

Anwendungsbeispiele und Projekte im Themenbereich Reguläre Sprachen

Kerstin Strecker

Anmerkung

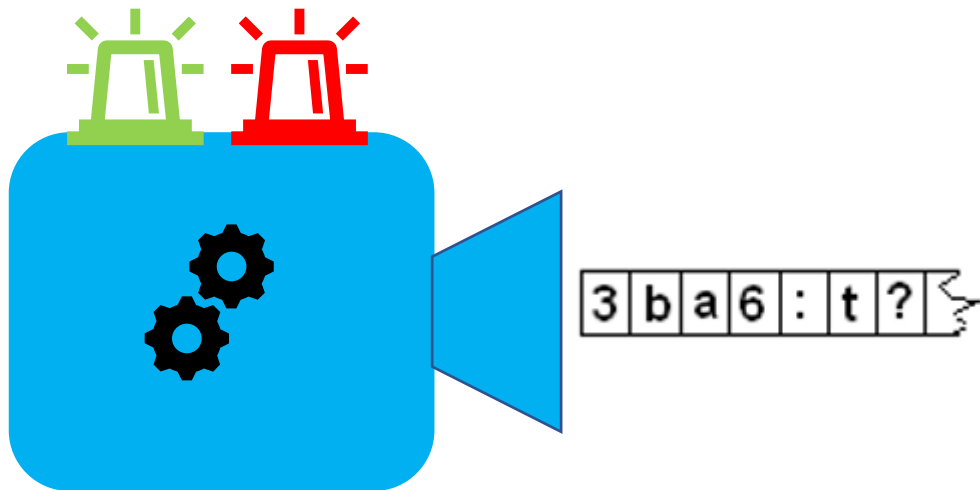
- Die Inhalte dieser Folien bringen ausschließlich Ansichten und Meinungen der Autorin zum Ausdruck. Für die korrekte Ausführbarkeit der angegebenen Beispielquelltexte dieser Folien wird keine Garantie übernommen. Auch eine Haftung für Folgeschäden, die sich aus der Anwendung der Quelltexte dieser Folien oder durch eventuelle fehlerhafte Angaben ergeben, wird keine Haftung oder juristische Verantwortung übernommen.
- Alle Programme sind nur exemplarisch zu sehen und müssen angepasst werden

Deterministische endliche Automaten

Das Wort wird zeichenweise eingelesen.

Wort abgearbeitet und der Automat im Endzustand => 

Wort abgearbeitet und der Automat nicht im Endzustand => 

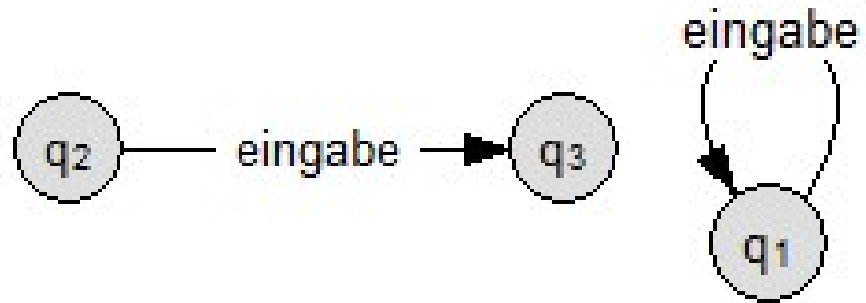


Elemente eines Zustandsgraphen (DEA)

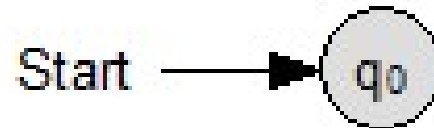
- Zustände:



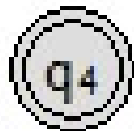
- Zustandsübergänge:



- Startzustand:

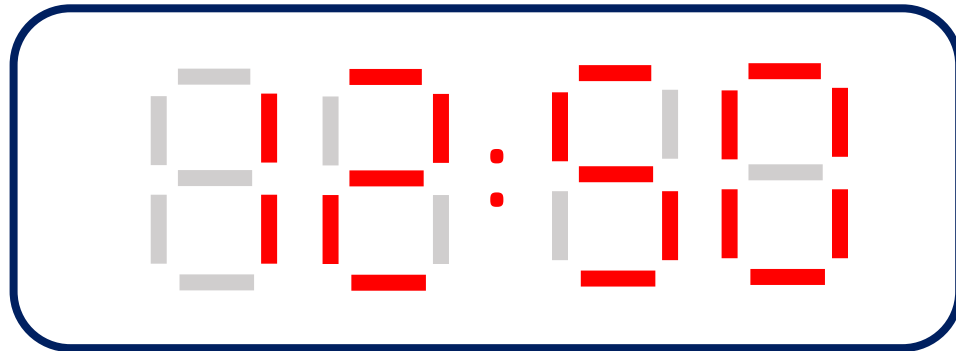


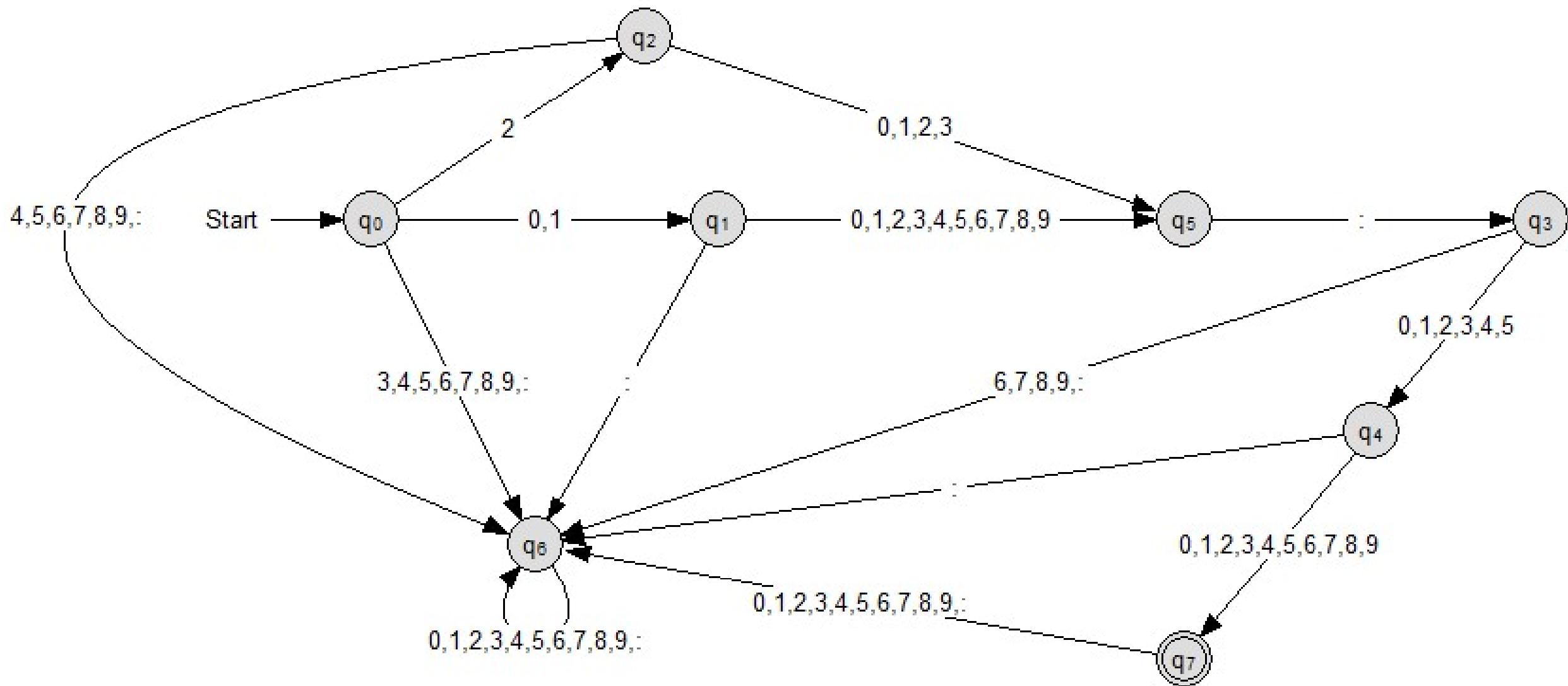
- Endzustände:



