Implementação de um simulador de autômatos com pilha(ACP) que permita ao usuário testar o reconhecimento de diversas cadeias para qualquer ACP de entrada.

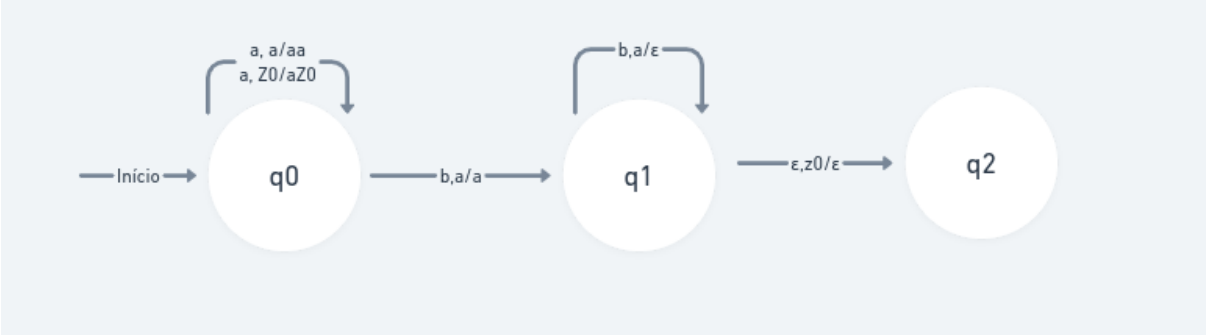
Manual de uso e Documentação

Isabele Cristina Lima Gomes - 2020010186
Ríad Oliveira de Morais - 2020010104
Davi Emanuel Lopes de Lima - 2020010156

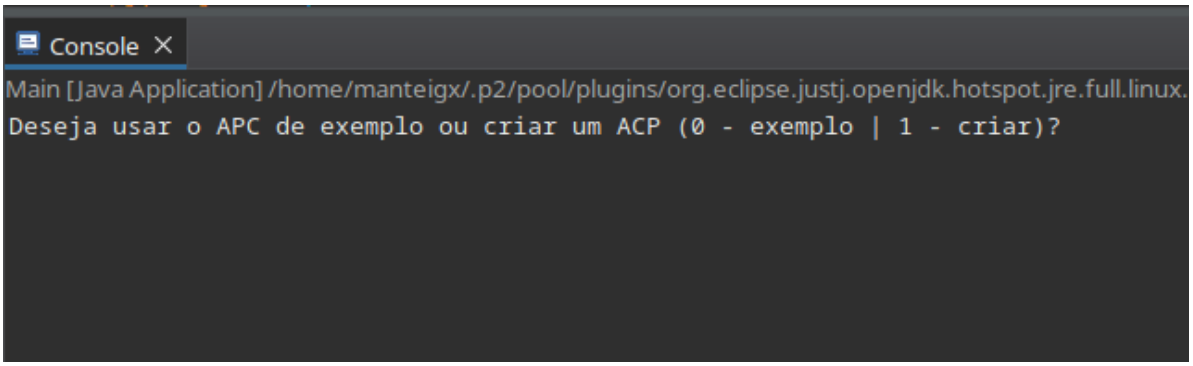
Mossoró-2022

Executado o arquivo Main.java começamos com duas opções para executar o programa:

