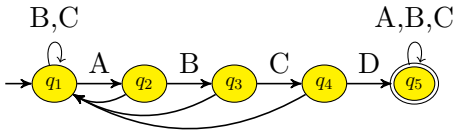
Algorithmen und Datenstrukturen
Musterlösung 13

Martin Avanzini <martin.avanzini@uibk.ac.at>
Thomas Bauereiß <thomas.bauereiss@uibk.ac.at>
Herbert Jordan <herbert@dps.uibk.ac.at>
René Thiemann <rene.thiemann@uibk.ac.at>

21. Juni, zur Besprechung am 28. Juni

Aufgabe 1) Knuth-Morris-Pratt

1. (a)

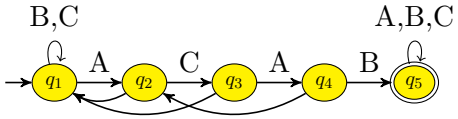


Hier entsprechen unbeschriftete Pfeile ϵ-Transitionen.

Akzeptierender Lauf:

$$q_1 \xrightarrow{A} q_2 \xrightarrow{B} q_3 \xrightarrow{C} q_4 \xrightarrow{\epsilon} q_1 \xrightarrow{A} q_2 \xrightarrow{B} q_3 \xrightarrow{C} q_4 \xrightarrow{D} q_5$$

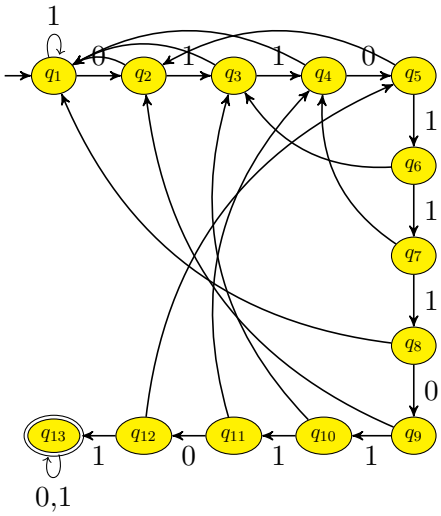
(b)



Akzeptierender Lauf:

$$q_1 \xrightarrow{A} q_2 \xrightarrow{C} q_3 \xrightarrow{A} q_4 \xrightarrow{\epsilon} q_2 \xrightarrow{C} q_3 \xrightarrow{A} q_4 \xrightarrow{B} q_5$$

(c)



Akzeptierender Lauf:

$$q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_4 \xrightarrow{0} q_5 \xrightarrow{1} q_6 \xrightarrow{1} q_7 \xrightarrow{\epsilon} q_4 \xrightarrow{0} q_5 \xrightarrow{1} q_6 \xrightarrow{1} q_7 \xrightarrow{1} q_8 \xrightarrow{0} q_9 \xrightarrow{\epsilon} q_2 \xrightarrow{\epsilon}$$
$$q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_4 \xrightarrow{0} q_5 \xrightarrow{1} q_6 \xrightarrow{1} q_7 \xrightarrow{1} q_8 \xrightarrow{0} q_9 \xrightarrow{1} q_{10} \xrightarrow{1} q_{11} \xrightarrow{0} q_{12} \xrightarrow{1} q_{13}$$

2.

```
1  public class Kmp {
2      int[] next;
3      String w;
4
5      public Kmp(String search) {
6      int i = 0;
7      int j = -1;
8      int m = search.length();
9      next = new int[m+1];
10     next[i] = j;
11     w = search;
12
13     while (i < m) {
14         if (j == -1 || w.charAt(i) == w.charAt(j)) {
15         i++; j++;
16         next[i] = j;
17         } else {
18         j = next[j];
19         }
20     };
21     }
22
23     void search(String s, boolean all) {
24     int i = 0;
25     int j = 0;
26     int m = w.length();
27     int n = s.length();
28     boolean found = false;
29     while (i < n) {
30         if (j == -1 || s.charAt(i) == w.charAt(j)) {
31         i++; j++;
32         } else {
33         j = next[j];
34         }
35         if (j >= m) {
36         found = true;
37         System.out.println("w found in s at position " + (i - m));
```

```
38          j = next[j];
39          if(!all) break;
40          }
41      };
42      if (!found) {
43          System.out.println("w not found in s");
44      }
45      }
46
47      void find(String s) {
48      search(s,false);
49      }
50
51      void findAll(String s) {
52      search(s,true);
53      }
54
55      public static void main(String[] args) {
56      Kmp k = new Kmp(args[0]);
57      k.find(args[1]);
58      System.out.println("");
59      k.findAll(args[1]);
60      }
61  }
```

Aufgabe 2) Boyer-Moore Kleinbuchstaben kennzeichnen entdeckte Konflikte.

```
ABCABCD
ABCd
   ABCD

ACACAB
ACAb
  ACAB
```

Aufgabe 3) Klausurvorbereitung