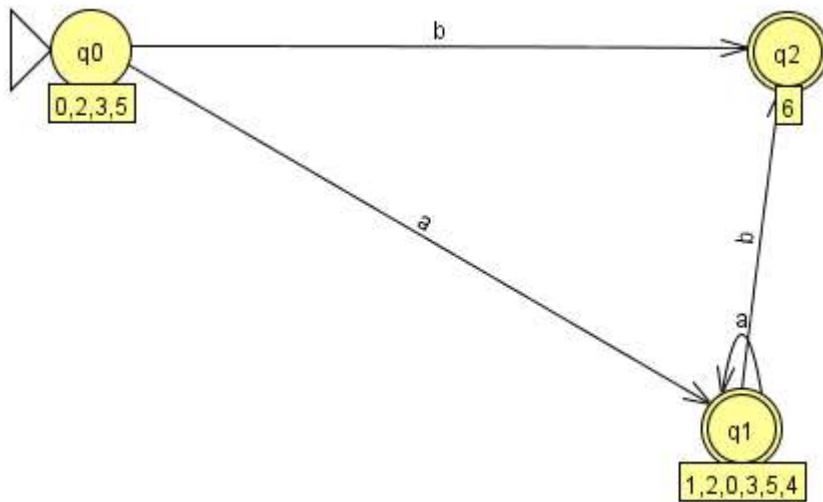


## Aufgabe 6

Minimieren Sie die Anzahl der Zustände der DEA aus Aufgabe 5.

a)

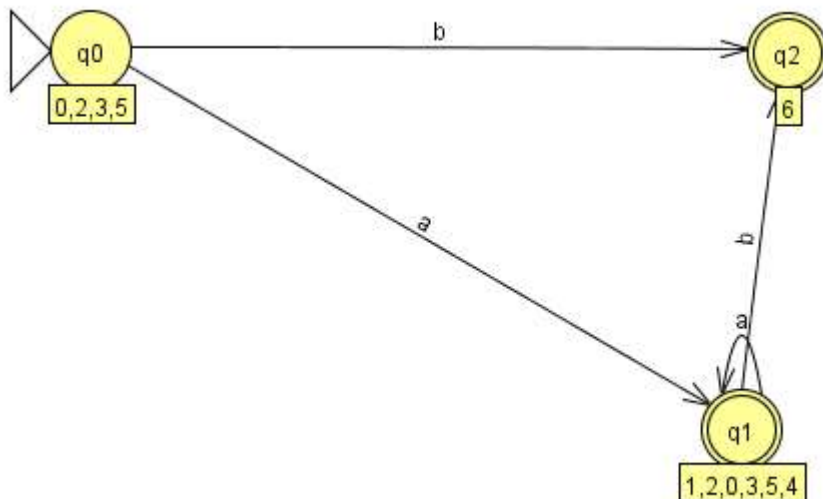


1. Zerlegung:  $\ddot{A}1 = \{q1, q2\}$   
 $\ddot{A}2 = \{q0\}$

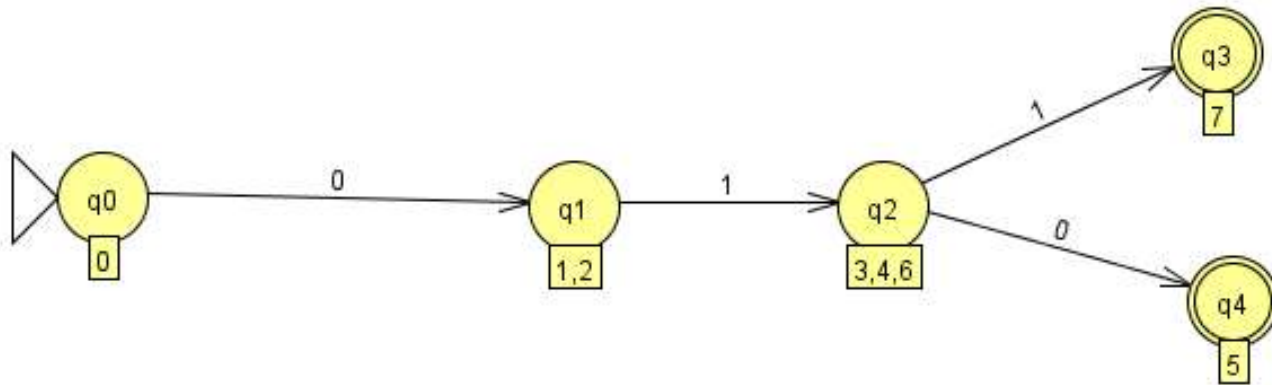
$u(q1, a) = 1 \in \ddot{A}1$

$u(q2, a) = \{\}$

Verfeinerung ergibt Ausgangsautomat



b)



