**1ª LISTA DE EXERCÍCIOS PARTE 6**

**1.**

**a)**

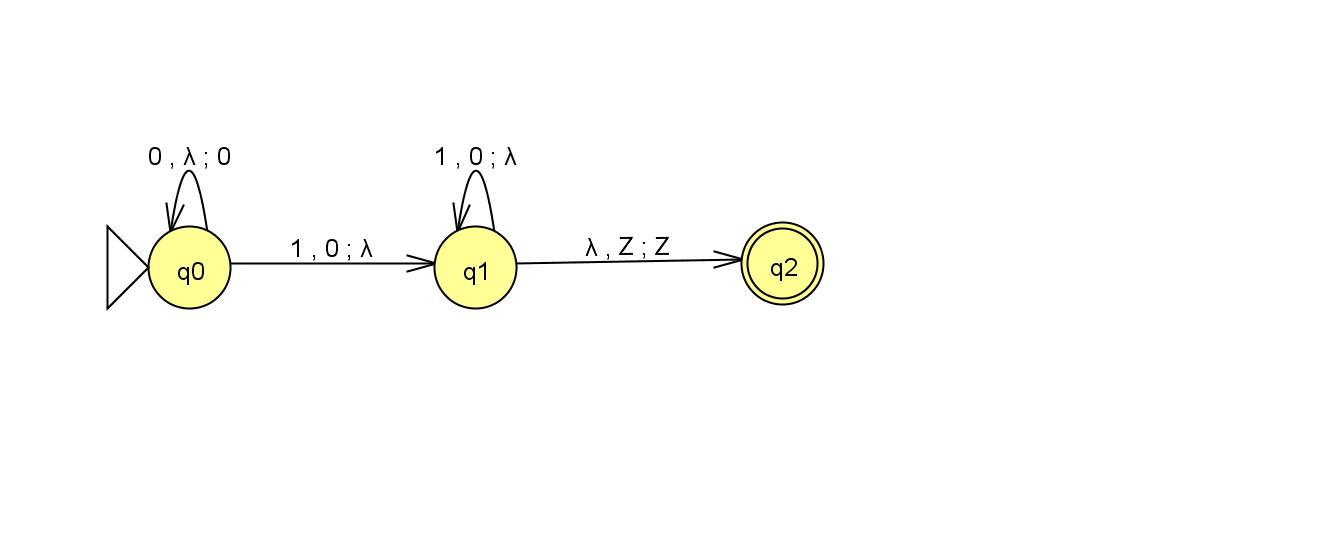


Figura 1. Questão 1, letra A

Fonte: Própria

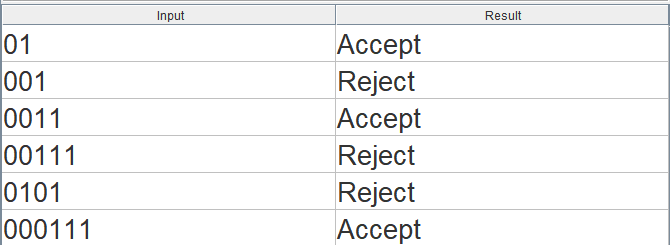


Figura 2. Alguns exemplos de entradas e resultados com base no AP da Questão 1 letra A