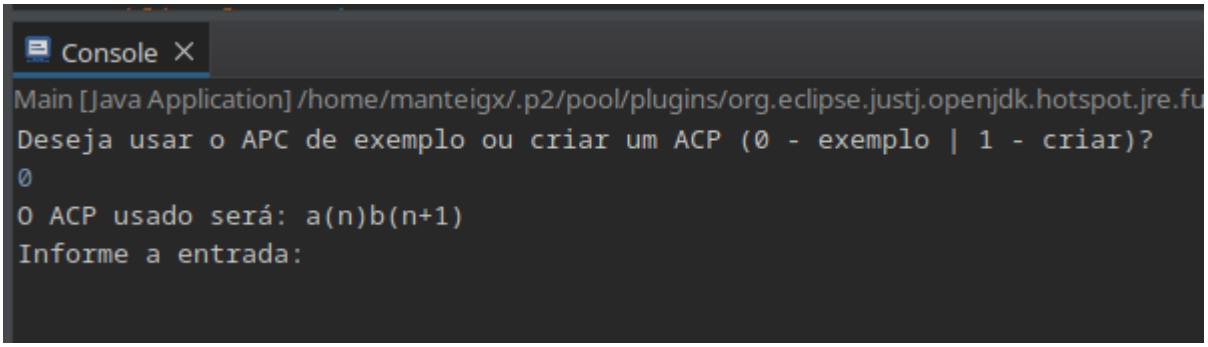
1. Usando o autômato de exemplo pré definido pelo sistema;



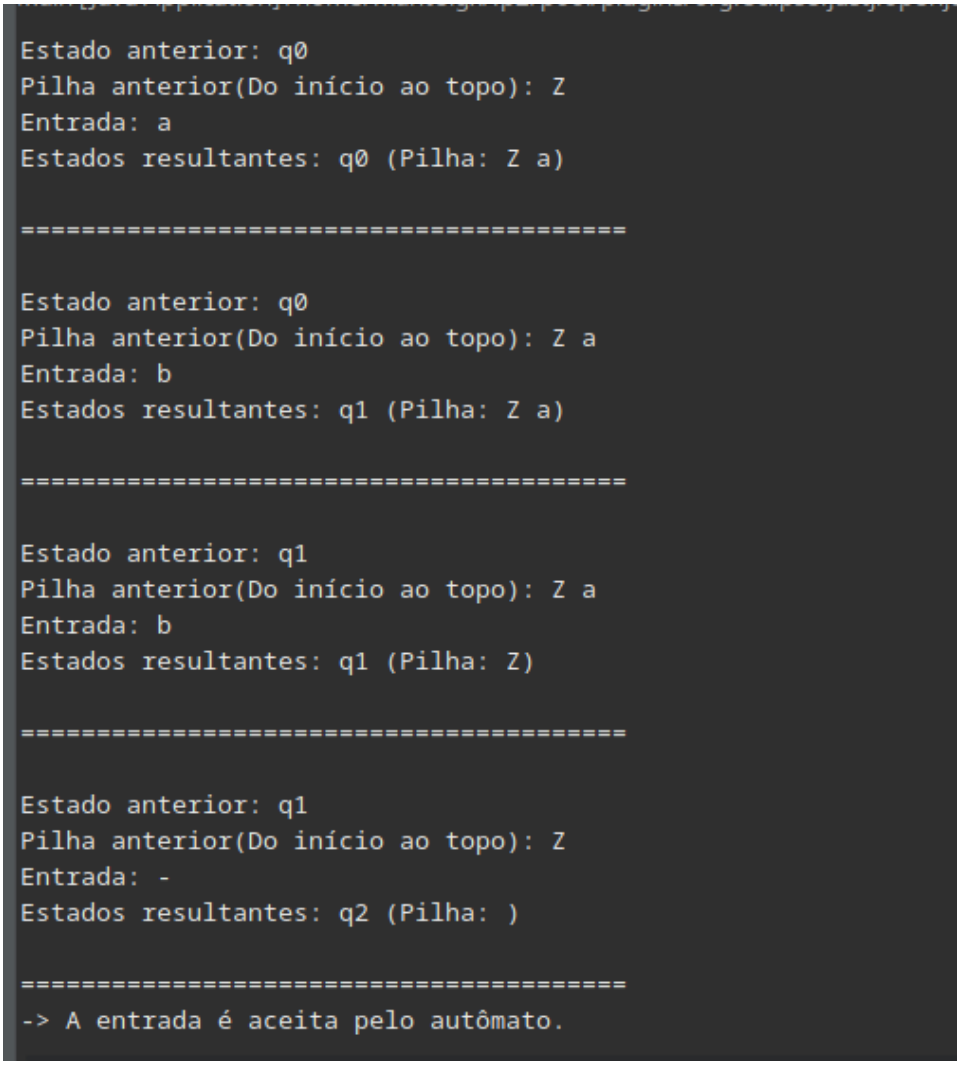
2. Ou criando o seu próprio autômato;



