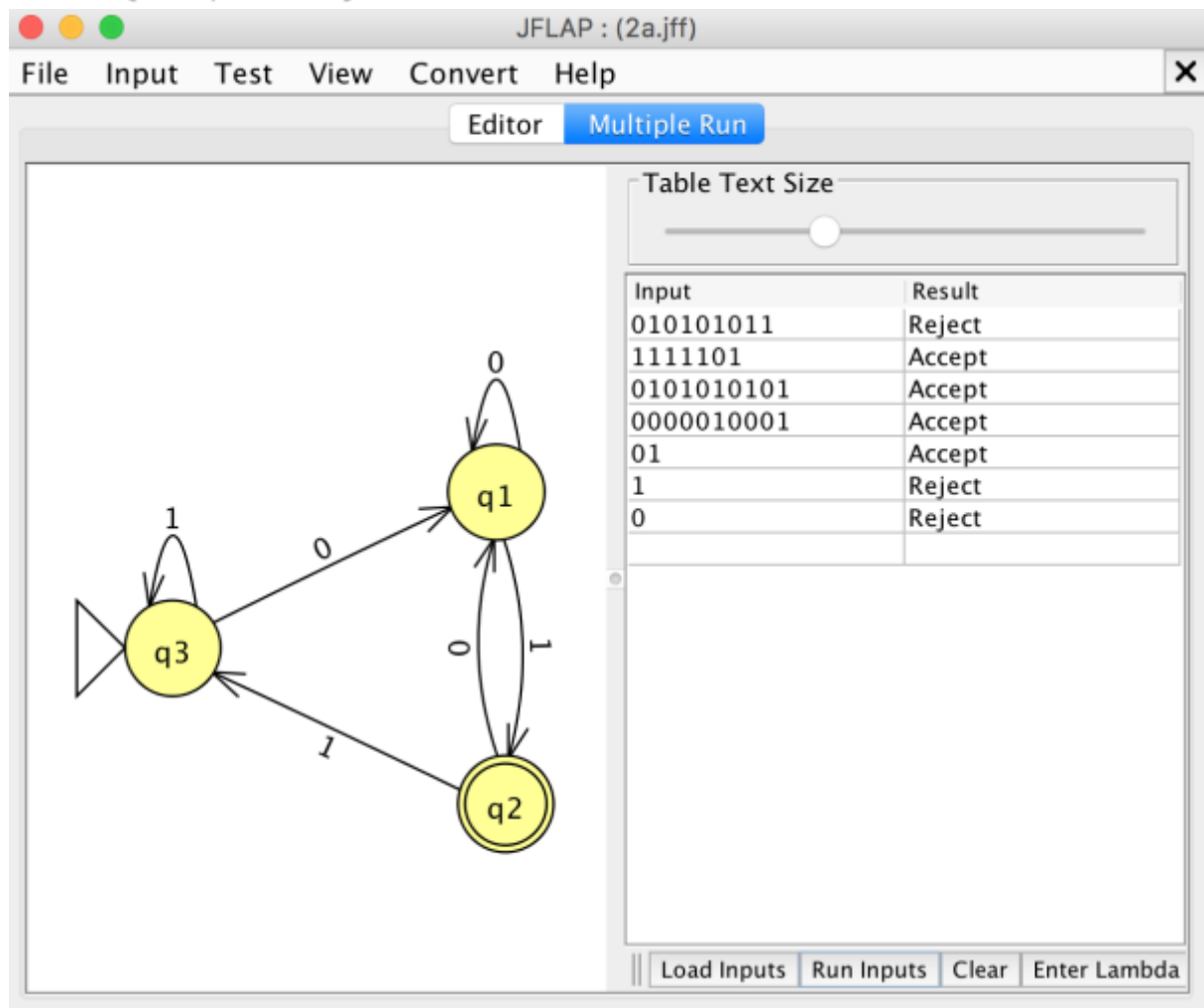


Konzepte konkreter und abstrakter Maschinen

Übungsblatt 1 20. Oktober 2016

Aufgabe 2

$$L_A = \{w01 \mid w \in V^*\}$$



$L_B = \{w \mid w \in V^* \wedge \|w\|_0 = \|w\|_1\}$, wobei $\|w\|_x$ die Anzahl des Zeichens x im Wort w ist.