Fonte: Própria

**b)**

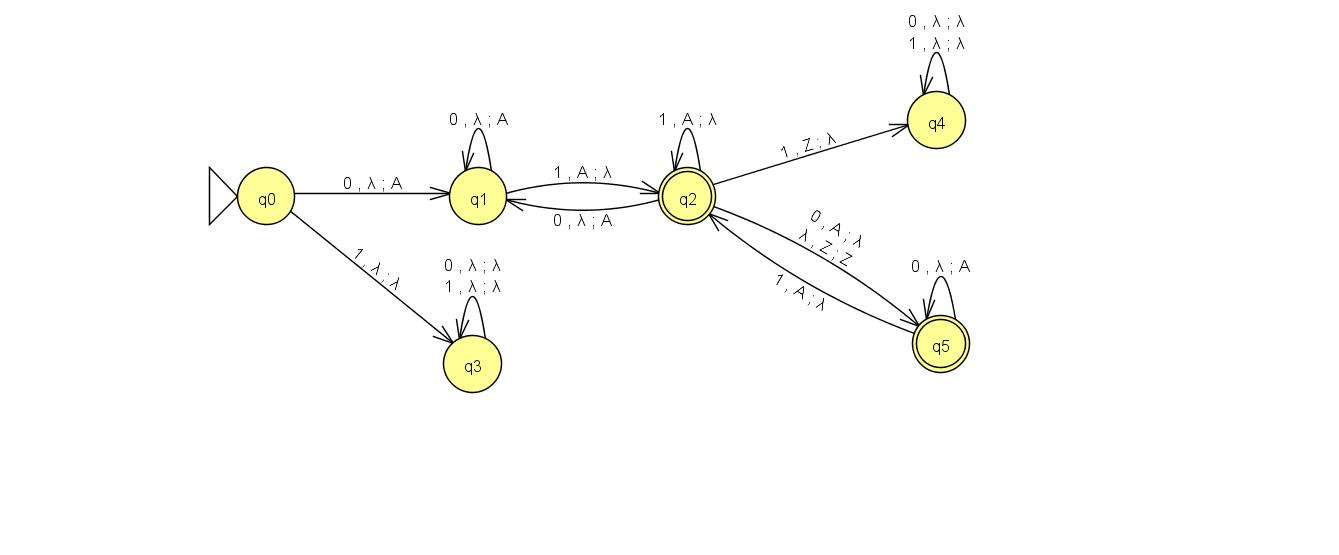


Figura 3. Questão 1, letra B

Fonte: Própria

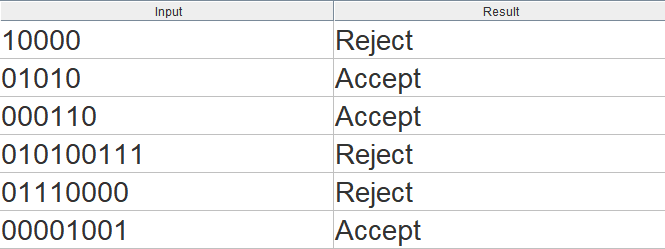


Figura 4. Alguns exemplos de entradas e resultados com base no AP da Questão 1 letra B

