_4 = a$ como ER, temos que $r_5 = r_4 + r_3$ é uma ER;

Um exemplo para gramática regular, solicitado na questão 1, é apresentado a seguir:

- GLD: $G = (\{S, A\}, \{a, b\}, P, S)$,
  onde $P = \{S \rightarrow aS|bS|A, A \rightarrow aa|bb\}$.
- GLE: $G = (\{S, A\}, \{a, b\}, P, S)$,
  onde $P = \{S \rightarrow Aaa|Abb, A \rightarrow Aa|Ab|\lambda\}$.

Questão 2

As implementações dos procedimentos, solicitados na questão 2, são apresentados, respectivamente, nas listagens 1 e 2.

Listagem 1 – Código para para o item (a) da questão 2.

```
Lista* lst_remove (Lista* lista, int chave) {
  Lista* ant = NULL;
  Lista* p lista;
  while(p!=NULL && p->info != chave) {
    ant = p;
    p = p->prox;
  }
  if(p==NULL)
    return lista;
  if(ant==NULL)
    lista = p->prox;
  else
    ant->prox = p->prox;
  free(p);
  return lista;
}
```

Fonte: Elaboração Própria.

Listagem 2 – Código para para o item (b) da questão 2.

```
Lista* lst_inverte (Lista* lista) {
  Lista* p = lista;
  Lista* ant = NULL;
  Lista* aux = p->prox;
  while (aux!=NULL) {
    p->prox = ant;
    ant = p;
    p = aux;
    aux = aux->prox;
  }
  p->prox = ant;
  lista = p;
  return lista ;
}
```

Fonte: Elaboração Própria.

Questão 3

As definições para automato de pilha e gramáticas livres de contexto solicitadas na questão 3 são apresentadas nas definições 7 e 8, respectivamente.

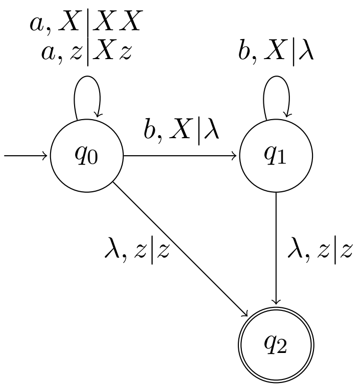
Definição 7 *Um* **autômato de pilha não determinístico** *(APN) é uma sétupla* $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$*, onde:*

- Q *é um conjunto finito de* **estados internos***;*
- Σ *é um conjunto finito de símbolos, chamado de* **alfabeto de entrada***;*
- Γ *é um conjunto finito de símbolos, chamado de* **alfabeto de pilha***;*
- $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to 2^{Q \times \Gamma^*}$ *é a* **função de transição***;*
- $q_0 \in Q$ *é o* **estado inicial***;*
- $z \in \Gamma$ *é o* **símbolo inicial da pilha***;*
- $F \subseteq Q$ *é o conjunto de* **estados finais***, ou* **estados de aceitação***.*

Definição 8 *Seja* $G = (V, T, P, S)$ *uma gramática. Ela é dita* **gramática livre de contexto** *se todas as produções em* P *têm a forma* $A \to \alpha$*, com* $A \in V$ *e* $\alpha \in (V \cup T)^*$*.*

Um exemplo de automato de pilha, solicitado na questão 3, é apresentado na figura 4.

Figura 4 – Representação diagramática de um de automato de pilha.



Fonte: Elaboração própria.

Um exemplo para gramática livre de contexto, solicitado na questão 3, é apresentado a seguir: $G = (\{S\}, \{a, b\}, P, S)$, com $P = \{S \to aSb|\lambda\}$.

Questão 4

As respostas para os dois itens da questão 4 são apresentados a seguir:

- As definições dos dois algoritmos básicos de buscas em grafos são apresentadas a seguir:
  - **Largura** (do inglês: *breadth-first search*): a estratégia expande a fronteira entre os vértices descobertos e não descobertos uniformemente por meio da largura da fronteira, como se fossem círculos concêntricos gerados por uma pedra que se deixa cair em uma superfície de água completamente parada.
  - **Profundidade** (do inglês: *depth-first search*): a estratégia é de buscar sempre que possível, o mais profundo no grafo. As arestas são exploradas a partir do vértice V mais recentemente descoberto que ainda possui arestas não exploradas saindo dele. Quando todas as arestas adjacentes a V tiverem sido exploradas, a busca anda para trás para explorar vértices que saem do vértice do qual V foi descoberto. O processo continua até que sejam descobertos todos os vértices alcançáveis a partir do vértice original.
- Os caminhos percorridos pelos dos algoritmos de busca são:
  - Busca usando *breadth-first search*: 0 2 3 1 5 7 8 4 6 9
  - Busca usando *depth-first search*: 0 2 1 4 8 5 6 3 7 9