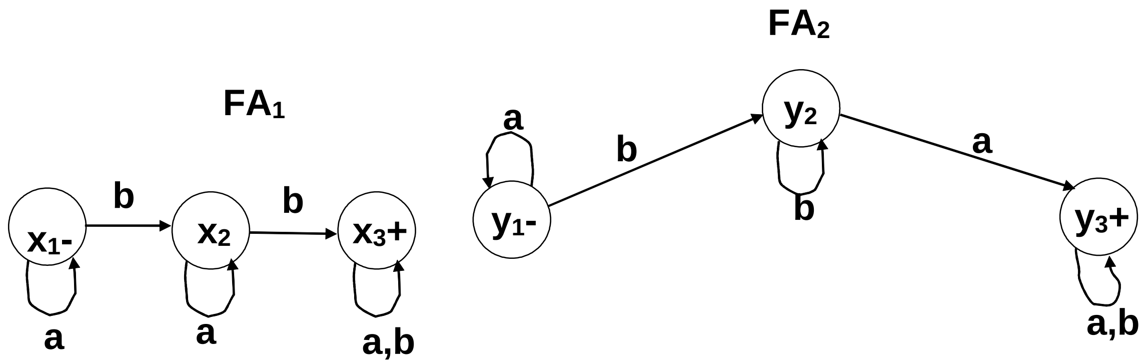
| | |
|---|---|
| **Prova P1** | **Disciplina: Teoria da Computação** |
| | **Professor: Ney Laert Vilar Calazans** |
| Aluno: | 09/outubro/2019 |

1. **(2 pontos) Forneça uma definição recursiva da linguagem BBB, definida sobre o alfabeto Σ={a,b} e que possui todas as palavras que contêm a sub-cadeia "*bbb*" e somente tais palavras. Lembre-se que existem três tipos de regras em uma definição recursiva: (1) regras de objetos fundamentais; (2) regras recursivas; e (3) a regra de completude sobre os outros dois tipos de regras.**

2. **(4 pontos) Aplicando os passos de algoritmo construtivo adequado proveniente da prova do Teorema de Kleene, gere o autômato finito união (FA1+FA2) dos dois autômatos finitos FA1 e FA2 dados abaixo. Ambos autômatos reconhecem palavras sobre o alfabeto Σ={a,b}. Após definir o autômato resultante, responda às seguintes questões, sobre ele:**
   a) **Qual a linguagem aceita por este autômato? Explique a estrutura da linguagem.**
   b) **A mesma linguagem poderia ser definida por um autômato com menos estados? Explique sua resposta.**




3. **(4 pontos) Elabore e desenhe uma máquina de Turing (TM) que aceita a linguagem definida sobre o alfabeto Σ={a,b} tal que todas as palavras da linguagem possuem pelo menos duas instâncias da sub-cadeia "*bab*", sendo que uma destas ocorre necessariamente no final da palavra. Após elaborar sua máquina, responda às seguintes questões sobre ela:**
   a) **Sua máquina altera a fita da máquina de alguma forma ao longo do processamento?**
   b) **A linguagem aceita por esta TM é regular ou não? Comente a resposta, fornecendo evidências que ela faz sentido no contexto de definição de linguagens formais.**
   c) **Existe alguma relação entre as questões dos itens a) e b) acima? Comente sua resposta.**

| Prova P1 | Disciplina: Teoria da Computação |
|---|---|
| | Professor: Ney Laert Vilar Calazans |

Aluno: **Gabarito** 09/outubro/2019

1. **(2 pontos) Forneça uma definição recursiva da linguagem BBB, definida sobre o alfabeto Σ={a,b} e que possui todas as palavras que contêm a sub-cadeia "*bbb*" e somente tais palavras. Lembre-se que existem três tipos de regras em uma definição recursiva: (1) regras de objetos fundamentais; (2) regras recursivas; e (3) a regra de completude sobre os outros dois tipos de regras.**