Fonte: Própria

**c)**

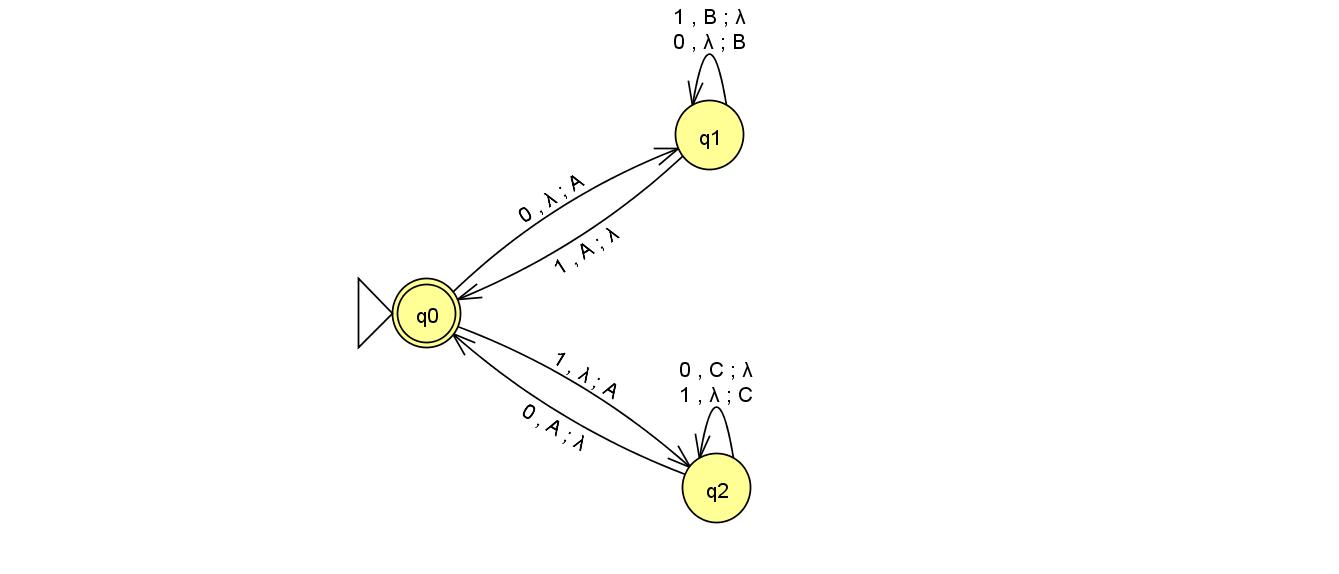


Figura 5. Questão 1, letra C

Fonte: Própria

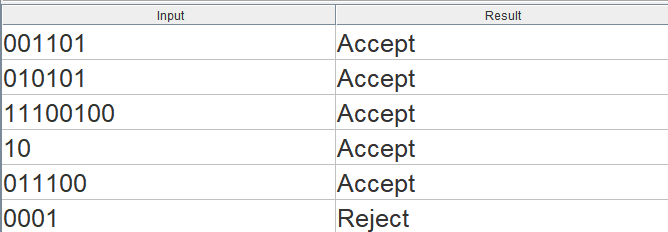


Figura 6. Alguns exemplos de entradas e resultados com base no AP da Questão 1 letra C

Fonte: Própria

**d)**

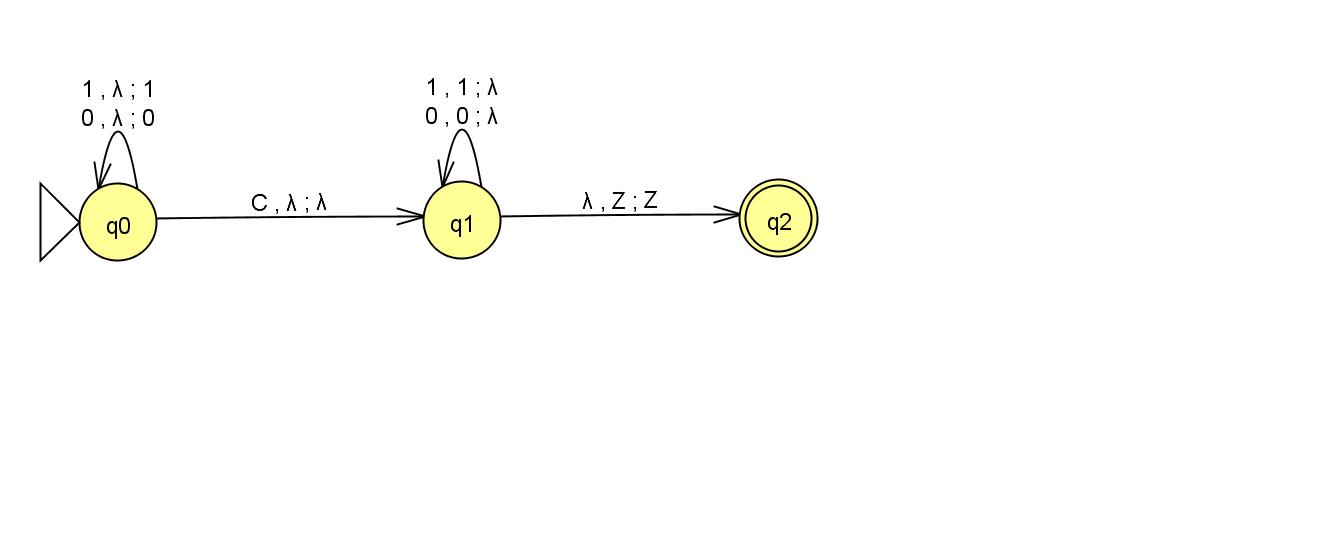


Figura 7. Questão 1, letra D

Fonte: Própria

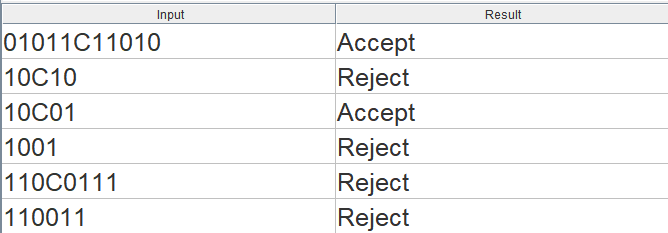


Figura 8. Alguns exemplos de entradas e resultados com base no AP da Questão 1 letra D

