

Konzepte konkreter und abstrakter Maschinen

Übungsblatt 0 19. September 2016

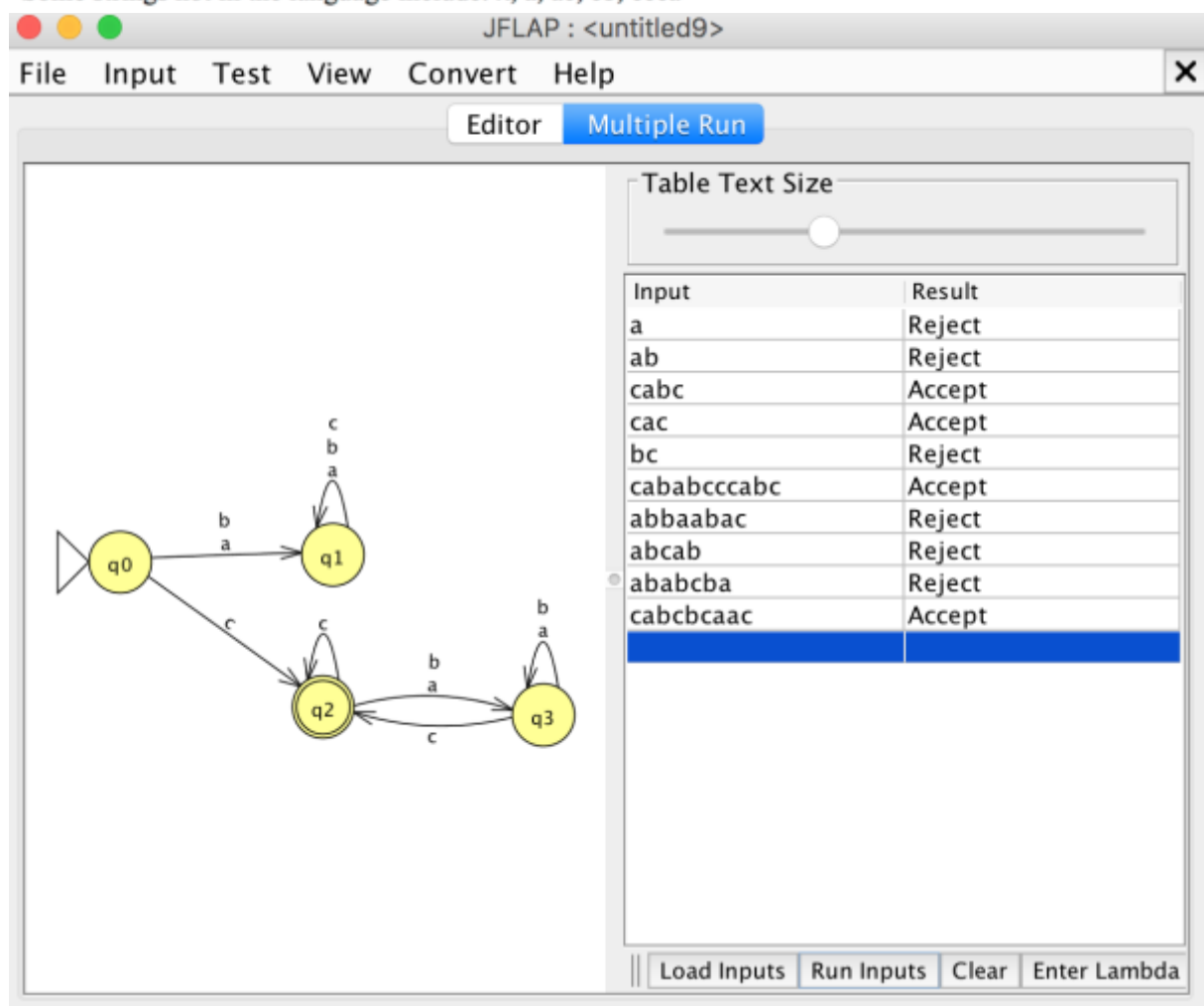
Aufgabe 3

Modul

Consider the regular language L over the alphabet $\{a, b, c\}$ comprised of all strings that begin and end with c . Let's construct a DFA M to recognize that language.

Some strings in the language include: $c, cc, ccc, cac, cabc, cabcbabc$

Some strings not in the language include: $\lambda, a, ac, cb, ccca$



Create a DFA for the language

$\{w \mid w \text{ is a string symbols from alphabet } \{a, b\} \text{ in which } a \text{ never follows } b\}$.

