

## Usamos uma variante da Máquina de Turing onde é permitido o não deslocamento da fita

Como mostrado no livro, essa variante é equivalente a uma Máquina de Turing, bastando apenas adicionar dois estados extras, um que move o cabeçote para a esquerda e outra para a direita, em sequência.

**Conforme conversado por e-mail, optamos por utilizar vídeos ao invés de screenshots para apresentar os testes realizados. Os mesmos se encontram no arquivo compactado da entrega.**

### Questão 1.a

#### Ideia básica

A ideia básica é comparar posições equivalentes das duas strings, caso encontre os caracteres sejam diferentes, rejeite

Outro dois pontos são quando: 1 - a primeira string é maior que a segunda e 2 - a segunda string é maior que a primeira. Se a primeira string é maior que a segunda, o estado q1 vai detectar essa diferença e vai rejeitar. Se a segunda for maior, o estado q1 também vai detectar essa diferença e rejeitará a entrada.

#### Algoritmo de alto nível

1. Se a máquina ler uma #, vá para o passo 5-final. O cabeçote da máquina está apontado para a extrema esquerda da fita, enquanto ler o símbolo x faça o cabeçote ir para a direita, se a máquina encontrar o símbolo "a", vá para o passo 2. Se encontrar um símbolo b, vá para o passo 3. Se encontrar um símbolo c, vá para o passo 4.

2. Enquanto a máquina ler símbolos não marcados(a,b ou c), faça o cabeçote ir para direita, quando encontrar o símbolo "#", faça o cabeçote ir para direita enquanto encontrar símbolos marcados. Se a máquina ler b ou c, rejeite. Se a máquina encontrar uma "a", marque este "a" e mova o cabeçote para a esquerda e enquanto ela ler símbolos a,b,c e x, faça o cabeçote ir para a esquerda, quando encontrar o símbolo #, mova o cabeçote e continue procurando símbolos a,b e c, movendo o cabeçote sempre para a esquerda. Se a máquina encontrar um símbolo x, faça o cabeçote ir para a direita e volte para o passo 1.

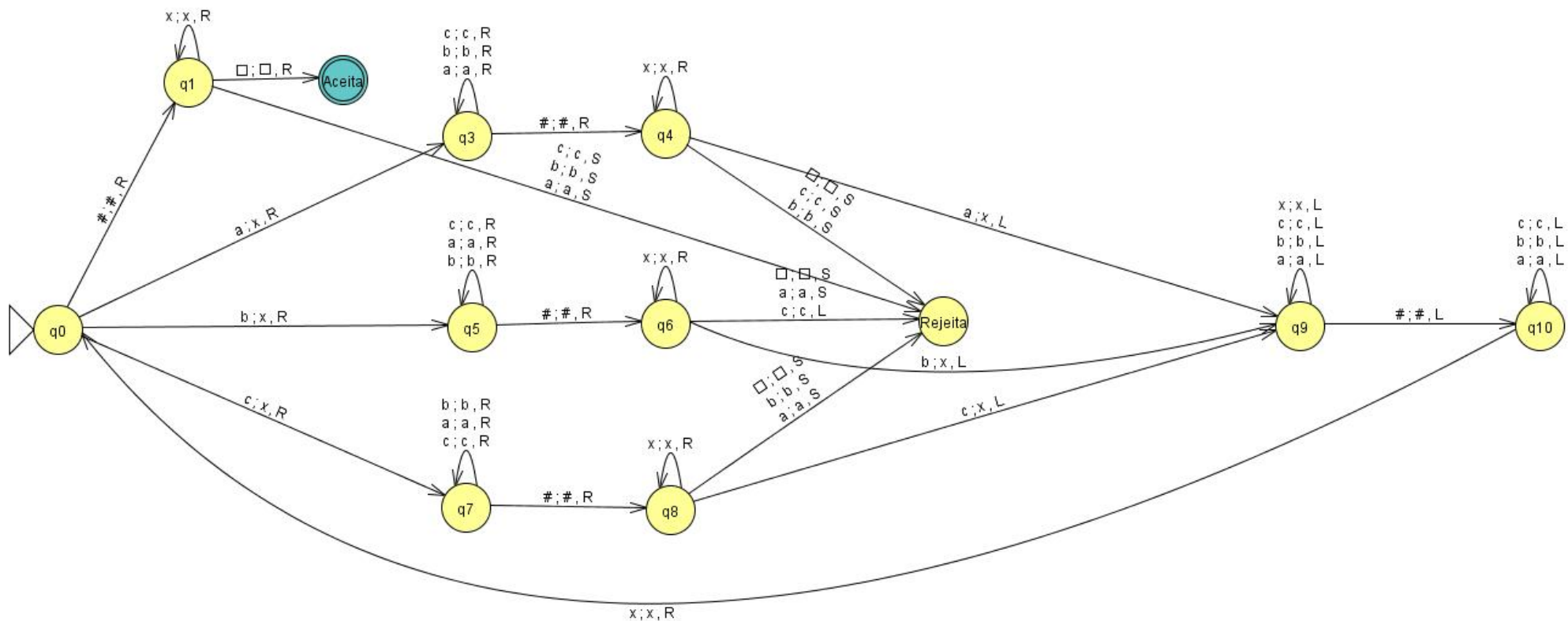
3. Enquanto a máquina ler símbolos não marcados(a,b ou c), faça o cabeçote ir para direita, quando encontrar o símbolo "#", faça o cabeçote ir para direita e enquanto encontrar símbolos marcados. Se a máquina ler a ou c, rejeite. Se a máquina encontrar uma b, marque este b e mova o cabeçote para a esquerda e enquanto ela ler símbolos a,b,c e x, faça o cabeçote ir para a esquerda, quando encontrar o símbolo #, mova o cabeçote e continue procurando símbolos a,b e c, movendo o cabeçote sempre para a esquerda. Se a máquina encontrar um símbolo x, faça o cabeçote ir para a direita e volte para o passo 1.