1. Zerlegung:  $\ddot{A}1 = \{q3, q4\}$   
 $\ddot{A}2 = \{q0, q1, q2\}$

$u(q1, 1) = q2 \in \ddot{A}2$

$u(q2, 1) = q3 \in \ddot{A}1 \rightarrow$  Verfeinerung (Abspaltung  $q2$ )

2. Zerlegung:  $\ddot{A}1 = \{q3, q4\}$   
 $\ddot{A}2 = \{q0, q1\}$   
 $\ddot{A}3 = \{q2\}$

$u(q0, 0) = q1 \in \ddot{A}2$

$u(q1, 0) = \{\} \rightarrow$  Verfeinerung (Abspaltung  $q1$ )

3. Zerlegung:  $\ddot{A}1 = \{q3, q4\}$   
 $\ddot{A}2 = \{q0\}$   
 $\ddot{A}3 = \{q1\}$   
 $\ddot{A}4 = \{q2\}$

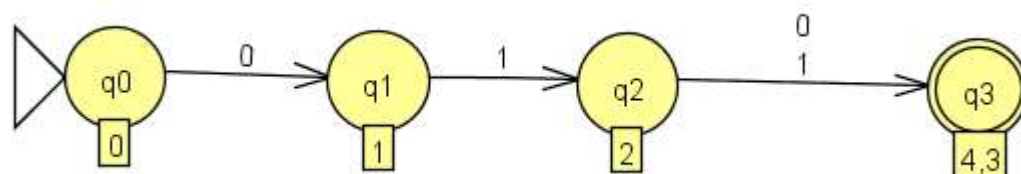
$u(q3, 0) = \{\}$

$u(q4, 0) = \{\}$

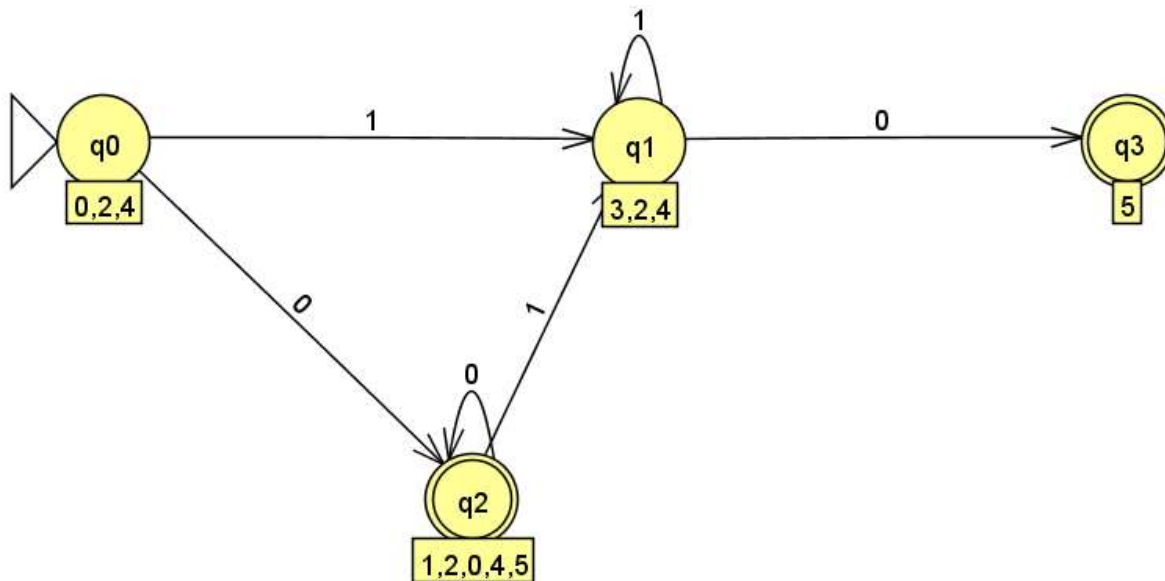
$u(q3, 1) = \{\}$

$u(q4, 1) = \{\}$

somit Zerlegung stabil, neuer minimaler Automat:



c)