„Realitätsnahe“ Aufgaben

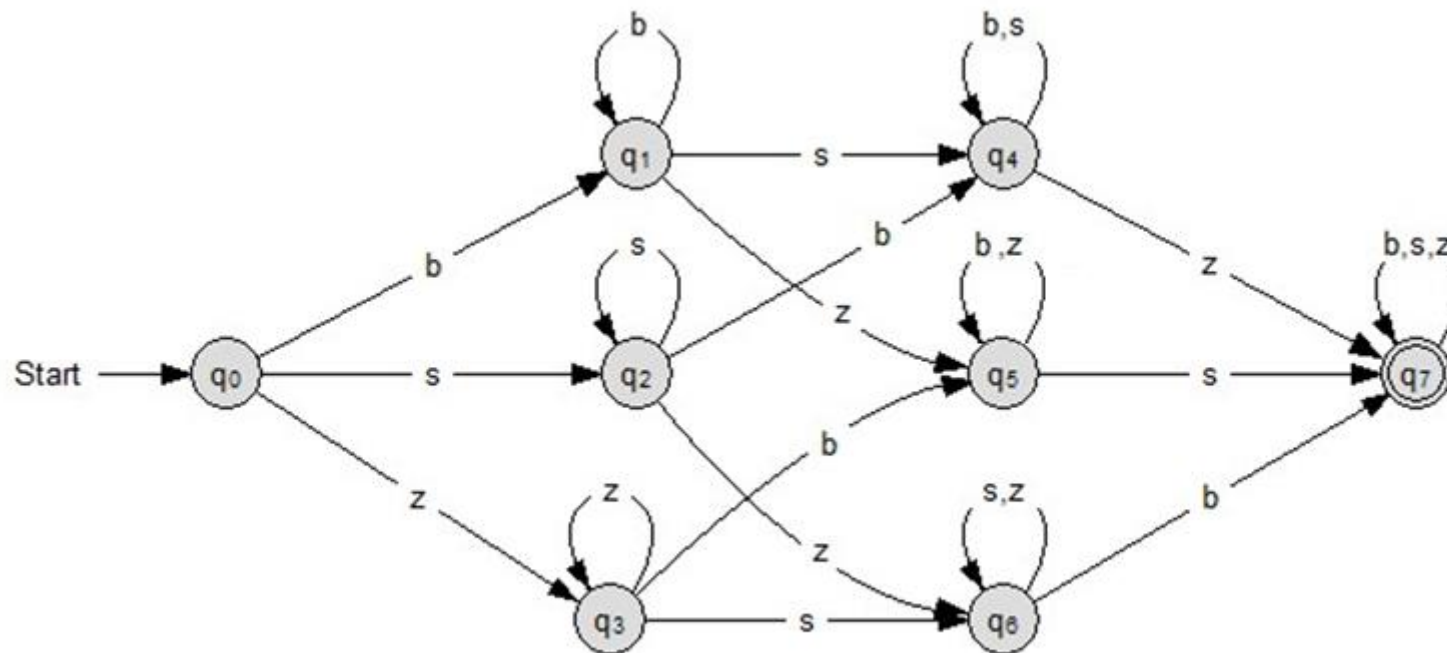
- Oft wird das Alphabet sehr groß => Komplexität
- Beispiel: Geben Sie einen endlichen Automaten an, der gültige Uhrzeiten einer Digitaluhr erkennt. Beispiel: „23:45“, „05:56“ oder „12:00“





Ausweg

- Alphabete geschickt wählen!
- Ein Automat überprüft, ob Passwörter sicher sind. Sichere Passwörter bestehen aus Buchstaben, Ziffern und Sonderzeichen. Alphabet $\{b,s,z\}$ statt $\{a,b,c,d,e,f,g,\dots,0,1,2,3,4,\dots,\$, \% , \& , ! , \dots\}$

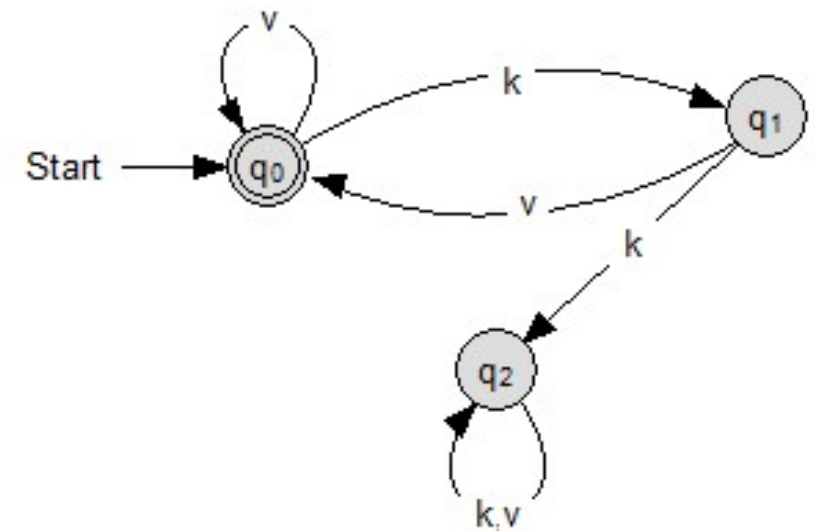


DEA in Programmcode übersetzen

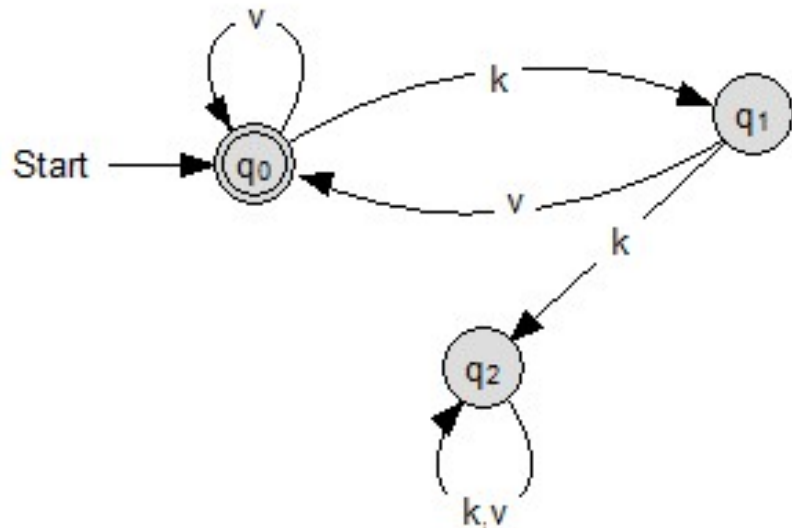
- Vorteil der deterministischen Darstellung \Leftrightarrow nichtdeterministischen Darstellung
- DEA als „sinnvolle“ Modellierung
- Konzept:
 - Variable Zustand einführen
 - Fallunterscheidung Zustand und
 - Fallunterscheidungen für jedes Zeichen des Eingabealphabets

Beispiel „Passwortgenerator“

- Der Schuladministrator lässt das Initialkennwort bei einem neuen Account zufällig generieren. Der besseren Lesbarkeit wegen hat er folgende Regel angegeben: nach einem Konsonanten muss ein Vokal folgen und darf kein zweiter Konsonant stehen.
- Geben Sie einen DEA an, der überprüft, ob es sich bei einem gegebenen Passwort um ein Initialkennwort handeln könnte.
- Alphabet $\{v,k\}$ für Vokal und Konsonant, statt $\{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z\}$



Implementierung



```
String eingabe = jTextField1.getText();
int zustand=0;
for (int i=0;i<eingabe.length();i++) {
    switch (zustand) {
        case 0: {
            if (eingabe.charAt(i)=='v') {zustand=0;};
            if (eingabe.charAt(i)=='k') {zustand=1;};
            break;}
        case 1: {
            if (eingabe.charAt(i)=='v') {zustand=0;};
            if (eingabe.charAt(i)=='k') {zustand=2;};
            break;}
        case 2: {
            if (eingabe.charAt(i)=='v') {zustand=2;};
            if (eingabe.charAt(i)=='k') {zustand=2;};
            break;}
    }
}
if (zustand==0) {jTextField1.setText("OK");}
else {jTextField1.setText(":-(");}
```

Projekt Eingabeformular

- Plausibilitätstest in Online-Formularen
- nach einer Idee von Dr. Annika Eickhoff-Schachtebeck
<https://www.uni-goettingen.de/de/629179.html> (22.5.21)

Lieferadresse

*Name:

*Vorname

*Strasse:

*Hausnummer:

*PLZ

*Ort

*Email:

Telefon:

*Passwort:

*Passwort wiederholen:

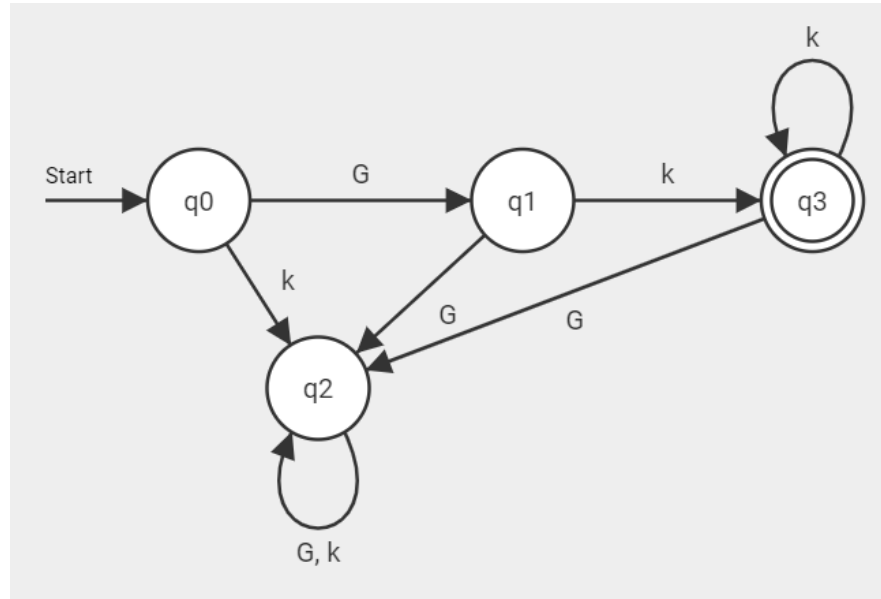
Alle mit * gekennzeichneten Felder sind Pflichtfelder