4. Enquanto a máquina ler símbolos não marcados(a,b ou c), faça o cabeçote ir para direita, quando encontrar o símbolo "#", faça o cabeçote ir para

direita enquanto encontrar símbolos marcados. Se a máquina ler a ou b, rejeite. Se a máquina encontrar uma b, marque este b e mova o cabeçote para a esquerda e enquanto ela ler símbolos a,b,c e x, faça o cabeçote ir para a esquerda, quando encontrar o símbolo #, mova o cabeçote e continue procurando símbolos a,b e c, movendo o cabeçote sempre para a esquerda. Se a máquina encontrar um símbolo x, faça o cabeçote ir para a direita e volte para o passo 1.

5 - Final. Enquanto a máquina ler x, mova o cabeçote para a direita, caso leia o símbolo algum símbolo que não x, rejeite. Se encontrar o símbolo em branco, aceite.

### **Diagrama da máquina**



### Questão 1.b

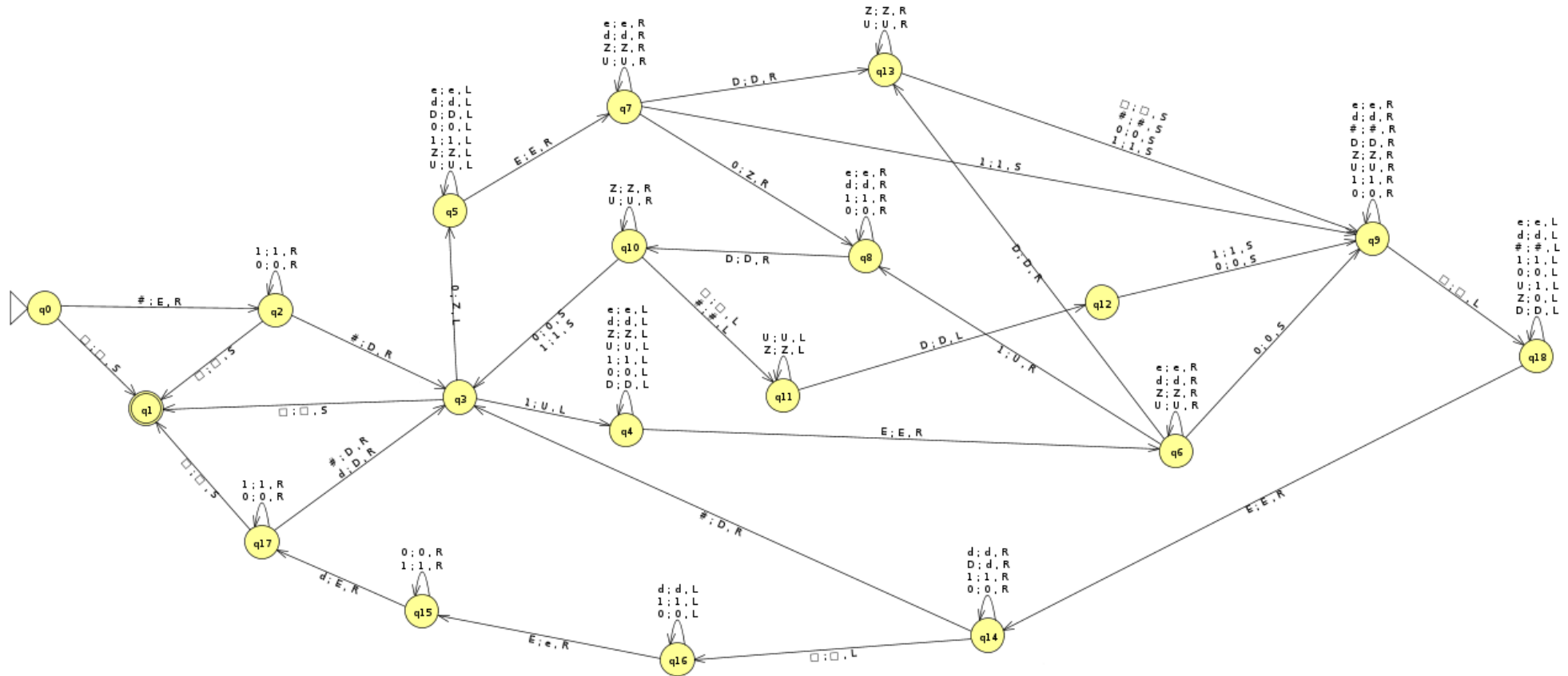
Máquina de Turing com fita única que computa a linguagem  $L = \{\#x_1\#x_2\#\dots\#x_n \mid x_i \in \{0,1\}^* \text{ e } x_i \neq x_j \text{ para cada } i \neq j\}$