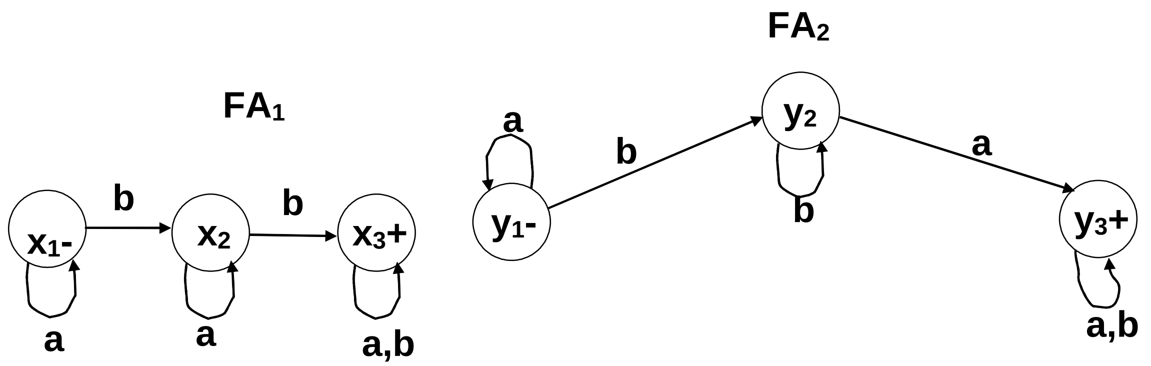
   **Solução:**

   **Regra 1: bbb é elemento de BBB;**

   **Regras 2: Se x é elemento de BBB, então xa, ax, xb e bx são elementos de BBB;**

   **Regra 3: Os únicos elementos da linguagem BBB são aqueles que podem ser produzidos a partir unicamente da aplicação das regras 1 e 2.**

2. **(4 pontos) Aplicando os passos de algoritmo construtivo adequado proveniente da prova do Teorema de Kleene, gere o autômato finito união (FA1+FA2) dos dois autômatos finitos FA1 e FA2 dados abaixo. Ambos autômatos reconhecem palavras sobre o alfabeto Σ={a,b}. Após definir o autômato resultante, responda às seguintes questões, sobre ele:**
   a) **Qual a linguagem aceita por este autômato? Explique a estrutura da linguagem.**
   b) **A mesma linguagem poderia ser definida por um autômato com menos estados? Explique sua resposta.**

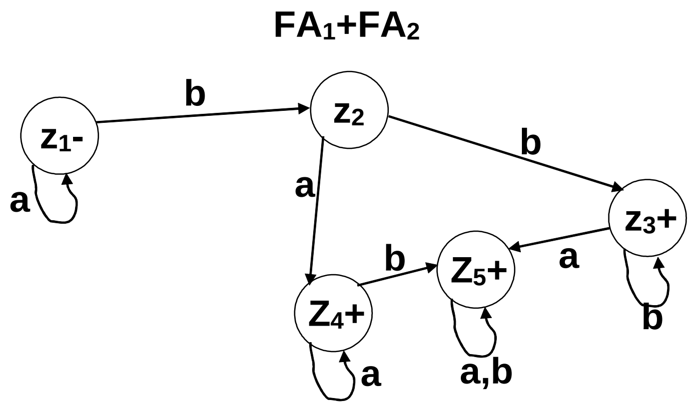



**Solução: Gerando tabelas de transição para os Autômatos FA1 e FA2, tem-se:**

| FA1 | a | B | FA2 | a | b |
|---|---|---|---|---|---|
| x1- | x1- | x2 | y1- | y1- | y2 |
| x2 | x2 | x3+ | y2 | y3+ | y2 |
| x3+ | x3+ | x3+ | +y3 | y3+ | y3+ |