Abschicken

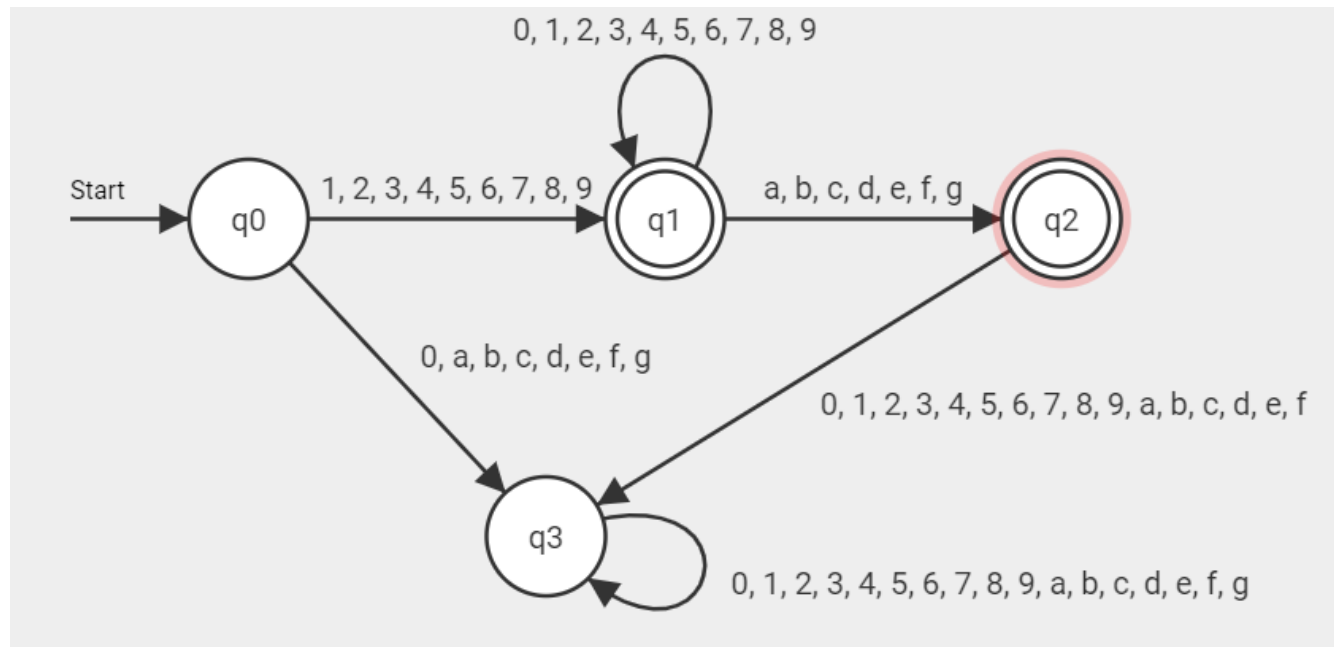
Projekt Eingabeformular

- Die Regeln erfinden die Schülerinnen und Schüler selbst (offene Aufgabe)
- Jeder Schüler / jede Schülerin entscheidet damit selbst über die Komplexität
- Beispiele:
 - Bei Nachnamen wird geprüft, ob er mit einem Großbuchstaben beginnt und danach nur Kleinbuchstaben folgen.
 - Bei Nachnamen sind auch Doppelnamen erlaubt
 - Bei Vornamen sind auch Doppelnamen erlaubt
 - Bei der email-Adresse muss genau ein @ enthalten sein
 - Eine email-Adresse muss Buchstaben vor dem @ haben und danach irgendwo einen Punkt
 - Die PLZ darf nur aus genau 5 Ziffern bestehen
 - Die Hausnummer ist eine Ziffernfolge ohne führende Nullen gefolgt von höchstens einem Kleinbuchstaben
 - ...

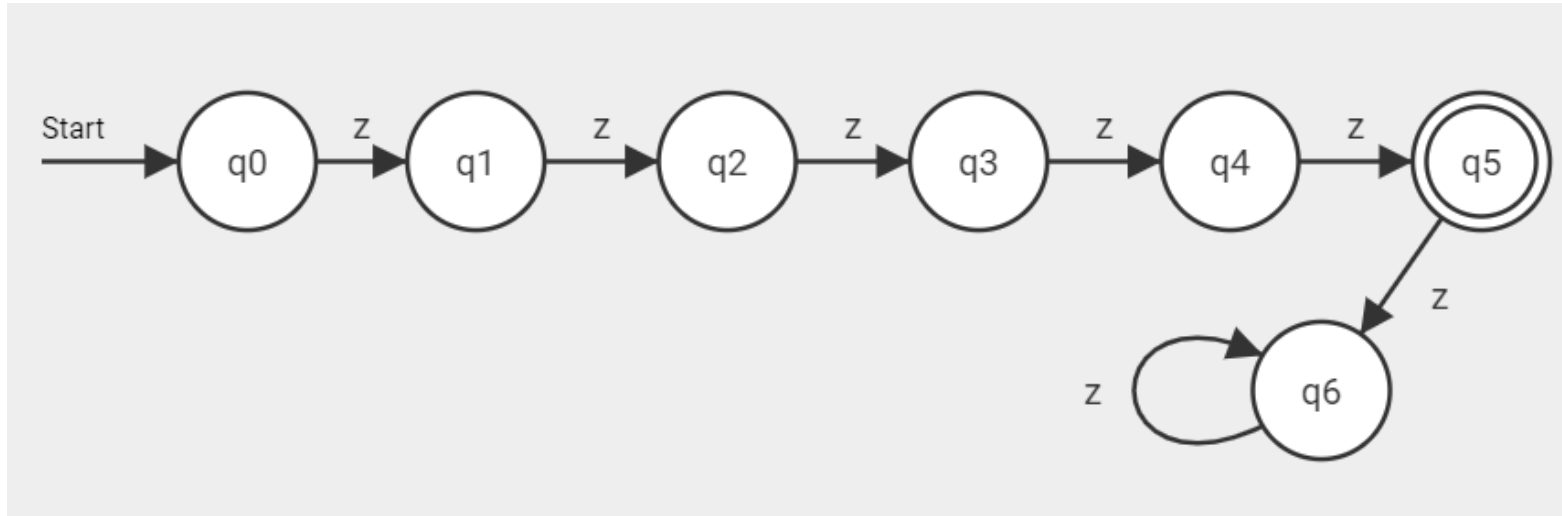
Beispiel Nachname / Vorname
(G= Großbuchstabe, k=Kleinbuchstabe):



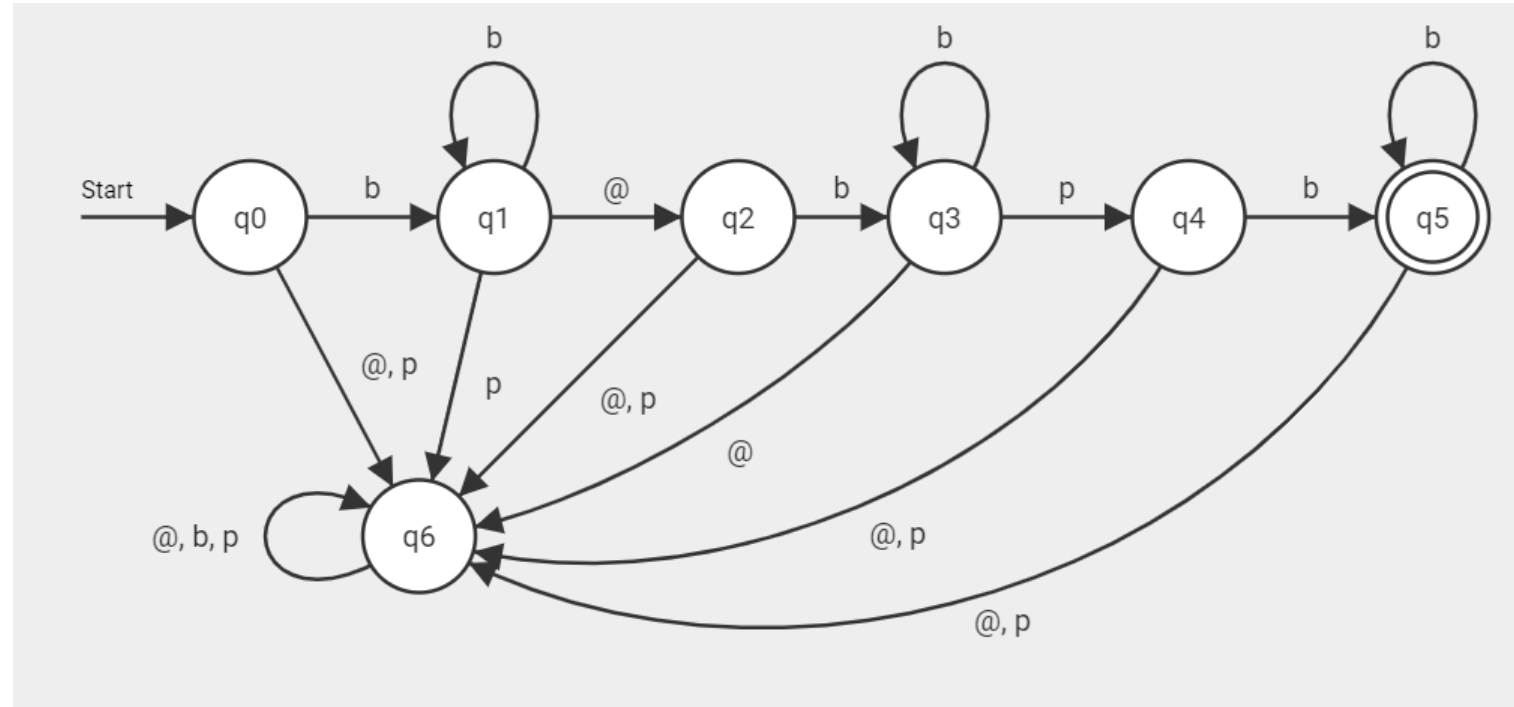
Beispiel Hausnummer:



Beispiel PLZ
(z steht für Ziffer):

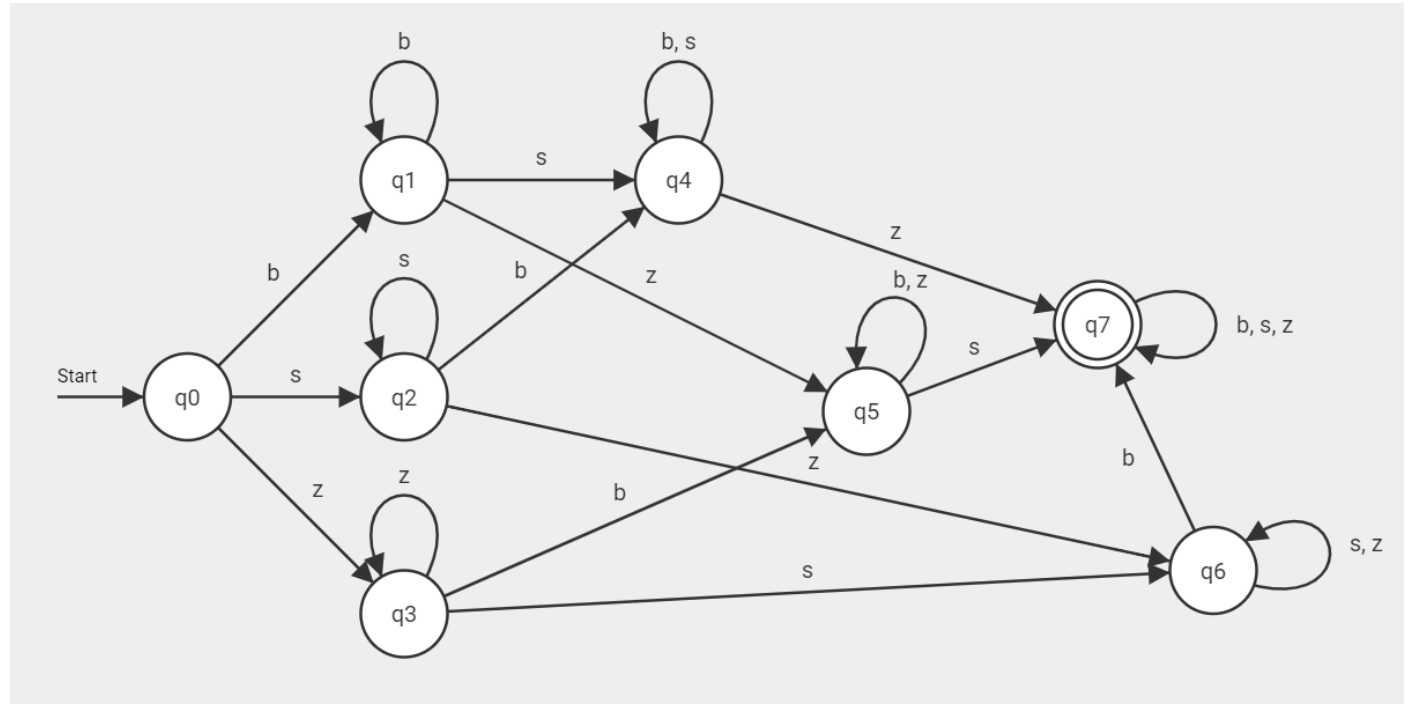


Beispiel Emailadresse
(b = Buchstabe, p=Punkt):



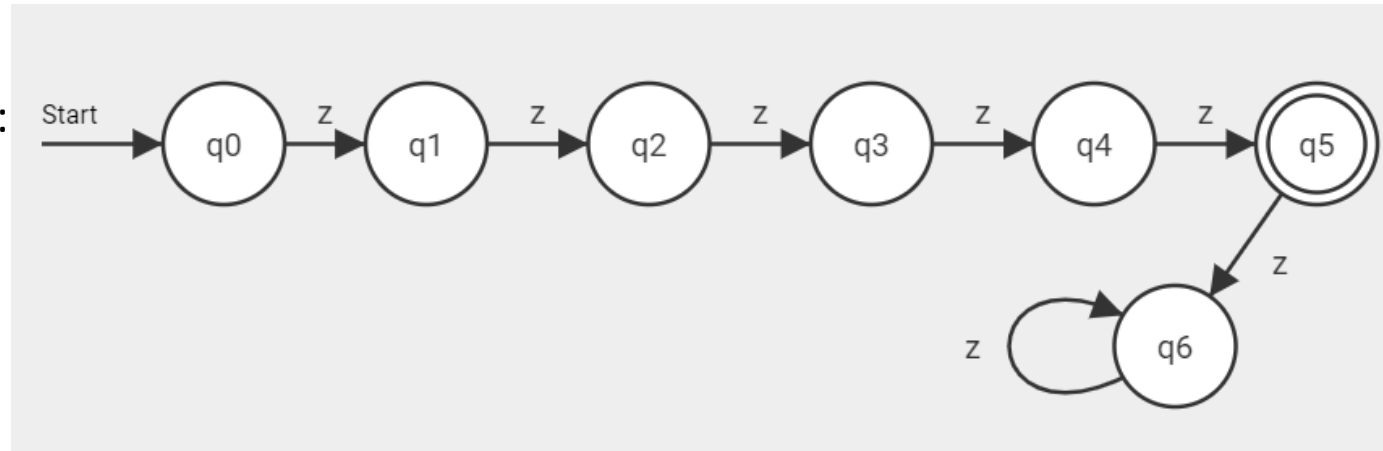
Beispiel Passwort

(b= Buchstabe, s=Sonderzeichen, z=Ziffer):



Beispiel Passwort

(z steht für Zeichen):



Implementierung

- Bei einer Implementierung kann durch Rot und Grün das Ergebnis der Plausibilitätsprüfung angezeigt werden. Zusätzlich können weitere Informationen zu den „Fehleingaben“ angezeigt werden.
- => Funktionsweise: siehe Mini-Logo-Parser (später)

Lieferadresse

*Name:	<input type="text" value="Strecker"/>		
*Vorname	<input type="text" value="Kerstin"/>		
*Strasse:	<input type="text" value="Goldschmidtstr."/>		
	*Hausnummer:	<input type="text" value="a7"/>	
*PLZ	<input type="text" value="37077"/>		
	*Ort	<input type="text" value="Göttingen"/>	
*Email:	<input type="text" value="kerstin.strecker@informatik.uni-goettingen"/>		
Telefon:	<input type="text"/>		
*Passwort:	<input type="password" value="*****"/>		
	*Passwort wiederholen:	<input type="password" value="*****"/>	

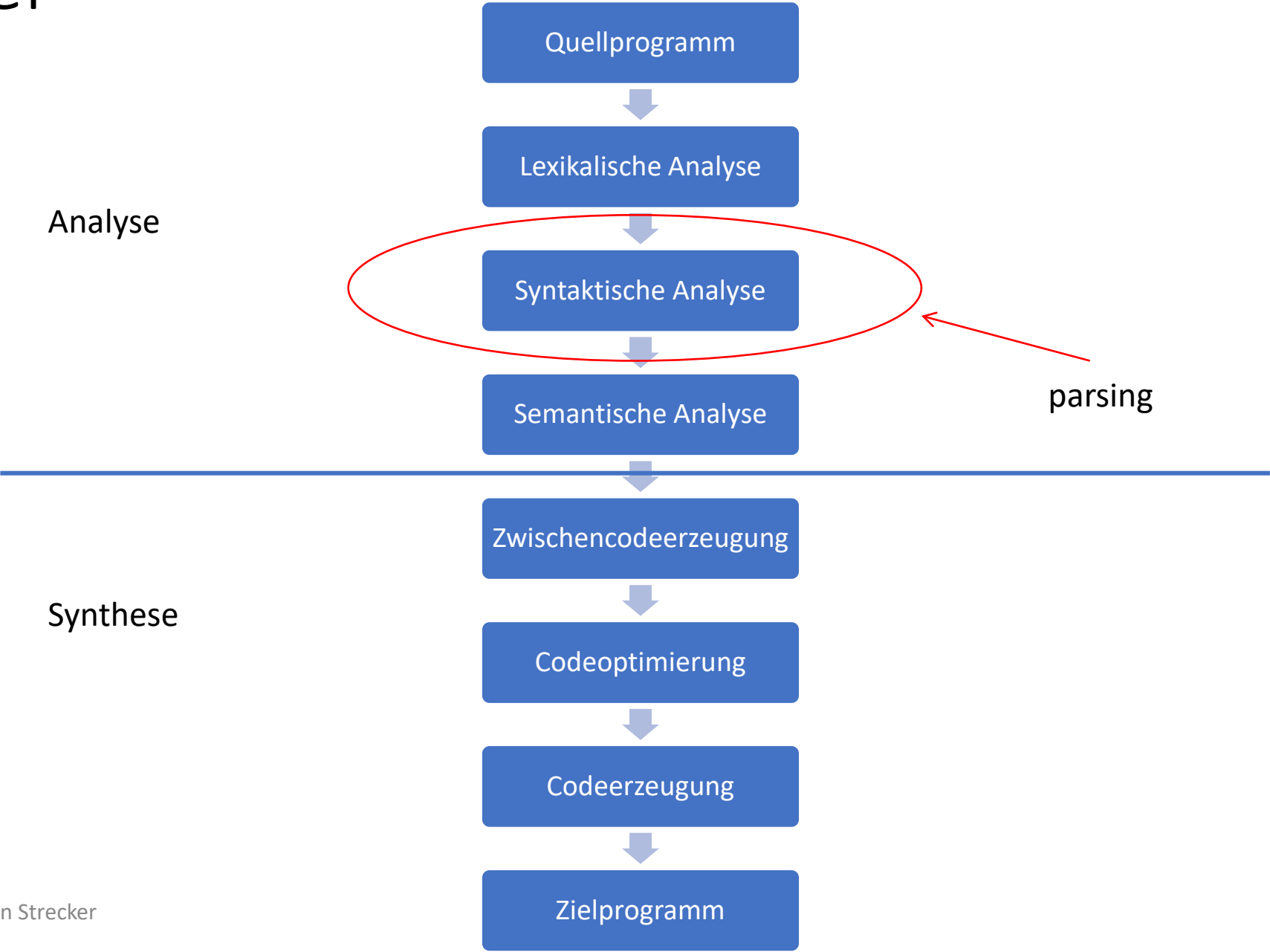
Alle mit * gekennzeichneten Felder sind Pflichtfelder