1. Zerlegung:  $\ddot{A}1 = \{q2, q3\}$   
 $\ddot{A}2 = \{q0, q1\}$

$u(q2, 0) = q2 \in \ddot{A}1$

$u(q3, 0) = \{\} \rightarrow$  Verfeinerung (Abspaltung  $q3$ )

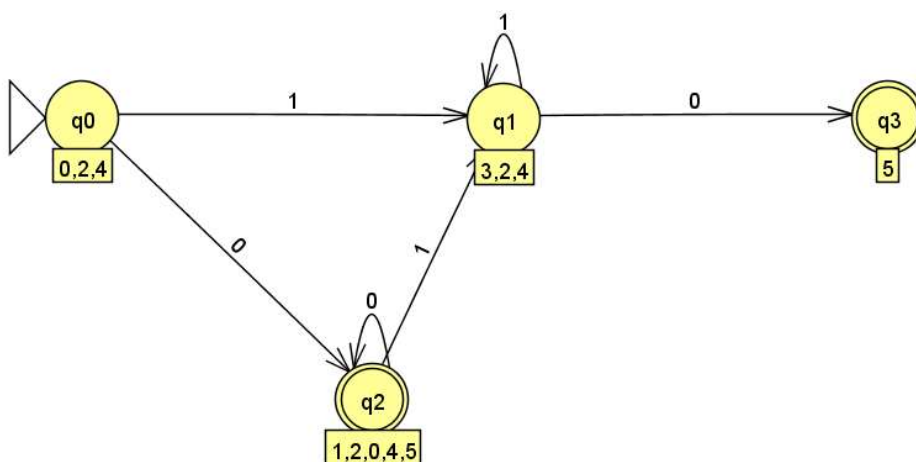
2. Zerlegung:  $\ddot{A}1 = \{q2\}$   
 $\ddot{A}2 = \{q3\}$   
 $\ddot{A}3 = \{q0, q1\}$

$u(q0, 0) = q2 \in \ddot{A}1$

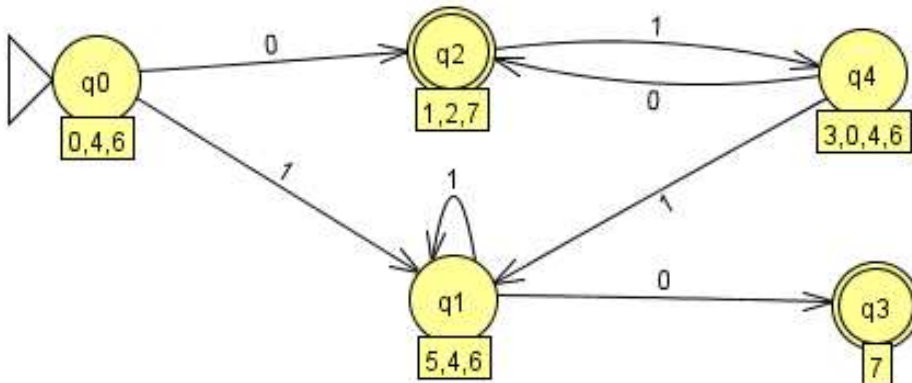
$u(q1, 0) = q3 \in \ddot{A}2 \rightarrow$  Verfeinerung (Abspaltung  $q1$ ) ergibt Ausgangsautomat:

$u(q0, 1) = q1 \in \ddot{A}3$

$u(q1, 1) = q1 \in \ddot{A}3$



d)