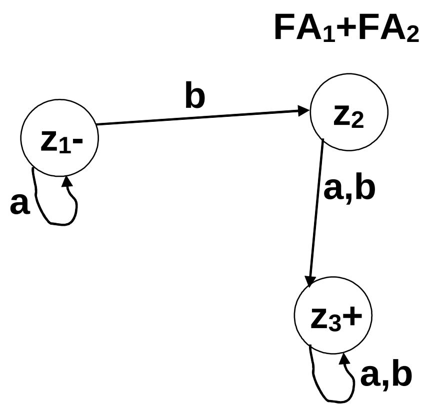
**A partir destas tabelas pode-se gerar a tabela do autômato união FA1+FA2, montando a sua tabela, segundo o método correto desenvolvido como parte da prova do Teorema de Kleene. O método diz que o estado inicial do autômato união é a composição dos dois estados iniciais dos autômatos originais, e que os demais estados correspondem da união correspondem à composição de pares de estados atingíveis pelo consumo de cada letra do alfabeto comum. Os estados finais do autômato união são as composições que contenha um estado final de pelo menos uma das máquinas. Isto fornece como resultado o autômato abaixo:**

| FA1+FA2 | a | b |
|---|---|---|
| -z1 = (-x1 ou -y1) | z1- | z2 = (x2 ou y2) |
| z2 = (x2 ou y2) | z4+ (x2 ou y3+) | z3+ (x3+ ou y2) |
| z3+ = (x3+ ou y2) | z5+ (x3+ ou y3+) | z3+ |
| z4+ = (x2 ou y3+) | z4+ | z5+ |
| z5+ = (x3+ ou y3+) | z5+ | z5+ |




a) A linguagem aceita por FA1+FA2 é a das cadeias que possuem pelo menos 2 b´s ou pelo menos 1 b seguido de 1 a. Expresso como uma ER, tem-se FA1+FA2 = a*b(b+a)(a+b)*.

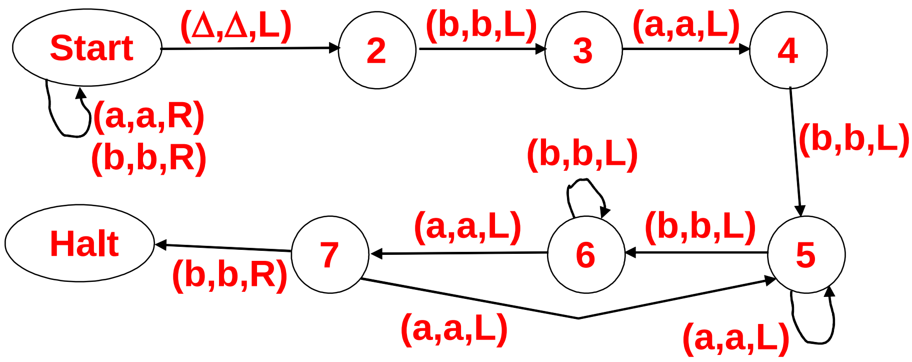
b) Sim, seria possível implementar FA1+FA2 assim (note-se que z3+, z4+ e z5+ são todos estados finais e formam uma classe de equivalência):




3. (4 pontos) Elabore e desenhe uma máquina de Turing (TM) que aceita a linguagem definida sobre o alfabeto Σ={a,b} tal que todas as palavras da linguagem possuem pelo menos duas instâncias da sub-cadeia "*bab*", sendo que uma destas ocorre necessariamente no final da palavra. Após elaborar sua máquina, responda às seguintes questões sobre ela:
   a) Sua máquina altera a fita da máquina de alguma forma ao longo do processamento?
   b) A linguagem aceita por esta TM é regular ou não? Comente a resposta, fornecendo evidências que ela faz sentido no contexto de definição de linguagens formais.
   c) Existe alguma relação entre as questões dos itens a) e b) acima? Comente sua resposta.

Solução: Uma TM que aceita a linguagem em questão é dada abaixo.

## TM que aceita linguagem de palavras com duas instâncias de *bab*, sendo a segunda no final da palavra




**Observações sobre esta TM:**

a) **A TM não altera a fita (algumas soluções corretas podem alterá-la, mas isto não é estritamente necessário para esta linguagem). Ou seja, uma TM sem capacidade de escrita na fita pode implementar sua funcionalidade.**

b) **Sim, é uma linguagem regular, que pode ser expressa por TM = (a+b)*bab(a+b)*bab.**

c) **Sim. Informalmente dito, uma TM com uma fita que não pode ser escrita (uma *Read-Only TM*) não consegue "guardar" fatos intermediários sobre o processamento das palavras na fita, baseando-se apenas na sua estrutura de estados dela para reconhecer palavras, uma característica de autômatos finitos.**