Escolhendo a opção 0 ele informa qual a linguagem aceita pelo autômato e pede uma cadeia de entrada:



Informando a cadeia de entrada abb:



O programa informa o estado em que o autômato está, a situação da pilha, a leitura de entrada e o estado resultante e no final, com a cadeia totalmente consumida, ele informa que a cadeia abb é aceita pelo autômato.

Informando a cadeia de entrada aab

```
Estado anterior: q0
Pilha anterior(Do início ao topo): Z
Entrada: a
Estados resultantes: q0 (Pilha: Z a)

========================================

Estado anterior: q0
Pilha anterior(Do início ao topo): Z a
Entrada: a
Estados resultantes: q0 (Pilha: Z aa)

========================================

Estado anterior: q0
Pilha anterior(Do início ao topo): Z aa
Entrada: b
Estados resultantes: q1 (Pilha: Z aa)

========================================

Estado anterior: q1
Pilha anterior(Do início ao topo): Z aa
Entrada: -
Estados resultantes: Ramificação Morta

========================================
-> A entrada não é aceita pelo autômato.
```

O mesmo acontece com essa cadeia mas no caso a cadeia aab não é aceita.

Escolhendo a opção 1 de criar o próprio autômato temos:

```
Main [Java Application] /home/manteigx/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x8
Deseja usar o APC de exemplo ou criar um ACP (0 - exemplo | 1 - criar)? 1
--------------------------------
Quantos estados terá o ACP: 3

Qual o nome do estado Inicial? q0
Ele é de aceitação (0 - não | 1 - sim)? 0

Qual o nome do estado 2º? q1
Ele é de aceitação (0 - não | 1 - sim)? 0

Qual o nome do estado 3º? q2
Ele é de aceitação (0 - não | 1 - sim)? 1

--------------------------------
```

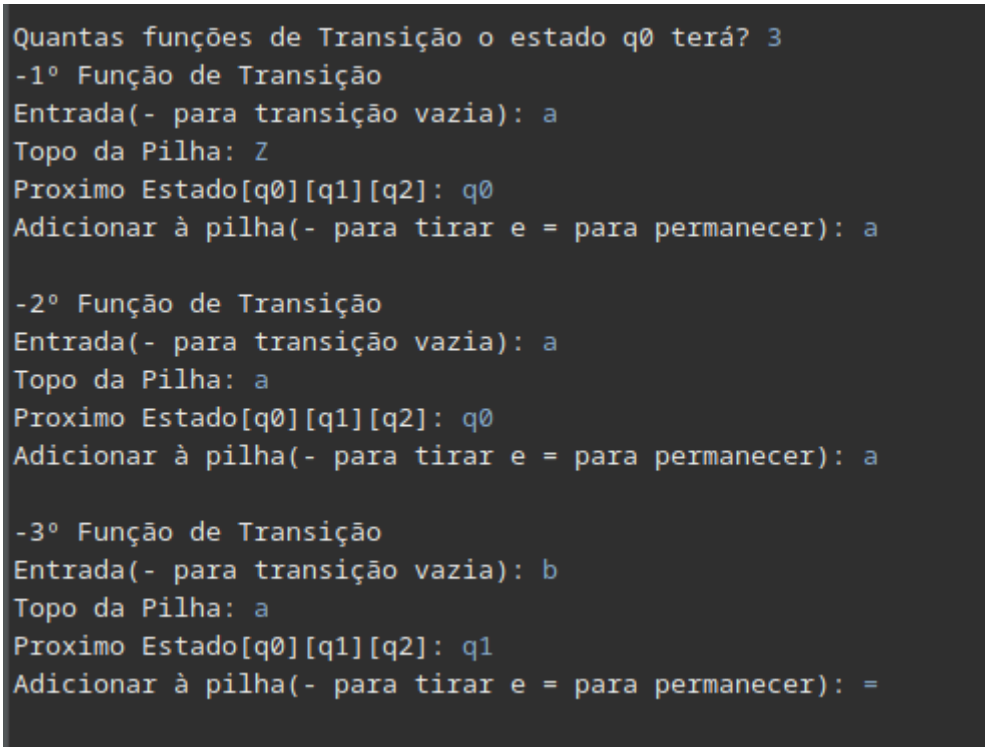
Primeiramente Definimos a quantidade de estados do autômato, Depois o nome de cada um e se ele é estado de aceitação ou não;

```
--------------------------------
Qual nome do ACP? a(n)b(n+1)
Quantos símbolos terá o alfabeto do Automato? 2
====
1: a
2: b
====

Quantos símbolos terá o alfabeto da Pilha? 1
====
1: a
====

Qual será o símbolos de inicio da Pilha? Z
--------------------------------
```