**Algoritmo de alto nível que descreve o funcionamento da máquina:**

$M$  = "Sobre a entrada  $x$ :

1. Se a entrada for vazia, *aceite*. Se o primeiro símbolo for um #, marque ele e prossiga para o próximo passo. Senão, *rejeite*.
2. Procure o próximo # e o marque. Se nenhum # for encontrado, *aceite*.
3. Através de um zigue-zague compare as duas cadeias à direita dos #s encontrados anteriormente. Se elas forem diferentes, prossiga para o próximo passo. Senão, *rejeite*.
4. Mova a marca mais à direita das duas para o próximo símbolo # à direita e vá para o próximo passo. Se nenhum símbolo # for encontrado antes de um símbolo em branco, mova a marca mais à esquerda para o próximo # à sua direita, a marca mais à direita para o # depois desse e vá para o próximo passo. Se nenhum # estiver disponível para a marca mais à direita, todas as cadeias foram comparadas, *aceite*.
5. Prossiga para o passo 3."

## Codificação da máquina



## Questão 2.a

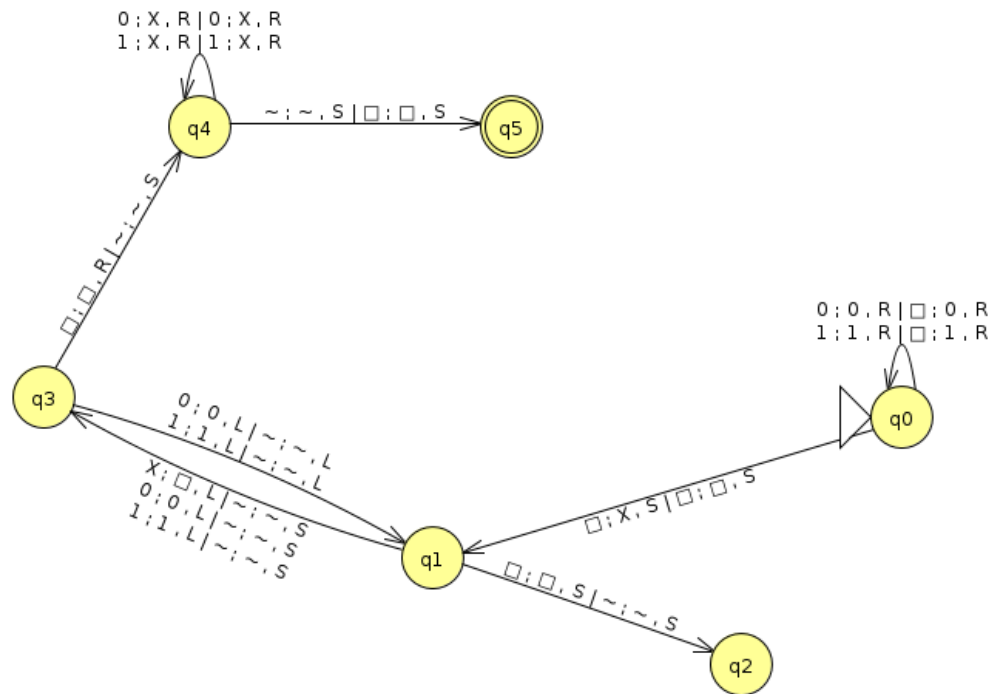
Máquina de Turing Multifitas que computa a linguagem  $L = \{ww \mid w \in \{0,1\}^*\}$

Algoritmo de alto nível que descreve o funcionamento da máquina:

$M$  = "Com duas fitas e sobre a entrada  $x$  que inicialmente está na fita 1:

1. Copie cada símbolo da fita 1 para a fita 2.
2. Encontre o meio da palavra na fita 2 e volte ao começo da fita 1. Assim cada fita tratará de uma metade da palavra.
3. Ao realizar o passo anterior, verifique a quantidade de símbolos que a fita 1 possui. Se esse número for par, prossiga para o próximo passo. Senão, *rejeite*.
4. Compare o símbolo que está em uma metade na fita 1 com o símbolo que está na outra metade na fita 2. Se eles forem iguais, prossiga para o próximo passo. Senão, *rejeite*.
5. Repita o último passo até não houver mais símbolos nas metades. Se não houver mais símbolos nas metades, *aceite*."

