

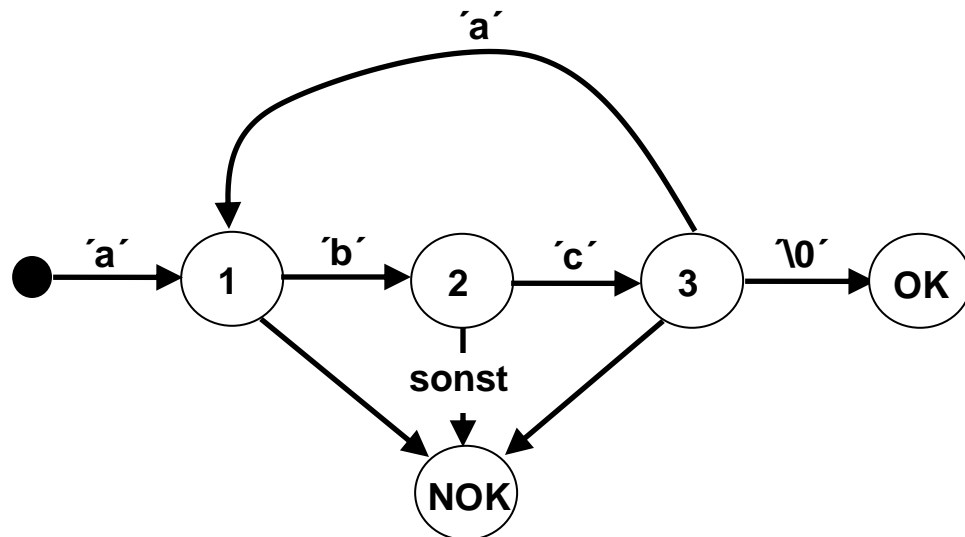
Hochschule Deggendorf Prof. Dr. Peter Jüttner	
Vorlesung: Grundlagen der Informatik	WS 2012
Übung 7	Termin 13.11.12

Endliche Automaten - Musterlösung

1. Erkennung einer Zeichenkette mittels eines Automaten

Programmieren Sie einen endlichen Automaten, der Zeichenketten akzeptiert, die aus beliebig langen Folgen der Zeichenkette „abc“ besteht, d.h. z.B. folgende Zeichenketten werden akzeptiert: „abc“, „abcabc“.

Entwerfen Sie Ihren Automaten zunächst grafisch auf Papier bevor sie ihn programmieren!



```

#include <stdio.h>
#include <stdlib.h>

int main(void)
{ typedef enum Zustaende { Start,
    a /* a gelesen */,
    b /* b gelesen */,
    c /* c gelesen */,
    OK /* Zeichenkette OK */,

```

```

        NOK /* Zeichenkette Nicht OK */};
Zustande zustand = Start;
char zeichenkette[100] = "\0";
unsigned char index = 0; /* aktueller Index in Zeichenkette */
printf("Bitte Zeichenkette eingeben (max 99 Zeichen):");
gets(zeichenkette);

while ((index <100) &&
        (zustand != OK) && (zustand != NOK))

{ switch(zustand)
  { case Start: if (zeichenkette[index] == 'a')
                zustand = a;
                else zustand = NOK;
                break;
    case a:     if (zeichenkette[index] == 'b')
                zustand = b;
                else zustand = NOK;
                break;
    case b:     if (zeichenkette[index] == 'c')
                zustand = c;
                else zustand = NOK;
                break;
    case c:     if (zeichenkette[index] == 'a')
                zustand = a;
                else if (zeichenkette[index] == '\0')
                zustand = OK;
                else zustand = NOK;
                break;
    case OK:    break;
    case NOK:   break;
    default: printf("ungültiger Zustand\n");
  };
  index++;
};
if (zustand == OK)
  printf("Zeichenkette %s OK\n",zeichenkette);
else printf("Zeichenkette %s NICHT OK\n",zeichenkette);

system("PAUSE");
return 0;
}

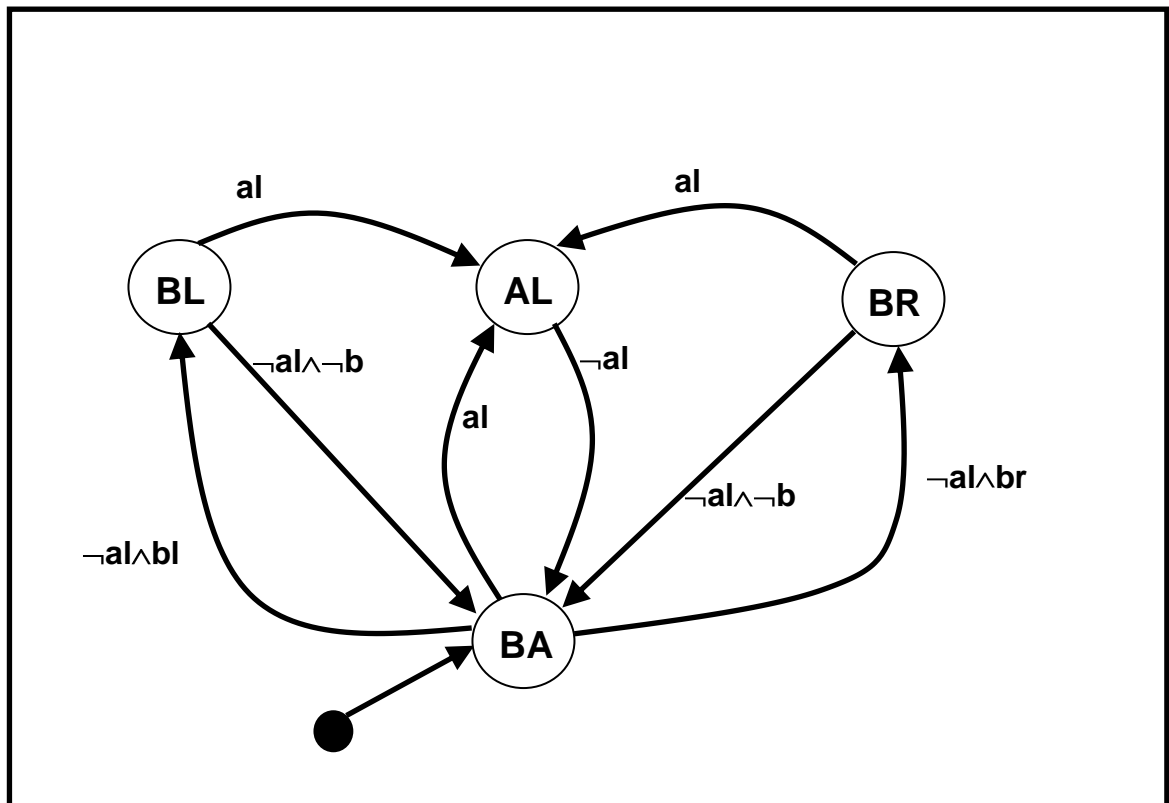
```