Das Feld Hausnummer enthält einen Fehler.
Zeichen 1: Ziffer erwartet

Compiler und Interpreter

- **Interpreter:** gleichzeitige Bearbeitung des Programms und der Eingabe => neue Analyse bei wiederholter Programmausführung mit anderer Eingabe
- **Compiler:** erst Programm analysieren und umwandeln ohne Eingabe, dann mit unterschiedlichen Eingaben anwenden

Compiler



Projekt Parser

- Kontextbezug:
 - Programmiersprache „LOGO“ (1980er Jahre)
 - Wir betrachten zunächst:
 - http://abz.inf.ethz.ch/wp-content/uploads/unterrichtsmaterialien/primarschulen/logo_heft_de.pdf , Zugriff: 27.1.17

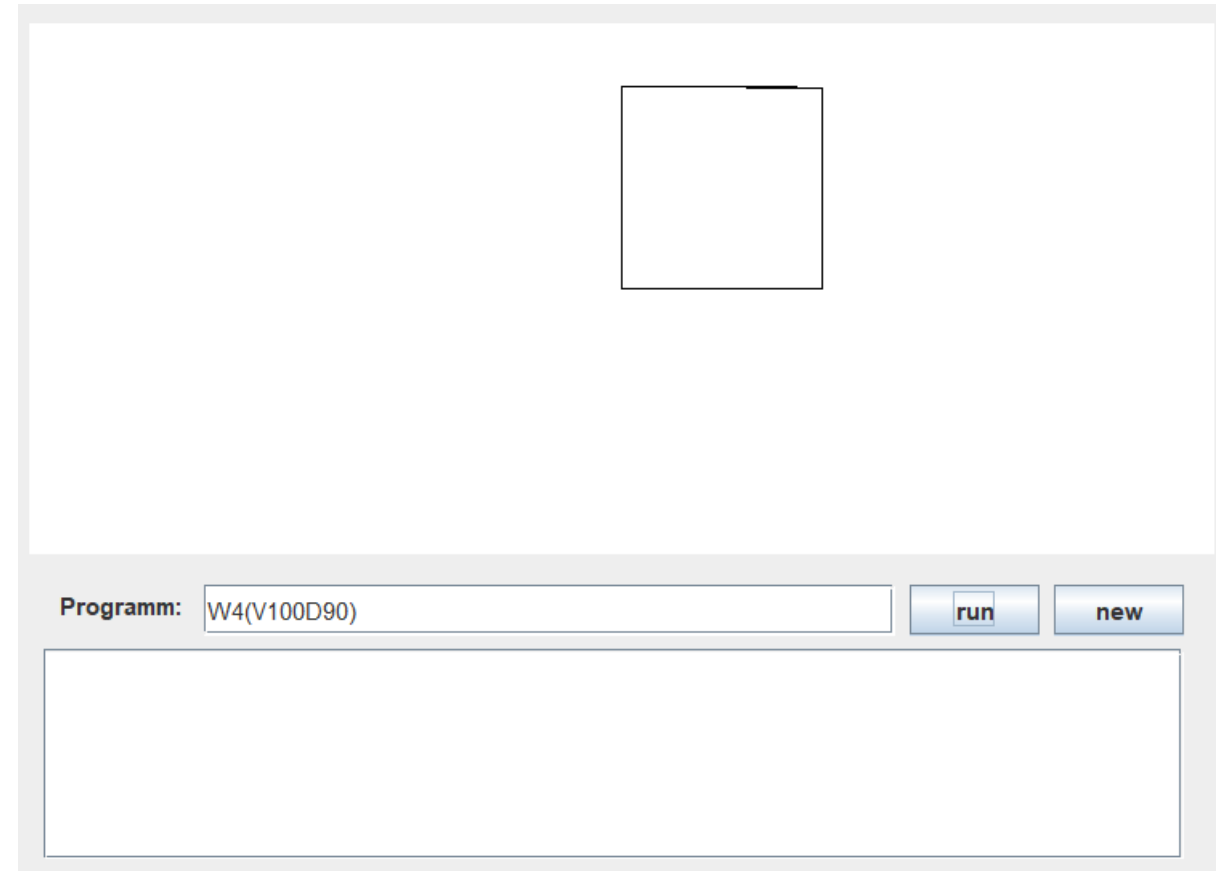
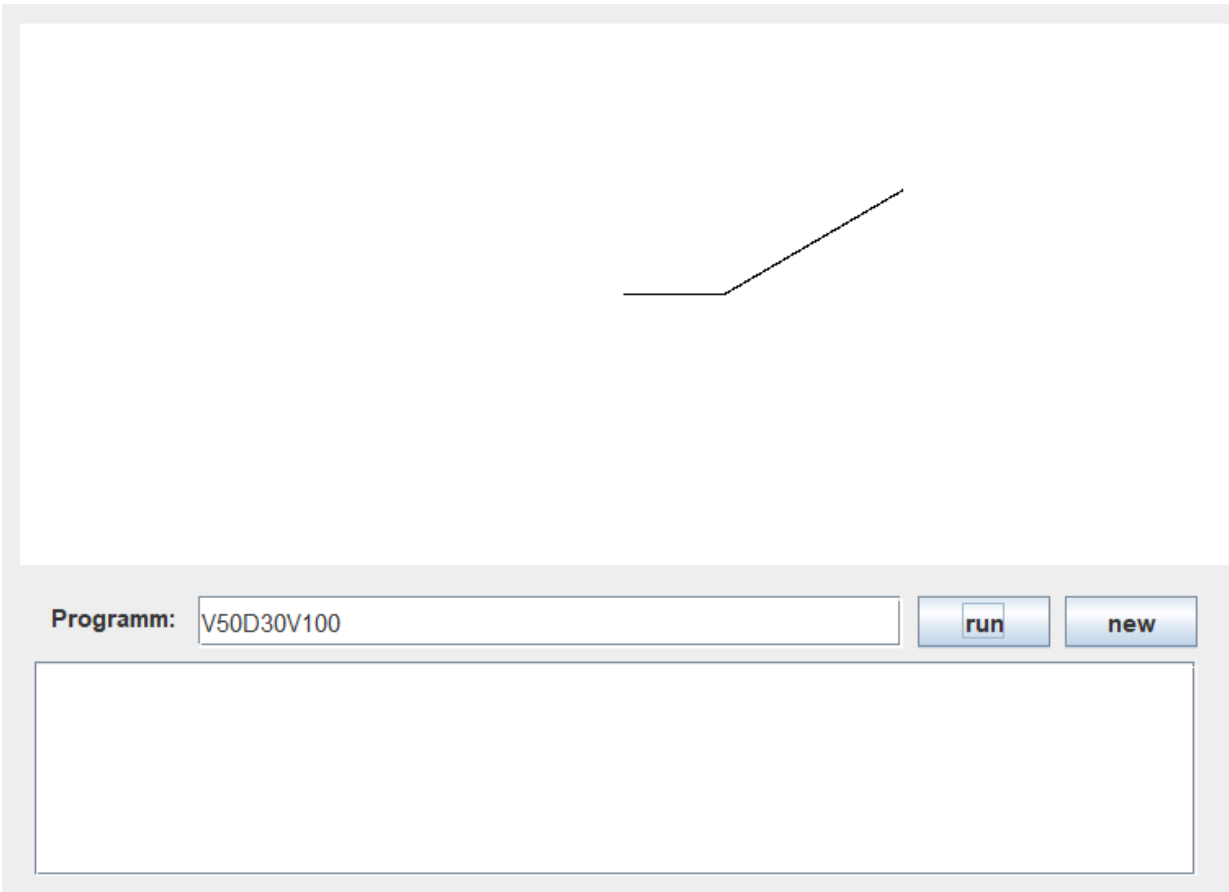
Aufgabe

- einen „Parser“ schreiben, für unser „MiniLogo“ (mit „nachgeschaltetem Interpreter“)
- Grundlage des parsens: endlicher Automat
- => keine beliebig tief geschachtelte Klammerstruktur (z.B. bei Schleifen,...)