### Codificação da máquina



### Questão 2.b

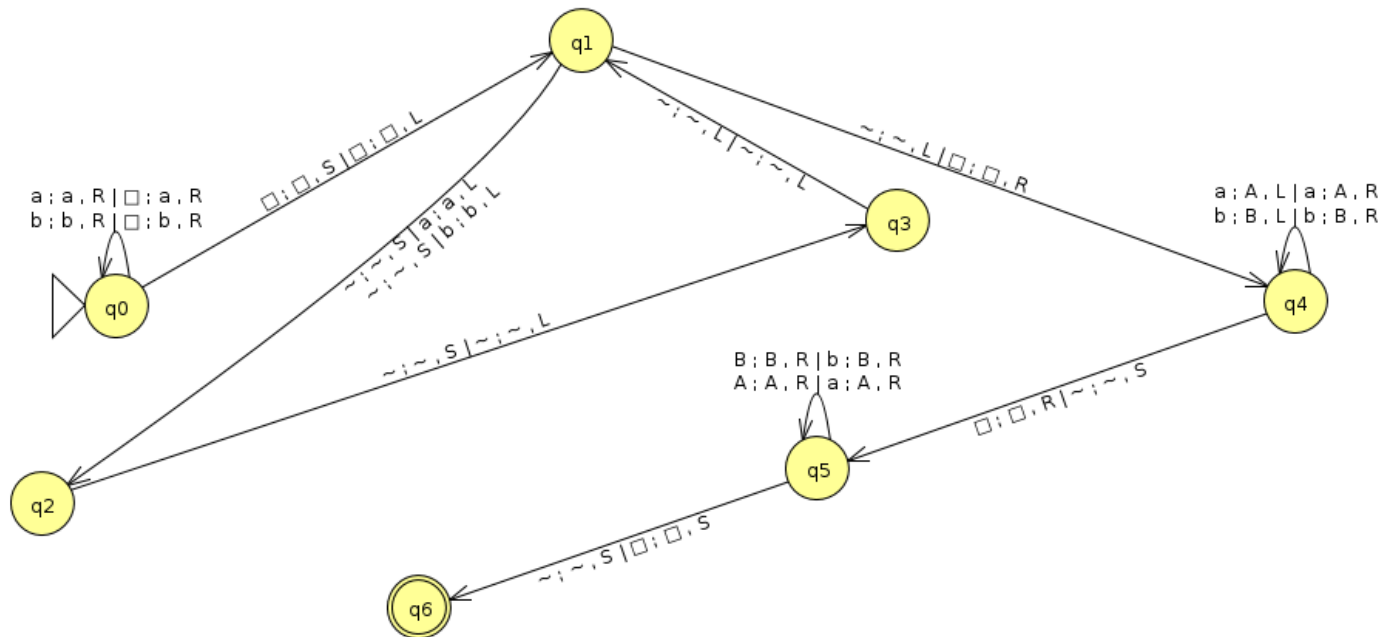
Máquina de Turing Multifitas que computa a linguagem  $L = \{ww^Rw \mid w \in \{a,b\}^*\}$

Algoritmo de alto nível que descreve o funcionamento da máquina:

$M =$  "Com duas fitas e sobre a entrada  $x$  que inicialmente está na fita 1:

1. Copie cada símbolo da fita 1 para a fita 2.
2. Vá para o começo da terceira metade da palavra da fita 1 e volte ao começo da fita 2.
3. Ao realizar o passo anterior, verifique a quantidade de símbolos que a fita 1 possui. Se esse número for múltiplo de 3, prossiga para o próximo passo. Senão, *rejeite*.
4. Compare os símbolos (dois terços da entrada) que estão nas duas fitas enquanto houver item na fita 1. Se eles forem iguais, prossiga para o próximo passo. Senão, *rejeite*.
5. Vá para o começo da primeira terça parte da entrada da fita 1 e no começo da terceira parte da entrada da fita 2.
6. Compare os símbolos que estão nas duas fitas enquanto houver item na fita 2. Se eles forem iguais, *aceite*. Senão, *rejeite*."

