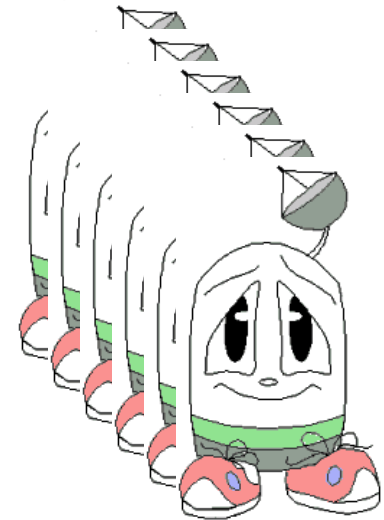


Automatos Finitos

Exercícios Resolvidos



Exercícios

1. Fazer um AFD M que reconhece:
 - a) $L(M) = \{x \in \{0,1\}^* \mid \text{nro de } 1\text{'s em } x \text{ é múltiplo de } 3\}$
 - b) $L(M) = \{x \in \{0,1\}^* \mid x \text{ contém a subcadeia } 001\}$
2. Fazer um AFD M que reconhece:
 - a) a formação de identificadores em Pascal
 - b) números inteiros sem sinal em Pascal
 - c) Operadores relacionais em Pascal
3. Para $\Sigma = \{0,1\}$ faça AFD's que reconheçam $L1 = \emptyset$ e $L2 = \Sigma^*$
4. Construa um AFD M que reconhece $L(M) = \{ab^n c \mid n \geq 0\}$
5. Construa um AFD M que reconhece $L(M) = \{a^n b^m \mid n, m \geq 0 \text{ e } n+m > 0\}$. Veja que $n+m > 0$ implica na não possibilidade da vazia nula.
6. Construa o AF e depois a GR para $L1 = \{a^n b^m \mid n, m \geq 0\}$ e $L2 = \{a, b\}^*$

AFD 1a e uma cadeia aceita (verde no JFLAP, com indicação do estado em que parou ao terminar a leitura da cadeia de entrada)

JFLAP : <untitled1>

File Input Test Convert Help

Editor Simulate: 001110 Simulate: 0011

δ :

```
graph LR; q0((q0)) -- 0 --> q0; q0 -- 1 --> q1((q1)); q1 -- 0 --> q0; q1 -- 1 --> q2((q2)); q2 -- 0 --> q1; q2 -- 1 --> q0; q1 -- 0 --> q1; q2 -- 0 --> q2; style q0 stroke:#000,stroke-width:2px; style q1 stroke:#000; style q2 stroke:#000;
```

$M = (\{q0, q1, q2\}, \{0, 1\}, q0, \delta, \{q0\})$

001110

Step Reset Freeze Thaw Trace Remove

Iniciar SCE_521_185 Microsoft PowerPoint ... JFLAP : <untitled1> PT 10:11

AFD 1a e uma cadeia NÃO aceita (rosa no JFLAP, com indicação do estado que parou ao terminar a leitura da cadeia de entrada)

The screenshot shows the JFLAP interface with a DFA diagram and a simulation window. The DFA has three states: q0 (start state, yellow), q1 (yellow), and q2 (green). Transitions are: q0 to q1 on 1, q1 to q0 on 1, q1 to q2 on 1, q2 to q0 on 1, and self-loops on 0 for all states. The simulation window shows the input string '0011' and the current state q2 highlighted in pink. The bottom toolbar includes buttons for Step, Reset, Freeze, Thaw, Trace, and Remove.

AFD 1b e uma cadeia aceita

JFLAP : <untitled2>

File Input Test Convert Help

Editor Simulate: 1001

δ :

```
graph LR; q0((q0)) -- 1 --> q0; q0 -- 0 --> q1((q1)); q1 -- 1 --> q0; q1 -- 0 --> q2((q2)); q2 -- 0 --> q2; q2 -- 1 --> q3(((q3))); q3 -- 0 --> q3; q3 -- 1 --> q3;
```

q3

1001

$$M = (\{q0, q1, q2, q3\}, \{0, 1\}, q0, \delta, \{q3\})$$

Step Reset Freeze Thaw Trace Remove

Windows Taskbar: Iniciar, SCE_521_185, Microsoft PowerPoint ..., JFLAP : <untitled2>, PT, 10:43