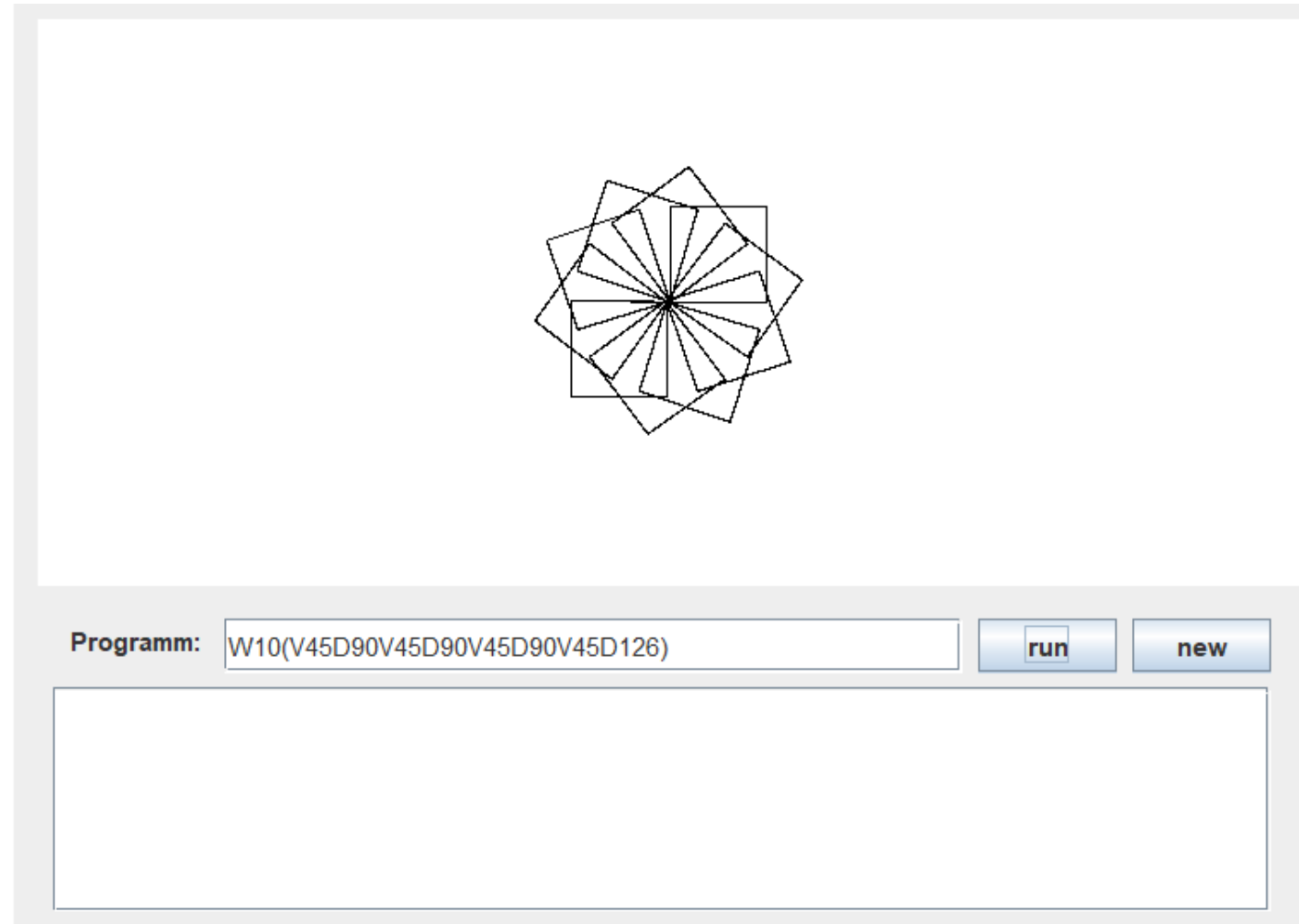
Mini-Logo

- Es gibt drei Befehle:
 - V gefolgt von einer möglichen Zahl bewirkt, dass sich die Turtle um diese Zahl nach vorne bewegt.
 - D gefolgt von einer möglichen Zahl bewirkt, dass sich die Turtle um diese Gradzahl nach rechts dreht.
 - W gefolgt von einer möglichen Zahl bewirkt, dass die Turtle die Befehle, die anschließend an die Zahl in runden Klammern eingefasst sind, so oft wie angegeben wiederholt.

Mini-Logo



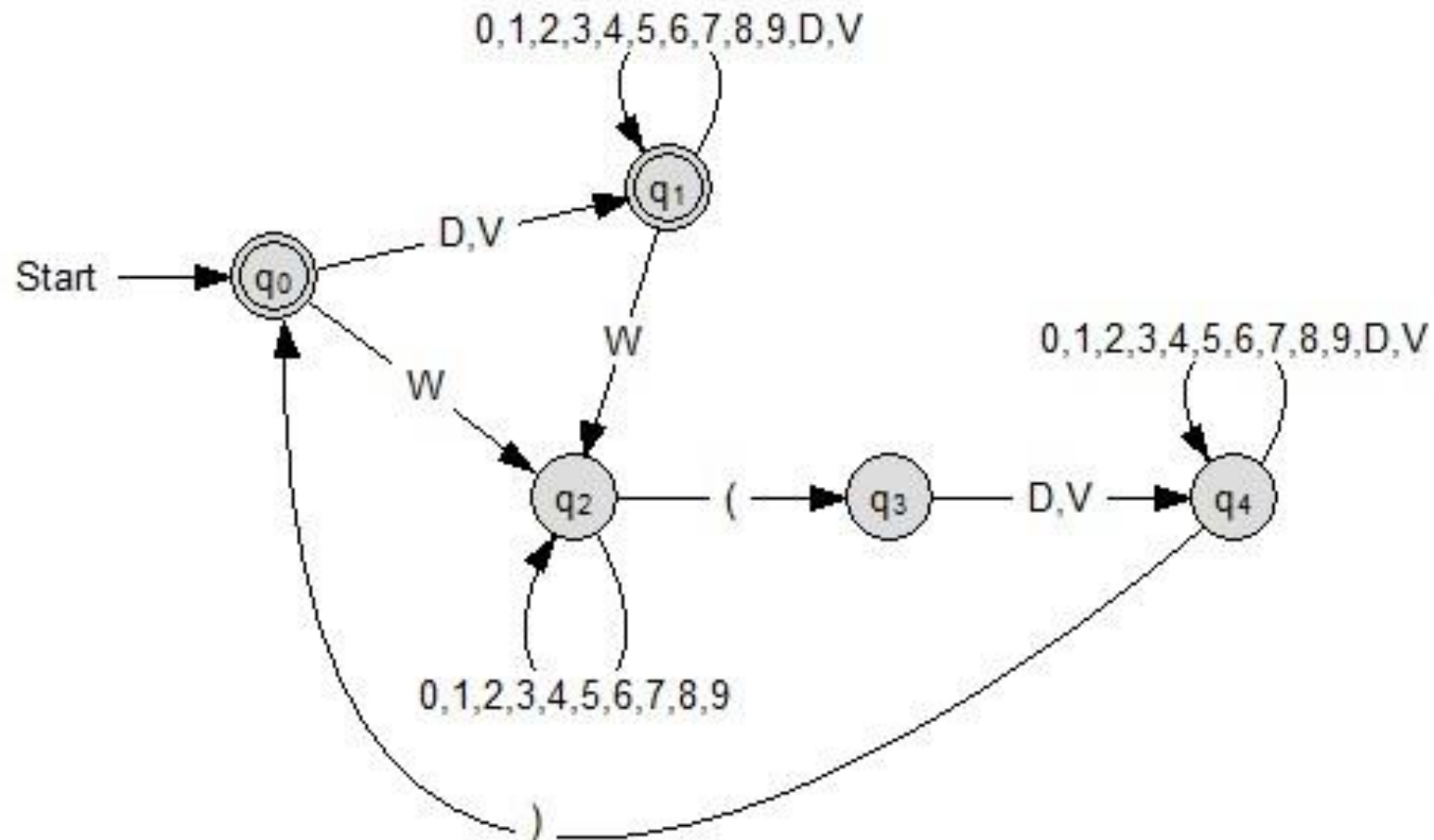
Mini-Logo



Aufbau

- zunächst ist ein Syntaxcheck notwendig, ob das eingegebene Mini-Logo-Programm syntaktisch korrekt ist

Endlicher Automat



Grammatik

- Wir verwenden die Regel: Terminale in Kleinbuchstaben, Nichtterminale in Großbuchstaben
- Alphabet = {d, v, w, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (,)}
- $S \rightarrow d \mid v \mid dA \mid vA \mid \text{EPSILON} \mid wB$
- $A \rightarrow dA \mid vA \mid 0A \mid 1A \mid \dots \mid 9A \mid d \mid v \mid 0 \mid 1 \mid \dots \mid 9 \mid wB$
- $B \rightarrow 0B \mid 1B \mid 2B \mid \dots \mid 9B \mid (C$
- $C \rightarrow dE \mid vE$
- $E \rightarrow dE \mid vE \mid 0E \mid 1E \mid \dots \mid 9E \mid) S$

Automaten in Java übersetzen

- wenn das MiniLogo-Programm abgearbeitet ist und der Automat nicht im Endzustand, kann das Programm nicht ausgeführt werden, es sollte eine Fehlermeldung geben
- wenn der Automat in einem Zustand ein Zeichen liest, für welches es keinen Zustandsübergang gibt, sollte es ebenfalls eine Fehlermeldung geben und das Programm kann nicht ausgeführt werden. Allerdings kann hier eine Fehlermeldung mit einem konkreten Hinweis ausgegeben werden.