## Codificação da máquina



## Questão 3.a

Para esse problema desenvolvemos 2 Máquinas de Turing: a principal e o bloco básico. O papel do bloco básico(o qual chamamos de block1 e com label "VoltaFita") tem o papel de voltar a fita

## Funcionamento da máquina

A máquina funciona sempre cortando o número de 0's pela metade, seguindo essa lógica, se há um número de 0's na entrada que seja uma potência de 2, então em algum momento sobrarão apenas um 0 não marcado(não cortado). Quando isso ocorrer, a máquina estará em seu estado inicial com um único 0, então ela marcará o 0 e segue para o estado q2, ali entra em loop procurando por 0's marcados e 1's, como não irá encontrar, ela vai achar um símbolo vazio e irá para o estado Aceite.

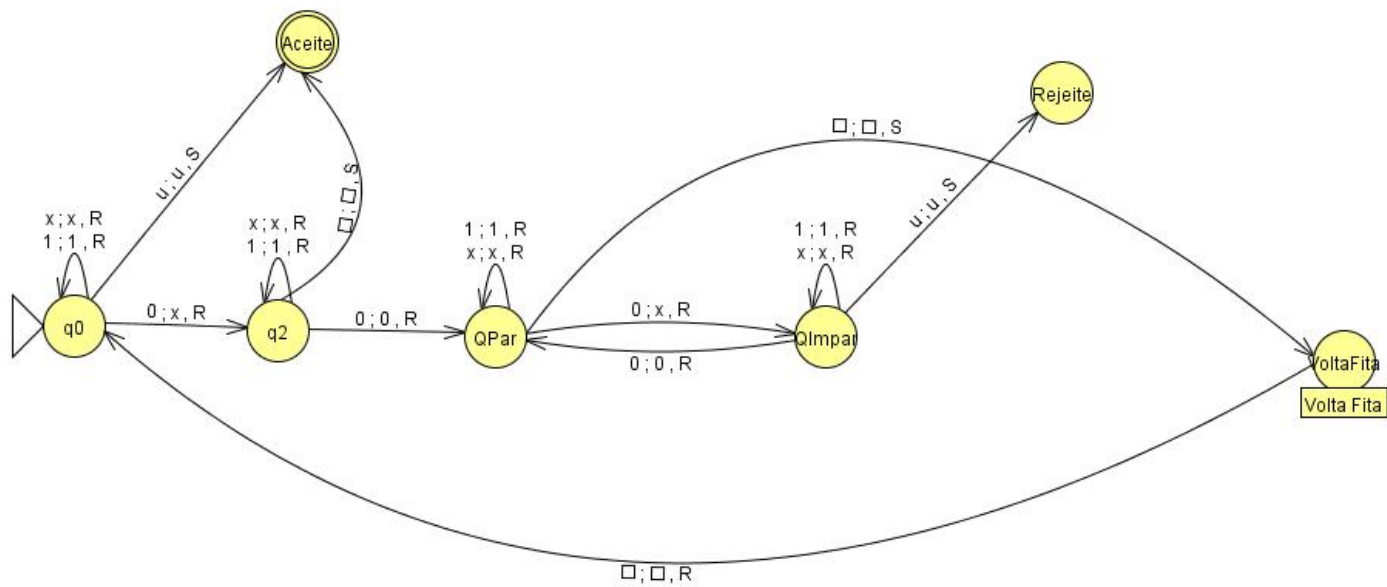
Outro caso é quando a entrada contém um número N ímpar de 0's (N diferente de 1), o qual o estado qÍmpar detecta, se ele não avistar outro 0(i.e encontrar o símbolo em branco), então vai para o estado Rejeite.

O último caso é o número de 0's ser par e não ser uma potência de 2. Nesse caso a máquina irá cortando sempre metade dos 0's, em algum momento isso vai resultar em um número ímpar(como todo número é produto de uma série de números primos, se esse número é par e não é uma potência de 2, então ele não pode ser representado por um produto de números 2's, logo vai ter outro primo nesse produto, que é evidentemente ímpar e, quando houver as sucessivas divisões por 2, esse número primo(diferente de 2) ficará evidenciado(exemplo: número  $6 = 2 * 3$ ,  $2*3/2 = 3$ ). Esse caso é coberto pelo anterior.