AFD 2a e cadeias aceitas e recusadas

$$M = (\{q_0, q_1\}, \{A..Z, a..z, 0..9, _ \}, q_0, \delta, \{q_1\})$$

JFLAP : <untitled3>

File Input Test Convert Help

Editor Multiple Inputs

L é o conjunto das letras
D é o conjunto dos dígitos
_ é o sublinhado

```
graph LR; q0((q0)) -- L̄ --> q1(((q1))); q1 -- "D̄  
L" --> q1;
```

Input	Result
_LLL	Accept
LDD	Accept
DD	Reject
LLL	Accept
D+D	Reject

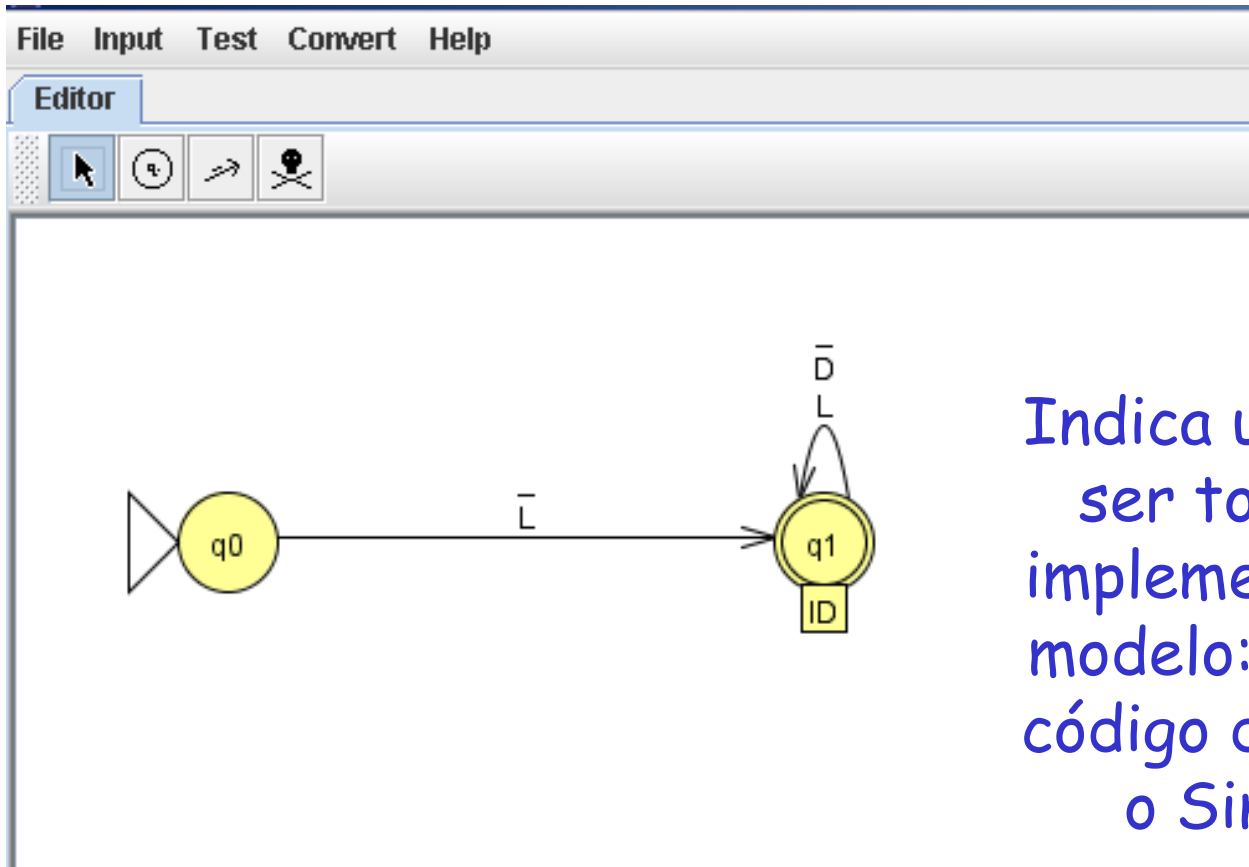
_LLL poderia ser _xyz
LDD poderia ser A23

...

```
var _xyz:integer;  
  __:real;  
  _ab_:boolean;  
begin  
  _xyz := 1;  
  __ := 23.9;  
  _ab_ := true  
end.
```

Run Inputs Clear Enter Lambda

Vejam a importância do rótulo



Indica uma ação a ser tomada na implementação do modelo: devolve o código de ID para o Sintático