Fonte: Própria

**e)**

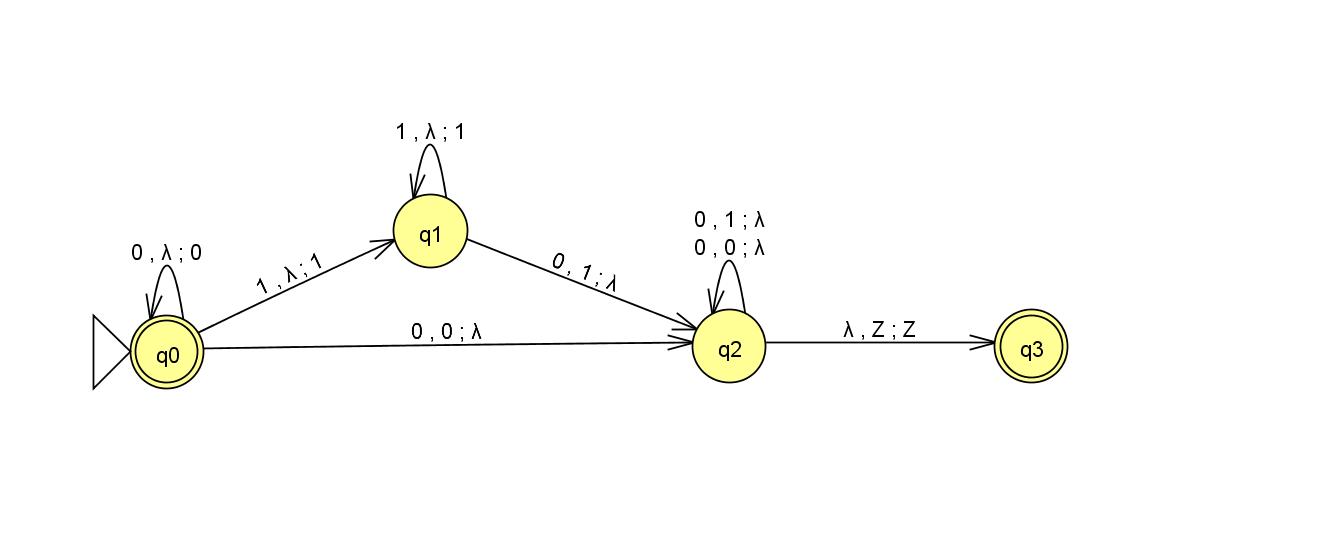


Figura 9. Questão 1, letra E

Fonte: Própria

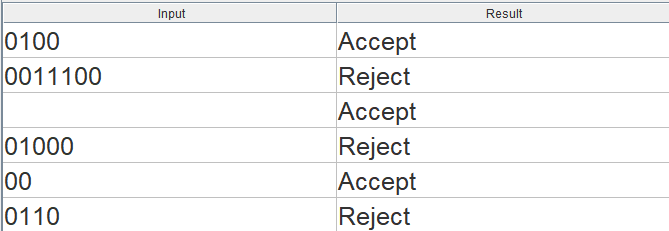


Figura 10. Alguns exemplos de entradas e resultados com base no AP da Questão 1 letra E