JFLAP : <untitled2>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

```

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

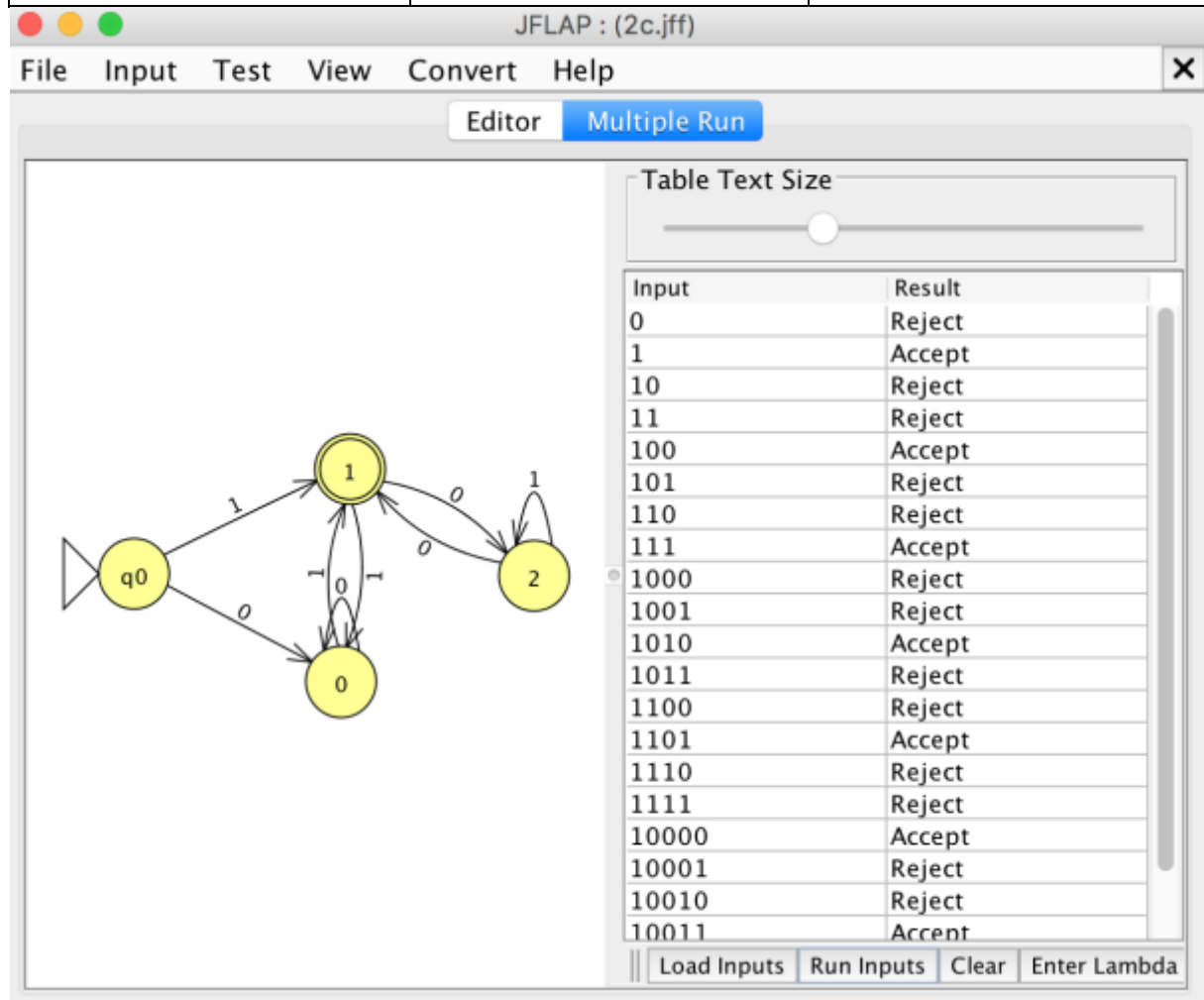
Input	Result
01010101	Accept
00000001	Reject
11111101	Reject
10	Accept
0011	Reject
00001111	Reject
1100	Reject
0110	Accept
1100111000	Reject

Load Inputs Run Inputs Clear Enter Lambda

nicht möglich => Fälle 000000111111 nicht abbildbar als Automat, da kein Zähler vorhanden

$L_C = \{w \mid w \in V^* \wedge (\|w\|_{dez} \bmod 3) = 1\}$, wobei $\|w\|_{dez}$ das dezimale Äquivalent des Wortes w ist, wenn dieses als Dualzahl interpretiert wird. und $\bmod k$ der Rest bei Division durch k ist.

Dez.Wert	Bin.Wert	Mod 3
0	0000	0
1	0001	1
2	0010	2
3	0011	0
4	0100	1
5	0101	2
6	0110	0
7	0111	1
8	1000	2
9	1001	0
10	1010	1
11	1011	2
12	1100	0
13	1101	1
14	1110	2
15	1111	0



$$L_D = \{0^k 1^k \mid k \in \mathbb{N}\}$$

wie bei L_B fehlt hier ein Zähler, mit dem die Anzahl 0en oder 1en „überprüft“ werden kann

Aufgabe 3

Wurde das letzte Mal schon abgegeben, dafür dieses Mal zusätzliche Abgabe des DFA Moduls.

Modul

Question 1.

1. Load the NFA in the file NF-1.jflap.
2. Enter inputs w_1 , w_2 , w_3 and w_4 in the discussion above and verify that the first three inputs are accepted and the last is rejected, as shown in the diagram below.
3. List all configurations for the inputs w_1 , w_2 , w_3 and w_4 .
4. Enter six more inputs of your own, three of which are accepted and the other three are rejected.

The screenshot shows the JFLAP interface with the NFA diagram on the left and a table of input results on the right. The NFA has states p, q, r, s, t, and u. State p is the start state, and state u is the final state. Transitions are: p to q on 'a', p to t on 'a', q to r on 'b', r to s on 'c', s to p on 'c', s to q on 'b', s to r on 'a', t to u on 'c', and u to t on 'c'.