### Algoritmo de alto nível

Algoritmo de alto nível:

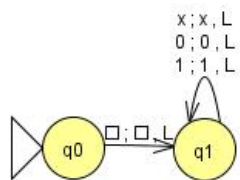
1. Fazer uma varredura da esquerda para a direita na fita, marcando um 0 não, e outro, sim.
2. Se no estágio 1, a fita continha um único 0, aceite.
3. Se no estágio 1, a fita continha mais que um único 0 e o número de 0s era ímpar, então rejeite.
4. Retorne para a extremidade esquerda da fita
5. Vá para o estágio 1



### Blocos utilizados

- Volta fita





### Questão 3.b

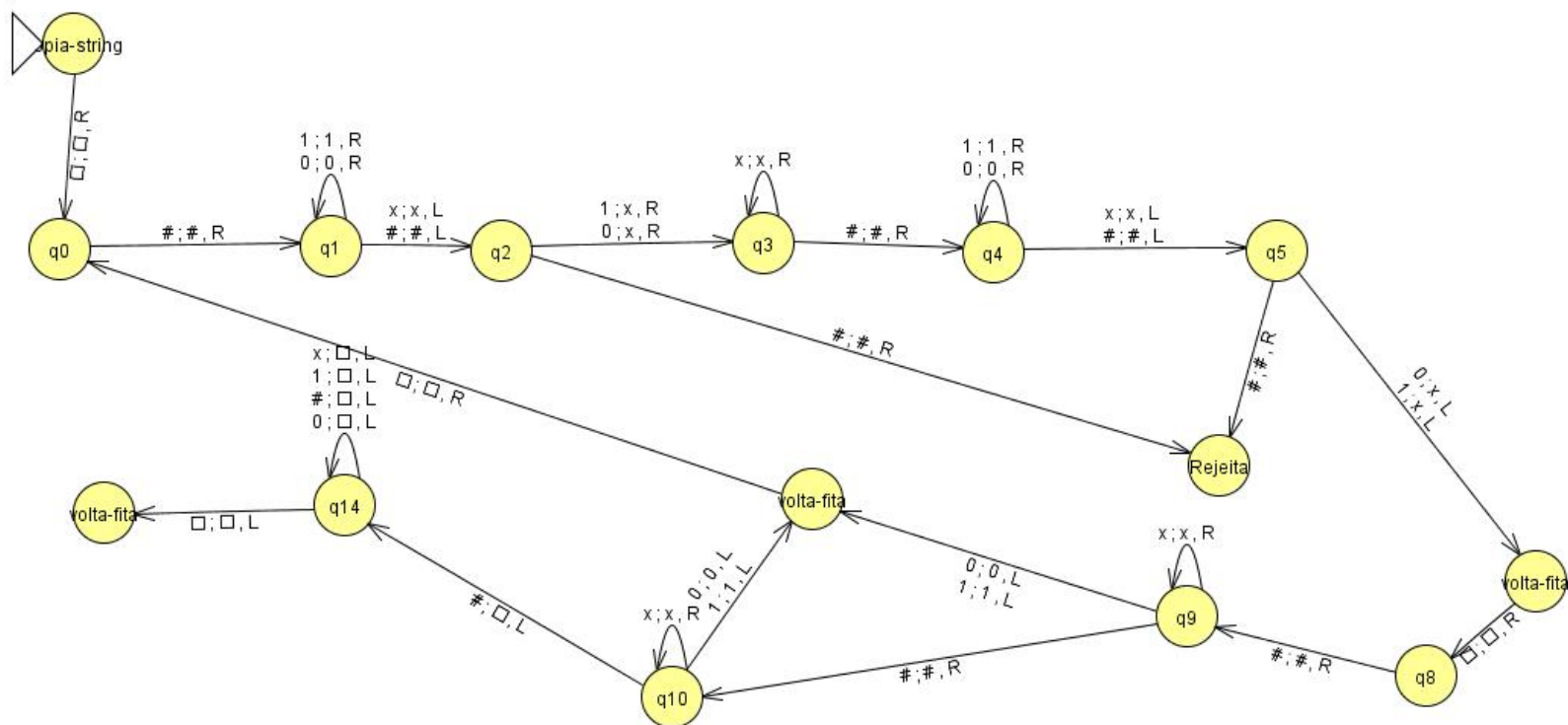
Para essa máquina o input deve seguir essa forma: #operando1#operando2#