Syntaxcheck

```
private boolean syntaxcheck(String eingabe) {
    boolean b= true;
    int zustand=0;
    for (int i=0;i<eingabe.length();i++) {
        if (b) {
            switch (zustand) {
                case 0: {
                    if (eingabe.charAt(i)=='V' || eingabe.charAt(i)=='D') {
                        zustand=1;
                    } else {
                        if (eingabe.charAt(i)=='W') {
                            zustand=2;
                        } else {
                            jTextArea1.setText("Zeichen "+ (i+1)+ ": V,D oder W erwartet" );
                            b=false;
                        }
                    }
                }
            }
        }
    };
}
```

```
break;
    case 1: { if (eingabe.charAt(i)=='V' || eingabe.charAt(i)=='D' ||
        ziffer(eingabe.charAt(i))) {
            zustand=1;
        } else {
            if (eingabe.charAt(i)=='W') {
                zustand=2;
            } else {
                jTextArea1.setText("Zeichen "+ (i+1)+ ": Ziffer oder V,D oder W erwartet" );
                b=false;
            }
        }
    };
break;
    case 2: { if (ziffer(eingabe.charAt(i))) {
        zustand=2;
    } else {
        if (eingabe.charAt(i)=='(') {
            zustand=3;
        } else {
            jTextArea1.setText("Zeichen "+ (i+1)+ ": Ziffer oder ( erwartet" );
            b=false;
        }
    }
    };
break;
    case 3: { if (eingabe.charAt(i)=='V' || eingabe.charAt(i)=='D') {
        zustand=4;
    } else {
        jTextArea1.setText("Zeichen "+ (i+1)+ ": V oder D erwartet" );
        b=false;
    }
    };
break;
```

```

        case 4: { if (eingabe.charAt(i)=='V' || eingabe.charAt(i)=='D' ||
            ziffer(eingabe.charAt(i))) {
                zustand=4;
            } else {
                if (eingabe.charAt(i)==' ') {
                    zustand=0;
                } else {
                    jTextArea1.setText("Zeichen "+ (i+1)+ ": Ziffer, V, D oder ) erwartet" );
                    b=false;
                }
            }
        };
        break;
    };
}
}
if (!(zustand==0 || zustand==1)) {
    b=false;
} // end of if
if (!b) {
    jTextArea1.setText(jTextArea1.getText() + "\n Programm kann nicht ausgeführt werden!");
} // end of if
return b;

```

```

private boolean ziffer (char c){
    boolean b=false;
    if (c=='0' || c=='1' || c=='2' || c=='3' || c=='4' ||
        c=='5' || c=='6' || c=='7' || c=='8' || c=='9') {
        b=true;
    } // end of if
    return b;
}

```

Syntaxcheck

Programm: V30D50(V40

run

Zeichen 7: Ziffer oder V,D oder W erwartet
Programm kann nicht ausgeführt werden!

Programm: 60

run

Zeichen 1: V,D oder W erwartet
Programm kann nicht ausgeführt werden!

Programm: V30W5(40D60

run

Zeichen 7: V oder D erwartet
Programm kann nicht ausgeführt werden!

Programm: V20R5(V30D45)

run

Zeichen 4: Ziffer oder V,D oder W erwartet
Programm kann nicht ausgeführt werden!

„Zielsprache“

- Turtle in der verwendeten Version des Java-Editors

```
turtle1.draw( (double) zahl);
```

```
turtle1.turn( (double) zahl);
```

- Siehe Java-Programm
- Übersetzung in Zielsprache: siehe Java-Programm

„Zielsprache“

- der vorhergehende Syntaxcheck ermöglicht, dass nur syntaktisch korrekte Programme in der „Zielsprache“ „ausgeführt“ werden.

Mini-Logo

- Aufgabe: SuS entwerfen Sie Ihr eigenes, schöneres „MiniLogo“, z.B.:
 - andere Syntaxwahl
 - ; am Ende des Befehls
 - Stiftfarbe setzen
 - links drehen
 - rückwärts
 - zwei geschachtelte Schleifen
 - if- Abfrage
 - ...

Zustandsgraphen Mealy-Maschinen

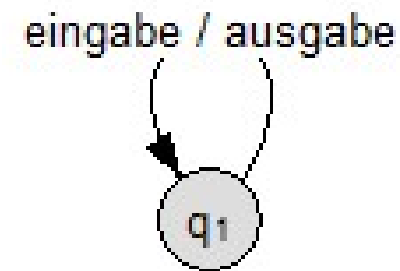
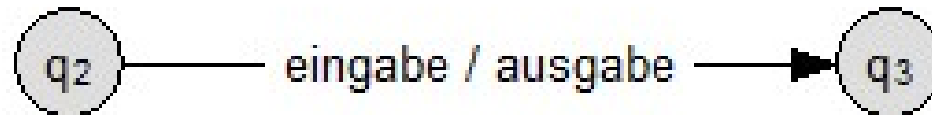
- Transduktoren: erzeugen eine Ausgabe

- Elemente:

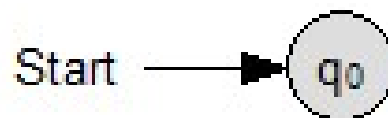
- Zustände:



- Zustandsübergänge:

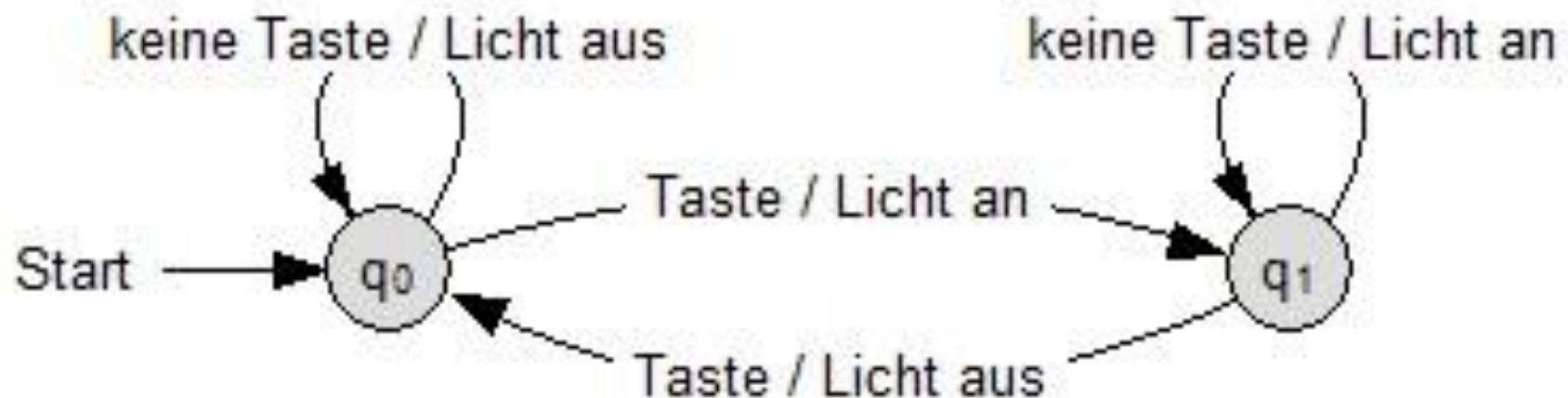


- Startzustand:



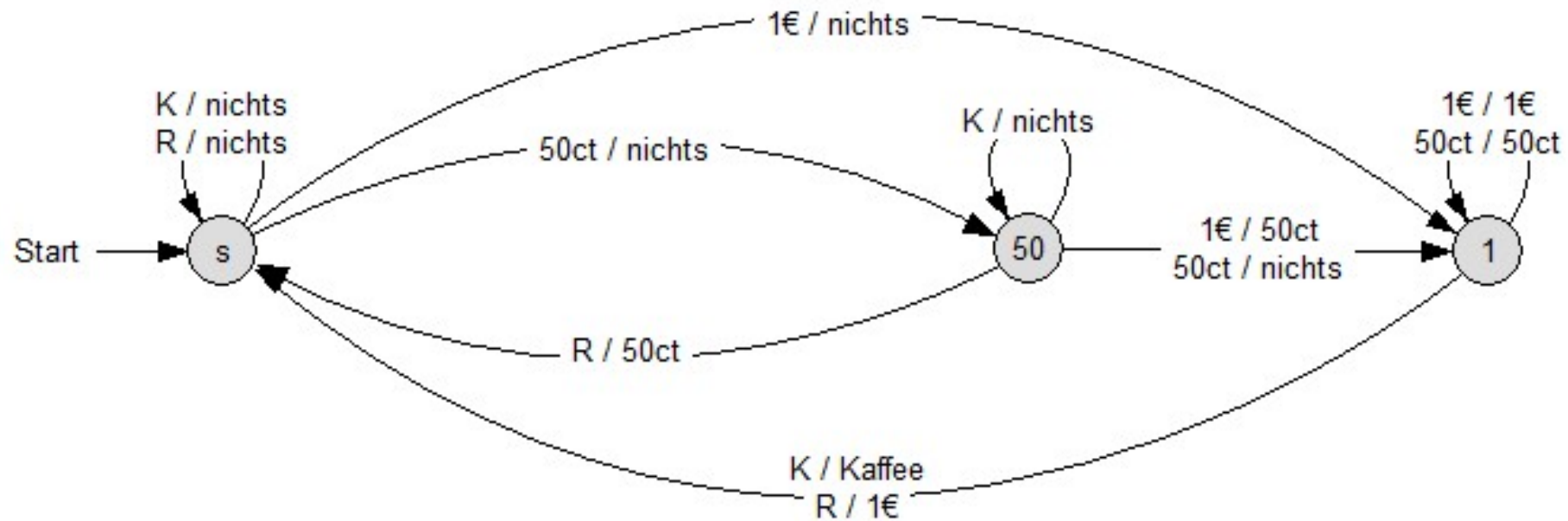
Beispiel Lichttaster

- Eingabealphabet $E = \{\text{Taste, keine Taste}\}$
- Ausgabealphabet $A = \{\text{Licht an, Licht aus}\}$