AFD 2b e cadeias aceitas e recusadas

$$M = (\{q_0, q_1\}, \{0..9\}, q_0, \delta, \{q_1\})$$

JFLAP : <untitled4>

File Input Test Convert Help

Editor Multiple Inputs

δ :

```
graph LR; q0((q0)) -- 0 --> q1(((q1))); q0 -- 1 --> q1; q0 -- 2 --> q1; q0 -- 3 --> q1; q0 -- 4 --> q1; q0 -- 5 --> q1; q0 -- 6 --> q1; q0 -- 7 --> q1; q0 -- 8 --> q1; q0 -- 9 --> q1; q1 -- 0 --> q1; q1 -- 1 --> q1; q1 -- 2 --> q1; q1 -- 3 --> q1; q1 -- 4 --> q1; q1 -- 5 --> q1; q1 -- 6 --> q1; q1 -- 7 --> q1; q1 -- 8 --> q1; q1 -- 9 --> q1;
```

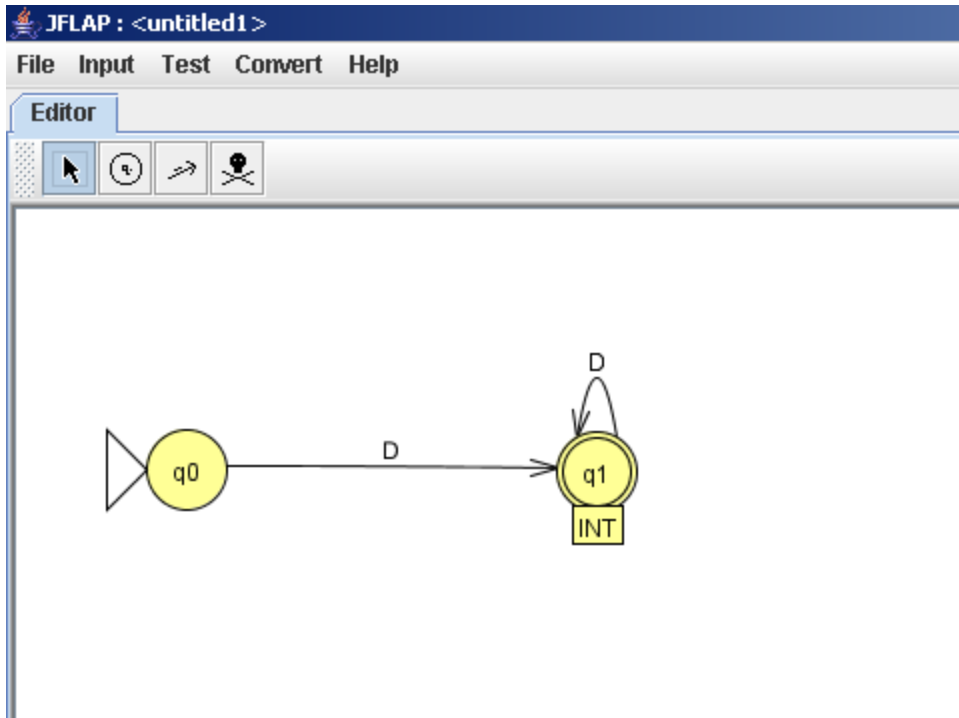
Input	Result
23	Accept
34567777	Accept
L23	Reject
3+6	Reject

Este autômato permite 0's a esquerda.

Façam uma versão que corrige o problema

Run Inputs Clear Enter Lambda

Vejam a importância do rótulo



Indica uma ação a ser tomada na implementação do modelo: devolve o código de INT para o Sintático