O resultado será escrito depois da último "#" e estará invertida

### Blocos utilizados

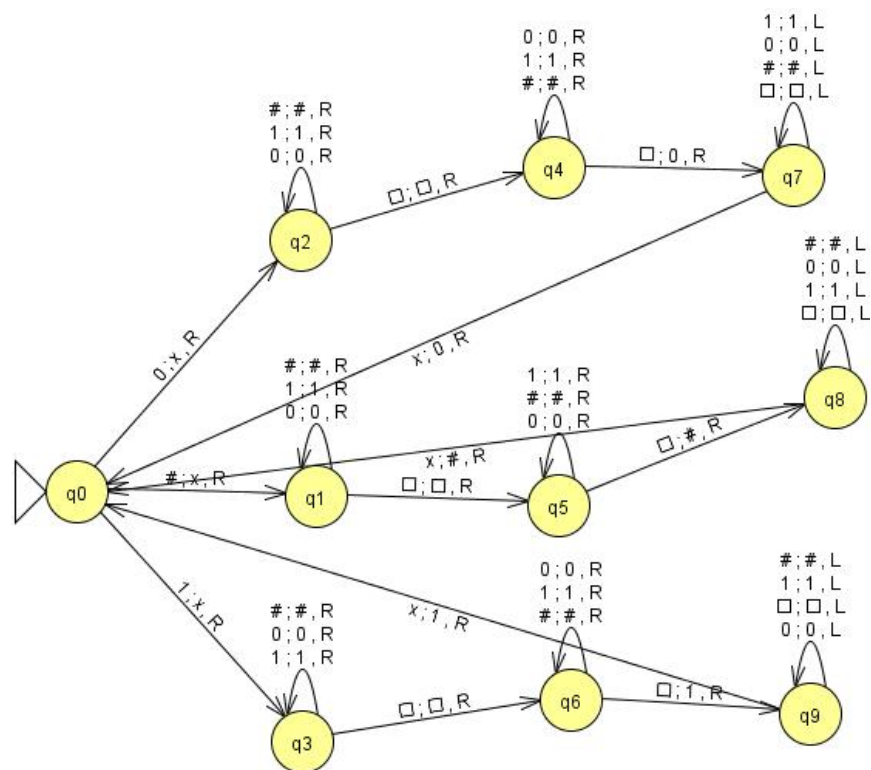
- Verifica tamanho

Verifica se os tamanhos dos dois operandos são iguais, se não forem, rejeita. Ele suja a entrada marcando com x's, então antes de começar a verificação, é necessário copiar a entrada para outro lugar.



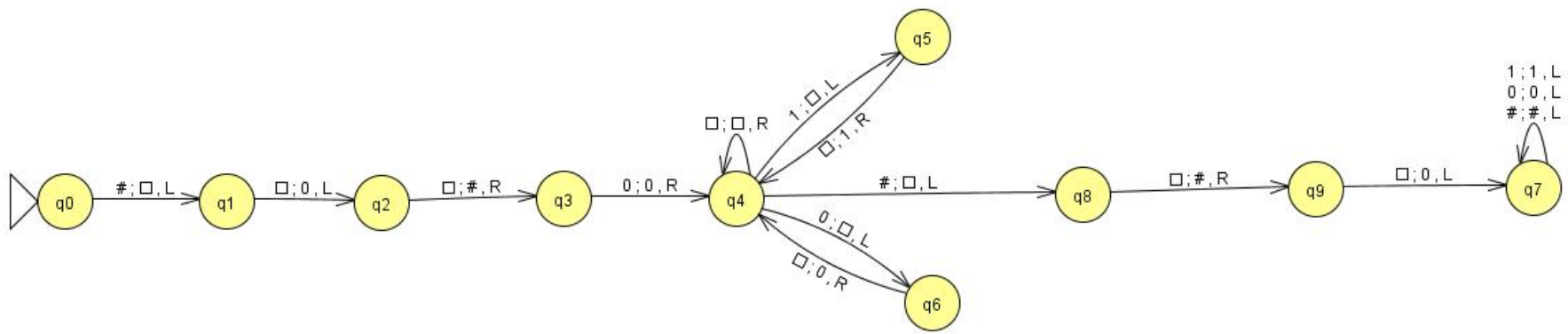
- Copia string

Usado pelo bloco anterior, apenas copia a entrada para outro lugar da fita(i.e deixando um símbolo em branco entre as duas)



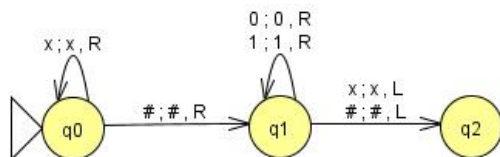
- Resolve overflow

Se o input estiver correto (i.e ter o mesmo número de bits para ambos os operandos), então a máquina utiliza esse bloco para colocar um 0 a mais em cada operando para evitar o overflow.



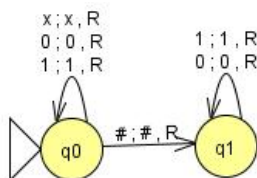
- Procura operando

Esse bloco procura o proximo bit a ser somado



- Procura onde escrever resultado

Esse bloco é usado para voltar a fita para o começo quando ela escreve o resultado parcial da fita



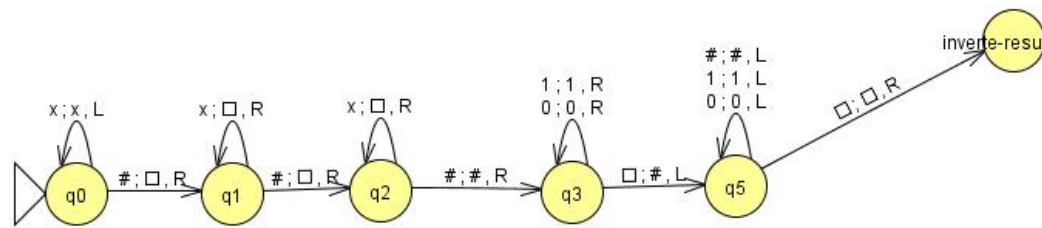
- Volta fita

Esse bloco é responsável de voltar o cabeçote da fita quando ela escreve o resultado na máquina