1. Zerlegung:  $\ddot{A}1 = \{q2, q3\}$   
 $\ddot{A}2 = \{q0, q1, q4\}$

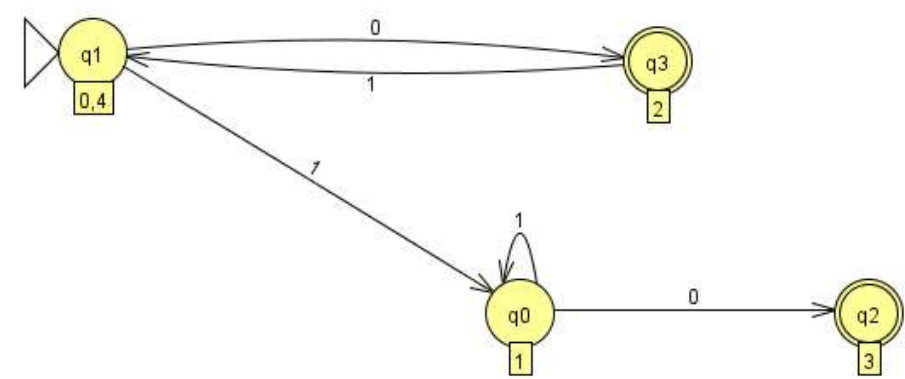
$u(q2, 1) = q4 \in \ddot{A}2$   
 $u(q3, 1) = \{\}$   $\rightarrow$  q3 abspalten:

2. Zerlegung:  $\ddot{A}1 = \{q2\}$   
 $\ddot{A}2 = \{q3\}$   
 $\ddot{A}3 = \{q0, q1, q4\}$

$u(q0, 0) = q2 \in \ddot{A}1$   
 $u(q4, 0) = q2 \in \ddot{A}1$   
 $u(q1, 0) = q3 \in \ddot{A}2 \rightarrow$  q1 abspalten:

3. Zerlegung:  $\ddot{A}1 = \{q2\}$   
 $\ddot{A}2 = \{q3\}$   
 $\ddot{A}3 = \{q0, q4\}$   
 $\ddot{A}4 = \{q1\}$

$u(q0, 0) = q2 \in \ddot{A}1$   
 $u(q4, 0) = q2 \in \ddot{A}1$  Verhalten gleich  
 $u(q0, 1) = q1 \in \ddot{A}4$   
 $u(q4, 1) = q1 \in \ddot{A}4$  Verhalten gleich, also Zerlegung stabil, keine weitere Verfeinerung!



## Aufgabe 7

Realisieren Sie die in Aufgabe 6 generierten DEA in einem C++-Programm.

a)

```
#include "scanner.h"
int scanner()
{
    int zustand=0; // Anfangszustand
    char zeichen='z'; // Anfangsbelegung
    while (zeichen !='\n')
    {
        zeichen = getchar(); // aktuelles Zeichen
        switch (zustand)
        {
            case 0:
                if (zeichen =='a')
                    zustand=1;
                else if (zeichen =='b')
                    zustand=2;
                else
                    zustand=99; // Fehlerzustand = 99
                break;
            case 1:
                if (zeichen =='a')
                    zustand=1;
                else if (zeichen =='b')
                    zustand=2;
                else if (zeichen == '\n')
                    zustand=1;
                else
                    zustand=99;
                break;
            case 2:
                if (zeichen != '\n') zustand=99;
                break;
            default:
                zustand=99;
                break;
        }
    }
    if ((zustand==1) || (zustand==2))
        printf("Eingabe akzeptiert!\n");
    else
        printf("Eingabe nicht akzeptiert!\n");
    return 0;
}
```

b)

```
int scannerb()
{
    int zustand=0; // Anfangszustand
    char zeichen='z'; // Anfangsbelegung
    while (zeichen !='\n')
    {
        zeichen = getchar(); // aktuelles Zeichen
        switch (zustand)
        {
            case 0:
                if (zeichen =='0')
                    zustand=1;
                else
                    zustand=99; // Fehlerzustand = 99
            case 1:
                if (zeichen =='1')
                    zustand=2;
                else
                    zustand=99;
                break;
            case 2:
                if ((zeichen =='0')||(zeichen =='1'))
                    zustand=3;
                else
                    zustand=99;
                break;
            case 3:
                if (zeichen != '\n') zustand=99; //Endzustand
                break;
            default:
                zustand=99;
                break;
        }
    }
    if (zustand==3)
        printf("Eingabe akzeptiert!\n");
    else
        printf("Eingabe nicht akzeptiert!\n");
    return 0;
}
```