JFLAP : <untitled10>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

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

Input	Result
	Accept
a	Accept
b	Accept
aa	Accept
ab	Accept
aaa	Accept
aab	Accept
abb	Accept
abbb	Accept
bbbbb	Accept
ba	Reject
bab	Reject
aba	Reject
abba	Reject
bbba	Reject
bbab	Reject

Load Inputs Run Inputs Clear Enter Lambda

Exercise 1

$L = \{w \in \Sigma^* \mid w \text{ has an even number of } a\text{'s and an odd number of } b\text{'s}\}.$

JFLAP : (DFA1_1.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

```

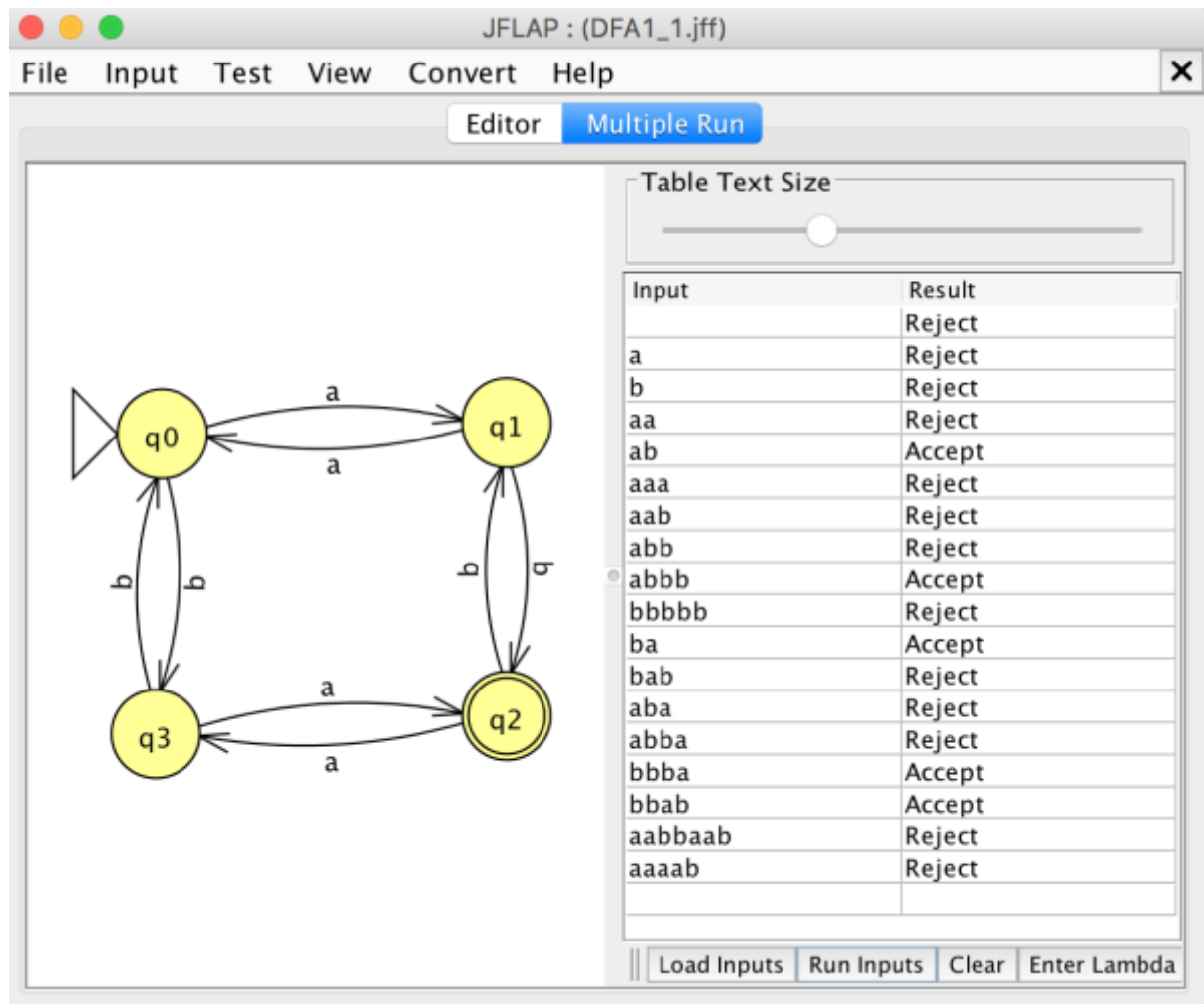
graph LR
    start(( )) --> q0((q0))
    q0 -- a --> q1((q1))
    q1 -- a --> q0
    q0 -- b --> q3(((q3)))
    q3 -- b --> q0
    q1 -- b --> q2((q2))
    q2 -- b --> q1
    q2 -- a --> q3
    q3 -- a --> q2
  