Fonte: Própria

**f)**

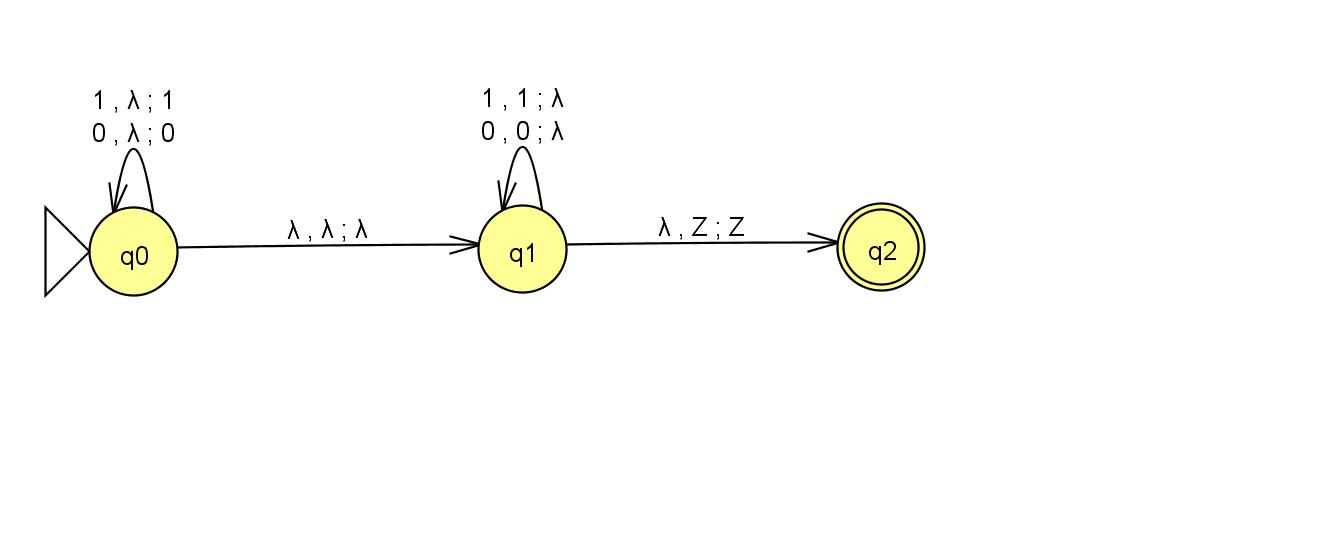


Figura 11. Questão 1, letra F

Fonte: Própria

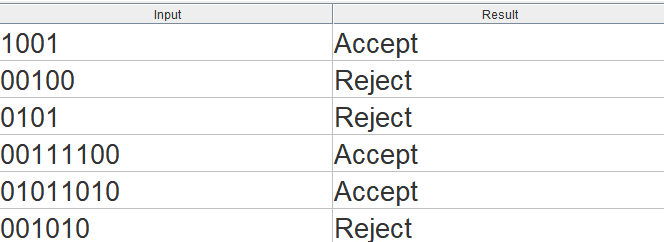


Figura 12. Alguns exemplos de entradas e resultados com base no AP da Questão 1 letra F