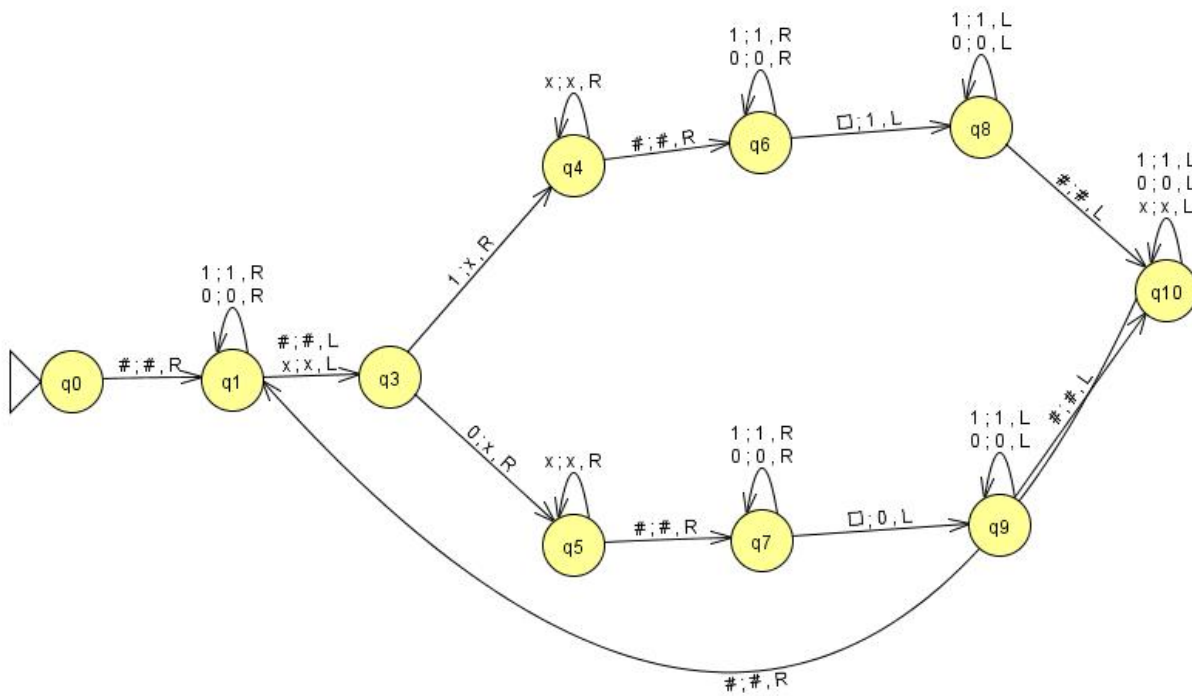
- Limpa-fita

No final da computação a máquina principal terá escrito o resultado depois do último "#", essa máquina vai limpar a fita e inverter o resultado. Nesse bloco há outro bloco, o inverte-fita, já que a máquina produz a soma na ordem inversa.



- Inverte-resultado

Esse bloco vai inverter o resultado da computação. Ele está contido no Limpa-fita.



## Algoritmo de alto nível

1. Procure na fita o primeiro bit não marcado, marque com x. Caso não haja algum bit não marcado, aceite
2. Procure pela próxima #, então procure pelo próximo bit não marcado, marque-o com x
3. Procure pela próxima #, então procure pelo próximo símbolo em branco
4. Caso os bits lidos sejam 0 e 0, escreva 0, volte para o começo da fita. Caso sejam 0 e 1, escreva 1, volte para o começo da fita. Caso sejam 1 e 0, escreva 1, volte para o começo da fita. Caso sejam 1 e 1 vá para o passo 5.
5. Escreva 0 no resultado, volte para o começo da fita. Vá para o passo 6.
6. Procure na fita o primeiro bit não marcado, marque-o com x
7. Procure pela próxima #, então procure pelo próximo bit não marcado, marque-o com x
8. Procure pela próxima #, então procure pelo próximo símbolo em branco
9. Caso os bits lidos sejam 0 e 0, escreva 1, volte para o começo da fita, vá para o passo 1. Caso sejam 0 e 1, escreva 0, volte para o começo da fita, vá para o passo 6. Caso sejam 1 e 0, escreva 0, volte para o começo da fita, vá para o passo 6. Caso sejam 1 e 1, escreva 1 e vá para o passo 6.

## Codificação da máquina completa