AFD 2c e cadeias aceitas e recusadas

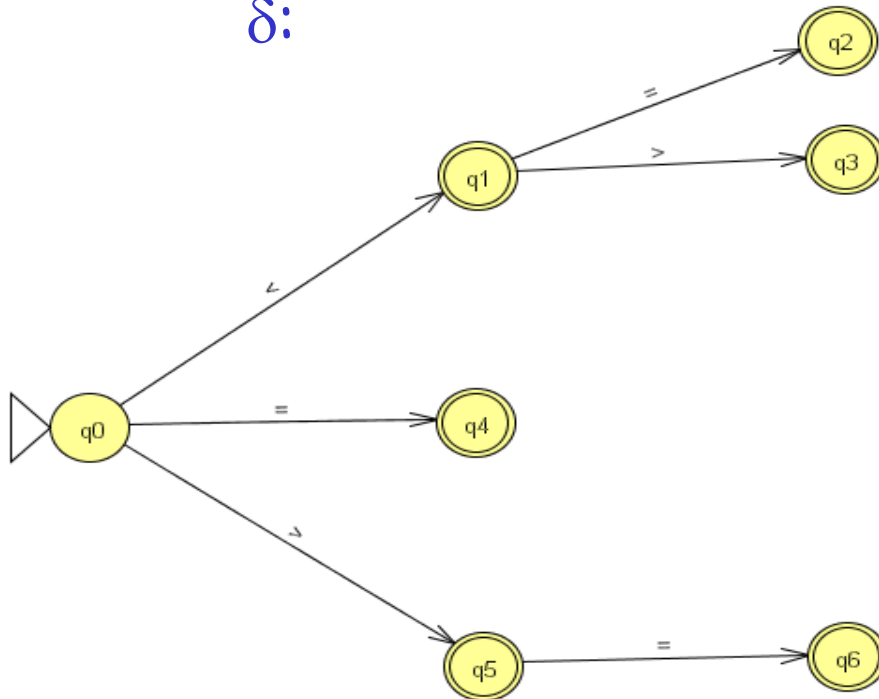
$M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{<, >, =\}, q_0, \delta, \{q_1, q_2, q_3, q_4, q_5, q_6\})$

JFLAP : <untitled5>

File Input Test Convert Help

Editor Multiple Inputs

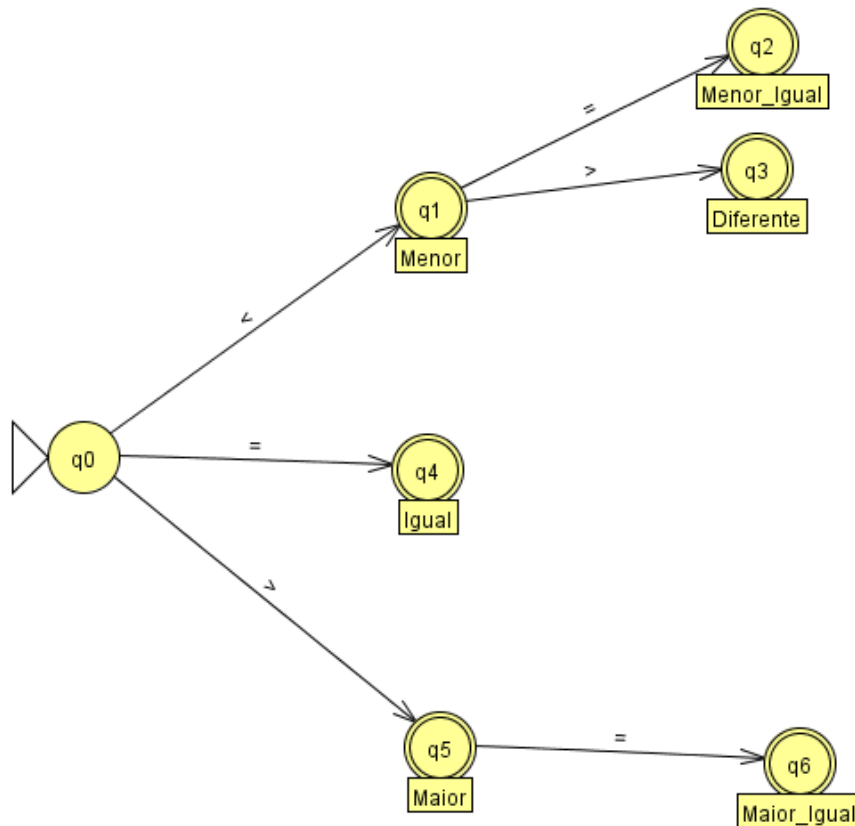
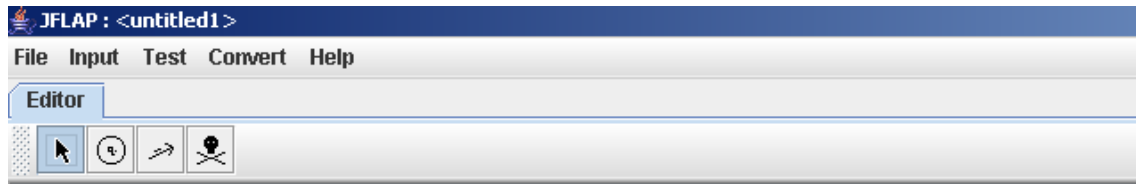
$\delta:$



Input	Result
=	Accept
< >	Accept
<	Accept
>	Accept
< =	Accept
> =	Accept
23	Reject
2 > 4	Reject

Run Inputs Clear Enter Lambda

Vejam a importância do rótulo



Indica ações a serem tomadas na implementação do modelo: devolve os códigos de cada operador relacional para o Sintático

Existe alguma diferença entre δ_1 e δ_2 ?
O que diferencia M1 de M2?