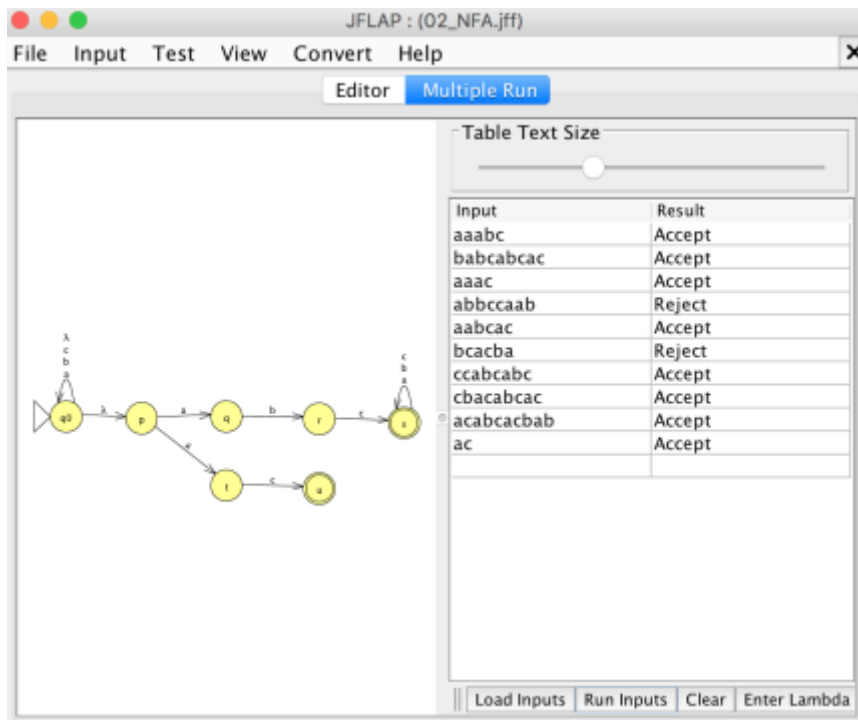
Input	Result
aaabc	Accept
babcbacac	Accept
aaac	Accept
abbccaab	Reject
aabccac	Accept
bcacba	Reject
ccabcbabc	Accept
cbacabcbac	Accept
acabcbabab	Accept
ac	Accept

Question 2. Complete the table above by filling out all blank cells in the table. Explain your answers.

	a	b	c	λ
p	{p,q,t}	{p}	{p}	ϕ
q	ϕ	{r}	ϕ	ϕ
r	ϕ	ϕ	{s}	ϕ
s	{s}	{s}	{s}	ϕ
t	ϕ	ϕ	{u}	ϕ
u	ϕ	ϕ	ϕ	ϕ

Question 3.

1. Load the NFA in the file NFA-2.jflap, as shown below.
2. Verify that $Q = \{q_0, p, q, r, s, t, u\}$.
3. Which state is the initial state?
4. What is F ?
5. Verify that $\delta(p, a) = \{q, t\}$.
6. What is $\delta(q_0, \lambda)$?
7. What is $\delta(r, b)$?
8. What is $\delta(t, \lambda)$?
9. What is $\delta(u, b)$?
10. What is $\delta(s, c)$?



q_0 is the initial state.

$F = \{s, u\}$

$\delta(q_0, \lambda) = \{q_0\}$

$\delta(r, b) = \emptyset$

$\delta(t, \lambda) = \emptyset$

$\delta(u, b) = \emptyset$

$\delta(s, c) = \{s\}$

Question 4. Do NFA-1.jflap and NFA-2.jflap accept the same set of words? Explain your answer in detail.

Yes, because they are the same NFA, except the λ transition in NFA-2.

Exercise 1

Define an NFA that recognizes the following language L over $\Sigma = \{a, b\}$:
 $L = \{ w \mid w \text{ ends with } aa \}$.

Recall that an NFA is defined as a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where

- Q is a finite set of states
- Σ is a finite alphabet
- δ is the transition function, $\delta: Q \times \Sigma \rightarrow \text{PowerSet}(Q)$
- q_0 is the start state ($q_0 \in Q$)
- F is a set of accept states ($F \subseteq Q$)

JFLAP : <untitled1>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