2. Blinkerfunktion

Die Blinkerfunktion (Blinken links, Blinken rechts und Warnblinken) in einem Kraftfahrzeug lässt sich als endlicher Automat darstellen. Programmieren Sie eine entsprechende Blinkersimulation als endlichen Automaten. Beachten Sie dabei, dass die Warnblinkfunktion höchste Priorität und einen eigenen Schalter hat.

Entwerfen Sie Ihren Automaten zunächst grafisch auf Papier bevor sie ihn

programmieren!



Bei der Implementierung unten wurde der Automat dahingehend vereinfacht, dass Blinker oder Warnblinker mit einem Ereignis ausgeschaltet werden, d.h. die in der Grafik noch unterschiedenen Ereignisse $\neg al \wedge \neg b$, $\neg al \wedge \neg b$, $\neg al$ werden zu einem Ereignis „Nicht Blinken“ zusammengefasst.

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main(void)
{ typedef enum Zustaende { BA /* Blinker und Warnblinker aus */,
    BL /* Blinken links */,
    BR /* Blinken rechts */,
    AL /* Alarmblinken */
};
```

```
typedef enum Aktionen { Bl /* Blinken links */,
    Br /* Blinken rechts */,
    Wb /* Warnblinken */,
    Nb /* kein Blinken */ };
```

```
typedef struct Tab_element
{ Zustaende fz;
    Aktionen ak;
};
```

```
typedef enum Ereignisse {al, bl, br, nb};
```

```
Tab_element z_tabelle [4][4];
```

```
z_tabelle[BA][al].fz=AL;  
z_tabelle[BA][al].ak=Wb;  
z_tabelle[BA][br].fz=BR;  
z_tabelle[BA][br].ak=Br;  
z_tabelle[BA][bl].fz=BL;  
z_tabelle[BA][bl].ak=Bl;  
z_tabelle[BA][nb].fz=BA;  
z_tabelle[BA][nb].ak=Nb;  
z_tabelle[BL][al].fz=AL;  
z_tabelle[BL][al].ak=Wb;  
z_tabelle[BL][br].fz=BL;  
z_tabelle[BL][br].ak=Bl;  
z_tabelle[BL][bl].fz=BL;  
z_tabelle[BL][bl].ak=Bl;  
z_tabelle[BL][nb].fz=BA;  
z_tabelle[BL][nb].ak=Nb;  
z_tabelle[BR][al].fz=AL;  
z_tabelle[BR][al].ak=Wb;  
z_tabelle[BR][br].fz=BR;  
z_tabelle[BR][br].ak=Br;  
z_tabelle[BR][bl].fz=BR;  
z_tabelle[BR][bl].ak=Br;  
z_tabelle[BR][nb].fz=BA;  
z_tabelle[BR][nb].ak=Nb;  
z_tabelle[AL][al].fz=AL;  
z_tabelle[AL][al].ak=Wb;  
z_tabelle[AL][br].fz=AL;  
z_tabelle[AL][br].ak=Wb;  
z_tabelle[AL][bl].fz=AL;  
z_tabelle[AL][bl].ak=Wb;  
z_tabelle[AL][nb].fz=BA;  
z_tabelle[AL][nb].ak=Nb;
```

```
Aktionen aktion = Nb;
```

```
Ereignisse ereignis = nb; /* Startzustand */
```

```
Zustaende zustand = BA;
```

```
while (1)/* Automat läuft endlos */
```

```
{  
    switch(aktion)  
    { case Bl: printf("\nEs wird links geblinkt\n"); break;  
      case Br: printf("\nEs wird rechts geblinkt\n"); break;  
      case Nb: printf("\nKein (Warn-)Blinken\n"); break;  
      case Wb: printf("\nWarnblinken\n"); break;  
      default: printf("Programmfehler 1\n"); break;  
    };  
}
```

```
printf("\nBitte Ereignis eingeben: \n");
printf("1 = Warnblinken\n");
printf("2 = Blinken links\n");
printf("3 = linken rechts\n");
printf("4 = Blinken oder Warnblinken aus\n");
printf("5 = Programmende\n");

scanf("%d",&ereignis);
if (ereignis == 5)
{ printf("\nProgrammende\n");
  getchar();getchar();
  return 0;
};
if (ereignis > 5)
printf("Programmfehler 2\n");

aktion = z_tabelle[zustand][ereignis-1].ak;
zustand = z_tabelle[zustand][ereignis-1].fz;
};

system("PAUSE");
return 0;
}
```