Fonte: Própria

**g)**

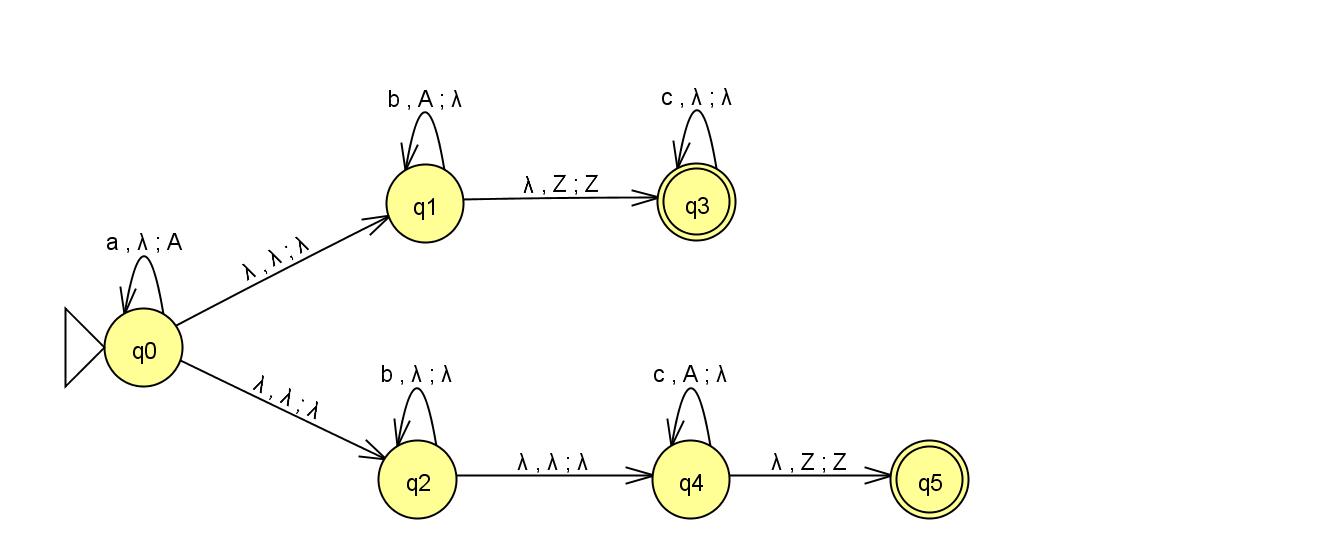


Figura 13. Questão 1, letra G

Fonte: Própria

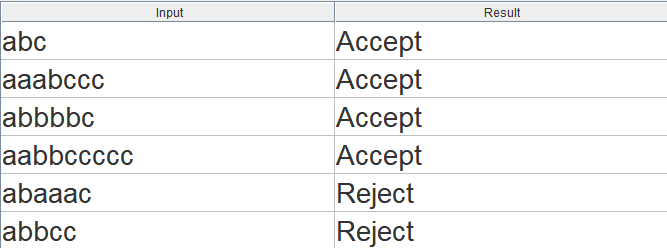


Figura 14. Alguns exemplos de entradas e resultados com base no AP da Questão 1 letra G

Fonte: Própria

**2.**

**a)**

**S → 0 | 5 | N5 | N0**

**N → 1 | 2 | 3 | ... | 9 | NN | N0**

L(G) = {w | w = todo número múltiplo de 5 incluindo o 0}

**b)**

**S → 0S1 | 1S0 | 01 | 10**

L(G) = {w є {0,1} **|** w possui os extremos, até o centro, opostos, por exemplo: se o primeiro for 1 o último tem que ser 0, se o segundo for 0 o penúltimo tem que ser 1 até chegar ao centro}

Outra solução que encontrei foi:

L(G) = {w є {0,1} **|** w tem tamanho par e a mesma quantidade de 0s e 1s e que se lida de trás pra frente trocando os 0s por 1s e vice versa terá que resultar na entrada original}

**S → aBca | aca**

**B → b | bB**

L(G) = {aca | n ≥ 0}

**d)**

**S → 0A | 1B**

**A → 0A | 0**

**B → 1B | 1**

L(G) = {(≥ 2}

**3.**

As Gramáticas Livres de Contexto (GLC) são gramáticas onde as regras de produção são definidas de forma mais livre do que nas gramáticas regulares. A GLC é um tipo mais complexo de geradores de linguagem, as quais materializam um completo entendimento do procedimento de construção das palavras pertencentes à linguagem. Elas reconhecem todas as linguagens regulares e mais outras. A GLC é aplicada em compiladores e conversores de documentos.

**4.**

**L(G) = { ≥ 0}**

S → λ | AB

A → 0A11 **|** λ

B → 0B **|** λ

**5.**

**a)**

O estudo das Linguagens Livres de Contexto tem uma fundamental importância para a ciência da computação, pois aborda um conjunto mais amplo de linguagens e tem grande importância dentro do estudo das Linguagens Formais, pois através delas pode ser descrita a maior parte das construções sintáticas das linguagens de programação. Com ela podemos tratar questões como as dos parênteses balanceados construções de blocos e estruturas, entre outras formalidades de linguagens de computação conhecidas. Seu estudo é fundamental para o desenvolvimento de compiladores, pois a partir destas são desenvolvidos os analisadores sintáticos, parte de um compilador.

**b)**

A aplicação dessas linguagens para os autômatos é usada em autômatos de pilha que se trata de um reconhecedor da linguagem, que consiste basicamente em um autômato que utiliza uma pilha, que serve como memória adicional. E aplicando a sua gramática, se tornam tão importantes porque podem descrever estruturas recursivas o que as tornam necessárias para uma variedade de aplicações, como por exemplo, a formação de um analisador sintático, componente de um compilador.

**c)**

|  |  |
| --- | --- |
| **LINGUAGENS LIVRES DE CONTEXTO** | **LINGUAGENS REGULARES** |
| **DIFERENÇAS** | |
| Possuem regras gerais | Possuem regras restritas |
| Pode expressar linguagens livres de contexto e também as regulares | Expressa apenas as linguagens regulares |
| Ao expressar em um autômato pode contar com o auxílio de uma memória para determinar a frequência de um elemento | Não conta com ajuda de memória |
| **SEMELHANÇAS** | |
| Podem expressar linguagens finitas | Podem expressar linguagens finitas |
| São fechadas em determinadas operações | São fechadas em determinadas operações |