```

Input	Result
	Reject
a	Reject
b	Accept
aa	Reject
ab	Reject
aaa	Reject
aab	Accept
abb	Reject
abbb	Reject
bbbbb	Accept
ba	Reject
bab	Reject
aba	Accept
abba	Reject
bbba	Reject
bbab	Reject
aabbaab	Accept
aaaab	Accept

Load Inputs Run Inputs Clear Enter Lambda

Repeat the above steps for the language

$L_1 = \{w \in \Sigma^* | w \text{ has an odd number of } a\text{'s and an odd number of } b\text{'s}\}$



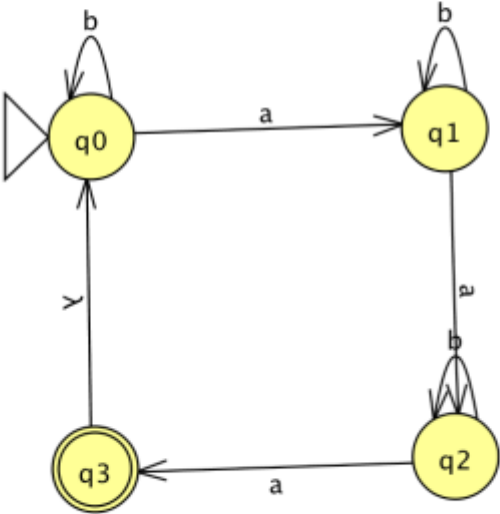
Repeat the above steps for the language

$L_2 = \{w \in \Sigma^* \mid \text{The number of } a\text{'s in } w \text{ is a multiple of 3}\}$

JFLAP : (DFA1_3.jff)

File Input Test View Convert Help

Editor Multiple Run



The DFA diagram shows four states: q0, q1, q2, and q3. q0 is the start state (indicated by a double arrow) and q3 is the accept state (indicated by a double circle). Transitions are as follows: q0 has a self-loop on 'b' and a transition to q1 on 'a'; q1 has a self-loop on 'b' and a transition to q2 on 'a'; q2 has a self-loop on 'b' and a transition to q3 on 'a'; q3 has a transition to q0 on 'b'.

Table Text Size

Input	Result
	Reject
a	Reject
b	Reject
aa	Reject
ab	Reject
aaa	Accept
aab	Reject
abb	Reject
abbb	Reject
bbbbb	Reject
ba	Reject
bab	Reject
aba	Reject
abba	Reject
bbba	Reject
bbab	Reject
aabbaabaa	Accept
aaaabaa	Accept

Load Inputs Run Inputs Clear Enter Lambda

Exercise 2

Construct a DFA for $L = \{ab^n a^m : n \geq 2, m \geq 3\}$.

JFLAP : (DFA2_1.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
	Reject
a	Reject
b	Reject
aa	Reject
ab	Reject
aaa	Reject
aab	Reject
abb	Reject
abbb	Reject
bbbbb	Reject
ba	Reject
bab	Reject
aba	Reject
abba	Reject
bbba	Reject
bbab	Reject
abbaaa	Accept
abbbbbaaa	Accept
abbbbaaaaaa	Accept

Load Inputs Run Inputs Clear Enter Lambda

Exercise 3

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

JFLAP : (DFA3_1.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

```

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

Input	Result
	Reject
ac	Accept
bc	Accept
aa	Reject
abc	Accept
aaac	Accept
aacb	Reject
abb	Reject
abbbc	Accept
bbbbbb	Reject
ba	Reject
babc	Accept
aba	Reject
abcba	Reject
bbbac	Accept
bbab	Reject
abbcaaa	Reject
abbbbcbaaa	Reject
abbbbaaaaaaac	Accept

Load Inputs Run Inputs Clear Enter Lambda