c)

```
int scannerc()
{
    int zustand=0; // Anfangszustand
    char zeichen='z'; // Anfangsbelegung, damit while-Schleife nicht blockiert
    while (zeichen !='\n')
    {
        zeichen = getchar(); // aktuelles Zeichen
        switch (zustand)
        {
            case 0:
                if (zeichen =='0')
                    zustand=2;
                else if (zeichen =='1')
                    zustand=1;
                else
                    zustand=99; // Fehlerzustand = 99
                break;
            case 1:
                if (zeichen =='0')
                    zustand=3;
                else if (zeichen =='1')
                    zustand=1;
                else
                    zustand=99;
                break;
            case 2:
                if (zeichen =='0')
                    zustand=2;
                else if (zeichen =='1')
                    zustand=1;
                else if (zeichen =='\n') // Endzustand!
                    zustand=2;
                else
                    zustand=99;
                break;
            case 3:
                if (zeichen !='\n') // Endzustand
                    zustand=99;
                break;
            default:
                zustand=99;
                break;
        }
    }
    if ((zustand==2) || (zustand==3))
        printf("Eingabe akzeptiert!\n");
    else
        printf("Eingabe nicht akzeptiert!\n");
    return 0;
}
```



## Compiler Übung 4 Lösungsansätze

### Wintersemester 2024/2025

```
#include "scannerd.h"
int scannerd()
{    // scanner für (01)*1*0
    int zustand=1; // Anfangszustand
    char zeichen='9'; // Anfangsbelegung
    while (zeichen !='\n')
    {
        zeichen = getchar(); // aktuelles Zeichen
        switch (zustand)
        {
            case 1:
                if (zeichen =='0')
                    zustand=3;
                else if (zeichen =='1')
                    zustand=0;
                else
                    zustand=99;        // Fehlerzustand = 99
                break;
            case 0:
                if (zeichen =='0')
                    zustand=2;
                else if (zeichen =='1')
                    zustand=0;
                else
                    zustand=99;
                break;
            case 2:
                if (zeichen !='\n')
                    zustand=99;
                break;
            case 3:
                if (zeichen =='1')
                    zustand=1;
                else if (zeichen =='\n')
                    zustand=3;
                else
                    zustand=99;
                break;
            default:
                zustand=99;
                break;
        }
    }
    if ((zustand==3) || (zustand==2))
        printf("Eingabe akzeptiert!\n");
    else
        printf("Eingabe nicht akzeptiert!\n");
    return 0;
}
```

## Aufgabe 8

```
%{
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
}%
ziffer [0-9]
letter [a-zA-Z]
sign [+ -]
zeichen [_]
%%
"+" {printf("Operator + gefunden!\n"); }
"-" {printf("Operator - gefunden!\n"); }
"=" {printf("Zuweisung = gefunden!\n"); }
"while" {printf("Schluesselwort while gefunden!\n"); }
{sign}?{ziffer}+ {printf("Ganzzahl gefunden! Zahlenwert=%i\n",atoi(yytext)); }
{sign}?((ziffer)+"."{ziffer}*)|({ziffer}*"."{ziffer}+)) {printf("Kommazahl gefunden!
Zahlenwert=%f \n",atof(yytext)); }
{letter}({letter}|{ziffer}|{zeichen})* {printf("Bezeichner gefunden! Name=%s \n",yytext); }
%%
yywrap()
{
    return 0;
};

void main()
{
    do { yylex (); }
    while (! feof(stdin));
}
```

## Aufgabe 9

Erstellen Sie eine Scanner mittels flex, der ganze Binärzahlen der Form IOIII und Dezimalzahlen erkennt und deren Wert ausgibt.

```
%{
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
}%
ziffer [0-9]
bin [IO]
sign [+ -]
%%
{sign}?{ziffer}+ {printf("Dezimalzahl gefunden! Zahlenwert=%i \n",atoi(yytext)); }
{sign}?{bin}+ {printf("Dualzahl gefunden! Zahlenwert=%i \n",bintoi(yytext)); }
%%
int bintoi(char* dual)
{
    int i=0,dz=0,sign=1;
    while (dual[i]!='\0')
    {
        switch (dual[i++])
```

```
{
    case 'I': dz=dz*2+1;
                break;
    case 'O': dz=dz*2;
                break;
    case '-': sign=-1;
                break;
}
}
return sign*dz;
}
yywrap()
{
    return 0;
};

void main()
{
    do { yylex (); }
    while (! feof(stdin));
}
```