```

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

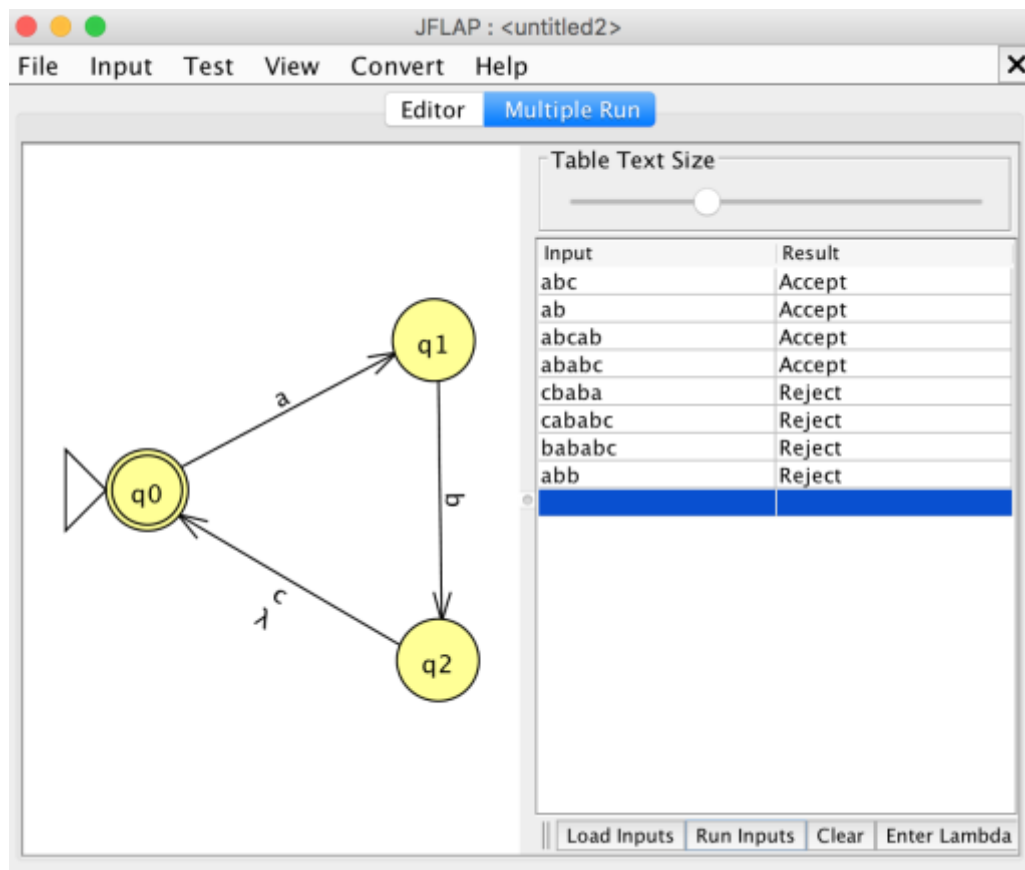
Input	Result
abaa	Accept
aa	Accept
bab	Reject
baba	Reject
aaaabb	Reject
bbbabb	Reject
bbbbbbaaa	Accept
bb	Reject

Load Inputs Run Inputs Clear Enter Lambda

Exercise 2

Problem:

Construct an NFA that accepts the language $\{ab, abc\}^*$. This is the set of strings where ab and abc may be repeated. Example strings include $abcb, ababcb, abcabcb, abcabcb, abcabcb$, and the empty string.



Exercise 3

Given the Alphabet $\{a,b,c\}$ construct a NFA which accepts $(a|b|c)^*c$