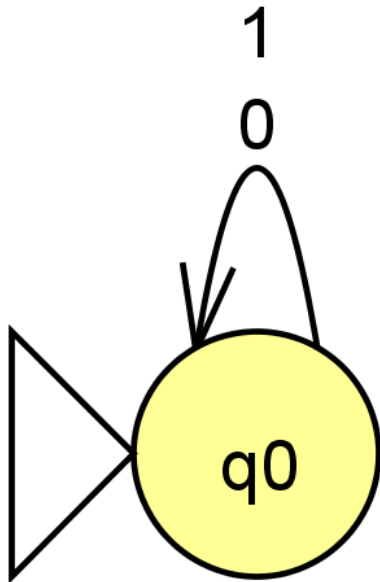
JFLAP : <untitled6>

File Input Test Convert Help

Editor Multiple Inputs

Input	Result
0	Reject
1	Reject
11	Reject
10	Reject
0001	Reject
	Reject

δ_1 :



JFLAP : <untitled6>

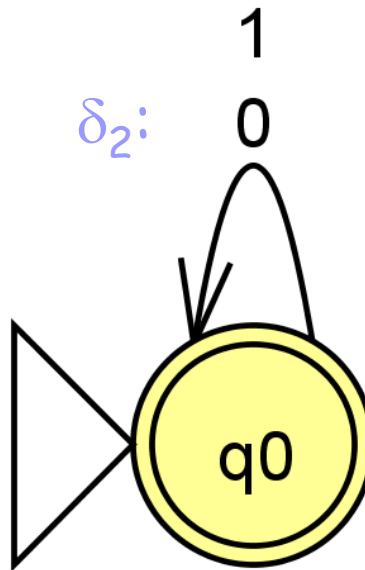
File Input Test Convert Help

Editor Multiple Inputs

Input	Result
0	Accept
1	Accept
11	Accept
10	Accept
0001	Accept
	Accept

Run Inputs Clear Enter Lambda

δ_2 :



AFD 4 e cadeias aceitas e recusadas

$$M = (\{q_0, q_1, q_2\}, \{a, b, c\}, q_0, \delta, \{q_2\})$$

JFLAP : <untitled7>

File Input Test Convert Help

Editor Multiple Inputs

δ :

```
graph LR; q0((q0)) -- a --> q1((q1)); q1 -- b --> q1; q1 -- c --> q2(((q2)))
```

Input	Result
a	Reject
abbc	Accept
ac	Accept
abc	Accept
acc	Reject

Run Inputs Clear Enter Lambda

AFD 5 e cadeia reconhecida

$$M = (\{q_0, q_1, q_2\}, \{a, b\}, q_0, \delta, \{q_1, q_2\})$$

The screenshot shows the JFLAP software interface. On the left, a DFA diagram is displayed with three states: q_0 (start state), q_1 , and q_2 . The transitions are: $q_0 \xrightarrow{a} q_2$, $q_0 \xrightarrow{b} q_1$, $q_2 \xrightarrow{a} q_2$, $q_2 \xrightarrow{b} q_1$, and $q_1 \xrightarrow{b} q_1$. The symbol $\delta:$ is shown next to the diagram.

On the right, a dialog box titled "Accepting configuration found!" is open. It displays a vertical sequence of configurations for the input string "aabbbbb". The configurations are:

- q_0 aabbbbb
- q_2 aabbbbb
- q_2 aabbbbb
- q_1 aabbbbb
- q_1 aabbbbb
- q_1 aabbbbb
- q_1 aabbbbb
- q_1 aabbbbb

The dialog box includes "Keep looking" and "I'm done" buttons.

6) Quais as linguagens reconhecidas pelos AFD's abaixo?