Em seguida digitamos o nome do autômato (algo estético que não influencia o ACP, deixamos a linguagem que ele aceita apenas por praticidade), Quantos símbolos terá o alfabeto do autômato e quais são assim como os da pilha (neste caso apenas empilhamos a). Por fim definimos o símbolo inicial da pilha Z.



Por fim definimos a quantidade e quais as transições de cada estado seguindo o seguinte padrão:

Entrada(- para transição vazia) : qual a entrada que será lida naquela transição, caso a transição seja vazia digitamos -

Topo da pilha : o que deve estar no topo da pilha

Próximo estado: qual o estado que a transição leva, se a transição é um loop basta digitar o estado atual

Adicionar à pilha (- para tirar e = para permanecer) : se a transição deve empilhar digite o que deve ser empilhado, se deve desempilhar digite “ - ” , e se a pilha deve permanecer a mesma digite “ = ”

Transições de q1

```
Quantas funções de Transição o estado q1 terá? 2
-1º Função de Transição
Entrada(- para transição vazia): b
Topo da Pilha: a
Proximo Estado[q0][q1][q2]: q1
Adicionar à pilha(- para tirar e = para permanecer): -

-2º Função de Transição
Entrada(- para transição vazia): -
Topo da Pilha: Z
Proximo Estado[q0][q1][q2]: q2
Adicionar à pilha(- para tirar e = para permanecer): -
```

Transições de q0

```
Quantas funções de Transição o estado q2 terá? 0
```

Testando a entrada abb:

```
Estado anterior: q0
Pilha anterior(Do início ao topo): Z
Entrada: a
Estados resultantes: q0 (Pilha: Z a)

========================================

Estado anterior: q0
Pilha anterior(Do início ao topo): Z a
Entrada: b
Estados resultantes: q1 (Pilha: Z a)

========================================

Estado anterior: q1
Pilha anterior(Do início ao topo): Z a
Entrada: b
Estados resultantes: q1 (Pilha: Z)

========================================

Estado anterior: q1
Pilha anterior(Do início ao topo): Z
Entrada: -
Estados resultantes: q2 (Pilha: )

========================================
-> A entrada é aceita pelo autômato.
```

Testando a entrada aab:

```
Estado anterior: q0
Pilha anterior(Do início ao topo): Z
Entrada: a
Estados resultantes: q0 (Pilha: Z a)

=======================================

Estado anterior: q0
Pilha anterior(Do início ao topo): Z a
Entrada: a
Estados resultantes: q0 (Pilha: Z aa)

=======================================

Estado anterior: q0
Pilha anterior(Do início ao topo): Z aa
Entrada: b
Estados resultantes: q1 (Pilha: Z aa)

=======================================

Estado anterior: q1
Pilha anterior(Do início ao topo): Z aa
Entrada: -
Estados resultantes: Ramificação Morta

=======================================
-> A entrada não é aceita pelo autômato.
```