JFLAP : <untitled10>

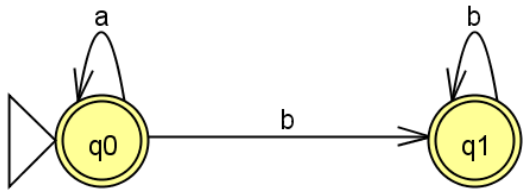
File Input Test Convert Help

Editor Multiple Inputs

$\delta_1:$

Input	Result
a	Accept
aa	Accept
aaa	Accept
b	Accept
bbbb	Accept
ab	Accept
aaab	Accept
abbbbb	Accept
aabb	Accept
bababa	Reject
bbbbaa	Reject

$$M1 = (\{q0, q1\}, \{a, b\}, q0, \delta, \{q0, q1\})$$



JFLAP : <untitled9>

File Input Test Convert Help

Editor Multiple Inputs

$\delta_2:$

```

    graph LR
      start(( )) --> q0(((q0)))
      q0 -- a --> q0
      q0 -- b --> q0
  
```

Input	Result
a	Accept
aa	Accept
aaa	Accept
b	Accept
bbbb	Accept
ab	Accept
aaab	Accept
abbbbb	Accept
aabb	Accept
bababa	Accept
bbbbaa	Accept

Run Inputs Clear Enter Lambda

$$M2 = (\{q0\}, \{a, b\}, q0, \delta, \{q0\})$$

$$L1 = \{a^n b^m \mid n, m \geq 0\}$$

$$G = (\{Q0, Q1, S\}, \{a, b\}, P, S)$$

$$P = \{Q0 \rightarrow aQ0$$

$$Q0 \rightarrow a$$

$$Q0 \rightarrow bQ1$$

$$Q0 \rightarrow b$$

$$Q1 \rightarrow bQ1$$

$$Q1 \rightarrow b$$

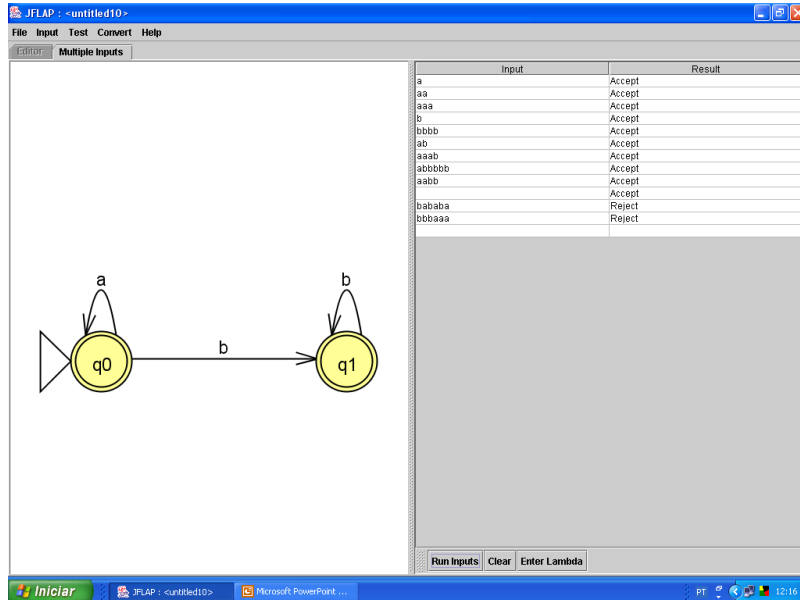
$$S \rightarrow aQ0$$

$$S \rightarrow a$$

$$S \rightarrow bQ1$$

$$S \rightarrow b$$

$$S \rightarrow \lambda\}$$



$L2 = \{a,b\}^*$

$G = (\{Q_0, S\}, \{a, b\}, P, S)$

$P = \{Q_0 \rightarrow aQ_0\}$

$Q_0 \rightarrow a$

$Q_0 \rightarrow bQ_0$

$Q_0 \rightarrow b$

$S \rightarrow aQ_0$

$S \rightarrow a$

$S \rightarrow bQ_0$

$S \rightarrow b$

$S \rightarrow \lambda$

The screenshot shows the JFLAP software interface. On the left, a finite automaton diagram is displayed with a single state q_0 (represented by a yellow circle) and a start symbol (represented by a triangle). There are two self-loops on q_0 : one labeled 'a' and one labeled 'b'. On the right, a table shows the results of running various inputs through the automaton. The table has two columns: 'Input' and 'Result'. The results are all 'Accept'.

Input	Result
a	Accept
aa	Accept
aaa	Accept
b	Accept
bbbb	Accept
ab	Accept
aaab	Accept
abbbb	Accept
aabb	Accept
aaab	Accept
bababa	Accept
bbbbaa	Accept