Beispiel Kaffeeautomat

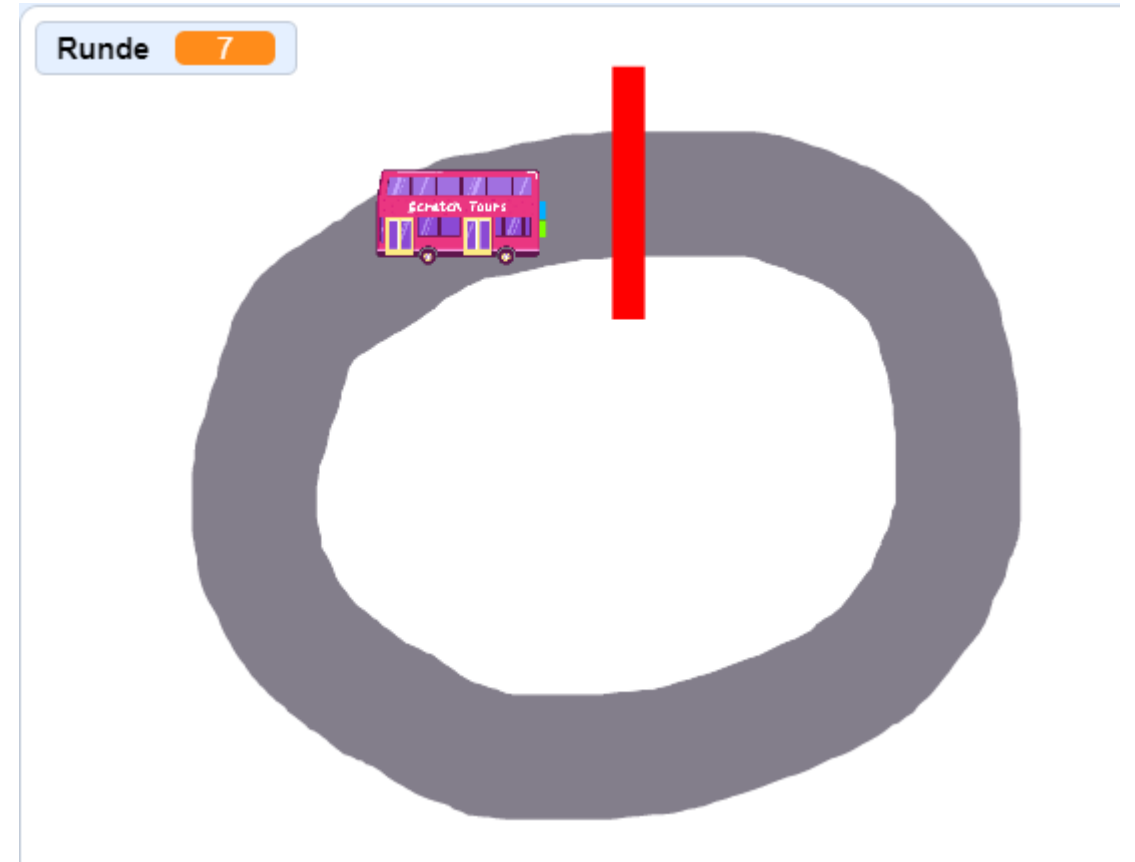
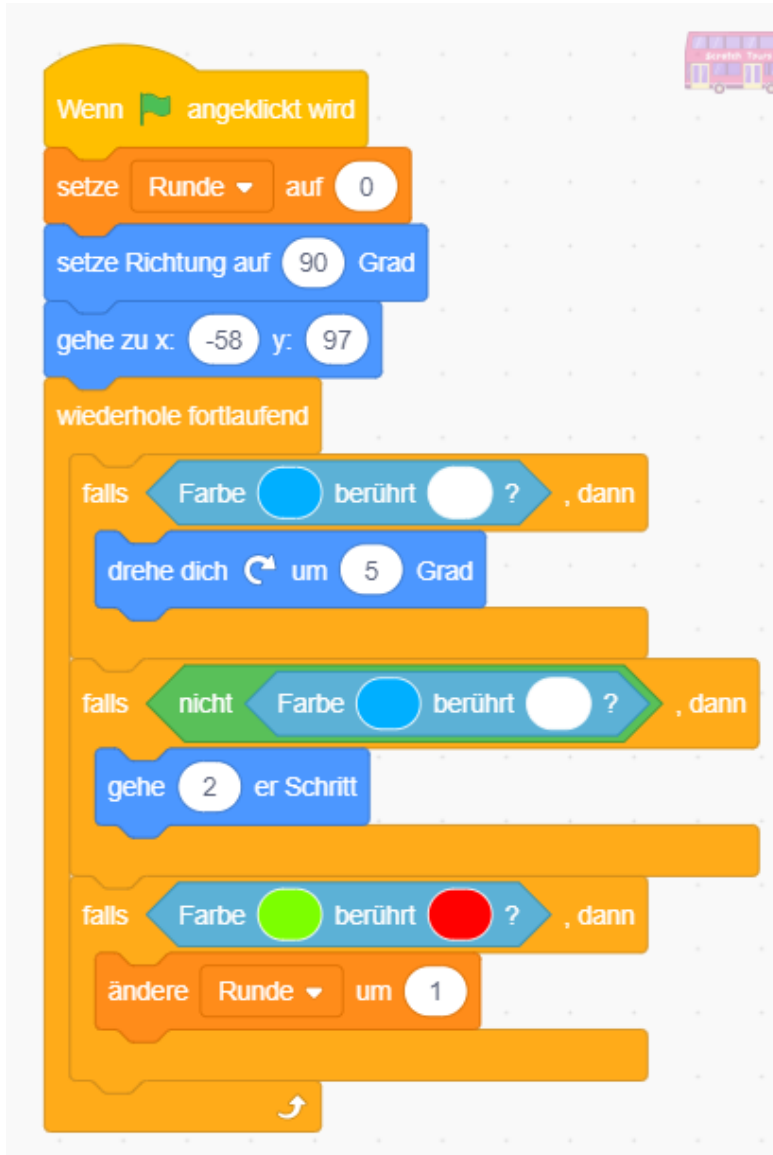
- Eingabealphabet $E = \{50\text{ct}, 1\text{€}, K, R\}$
- Ausgabealphabet $A = \{50\text{ct}, 1\text{€}, \text{Kaffee}, \text{nichts}\}$



Umsetzung Mealy-Automat in Programmcode

- z.B. „indirekt“ als „flag“
- Modellierung der Funktionalität mit Mealy-Automat => „systematische“ Umsetzung in Programmcode

Beispiel Rundenzähler

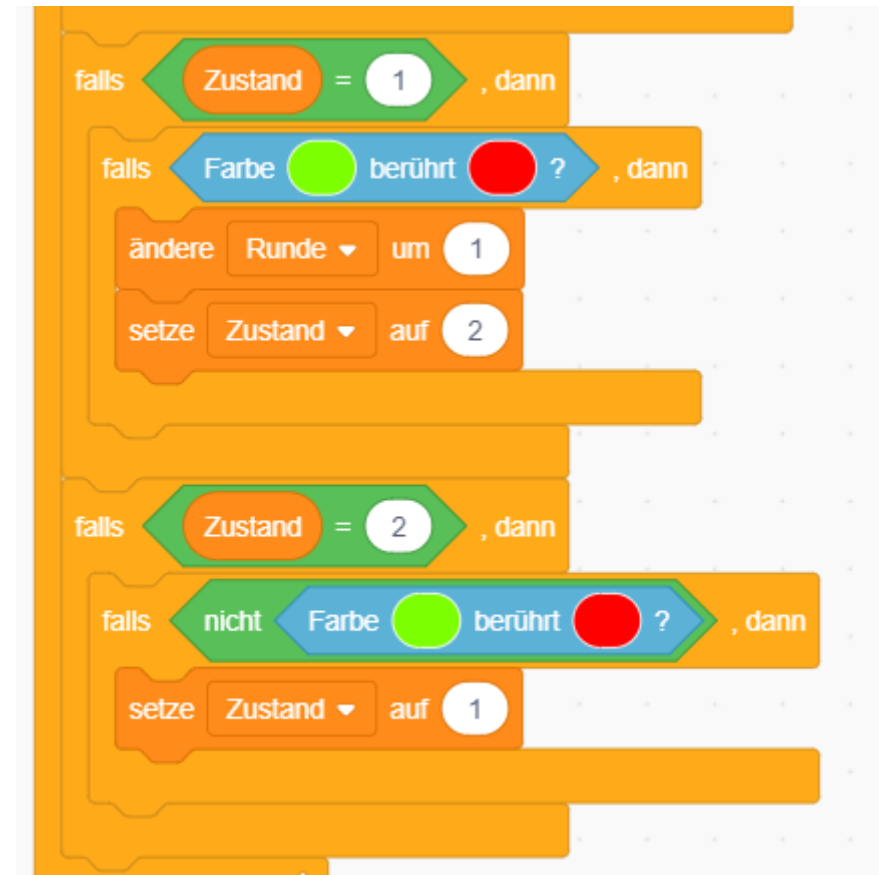


„Typisches“ Problem in der Schule:

der Bus überschreitet einmal die rote Linie => mehrere Runden werden gezählt, der Schüler möchte aber, dass nur eine Runde hochgezählt wird

Beispiel Rundenzähler

- Man braucht eine „flag“ bzw. einen zweiten Zustand



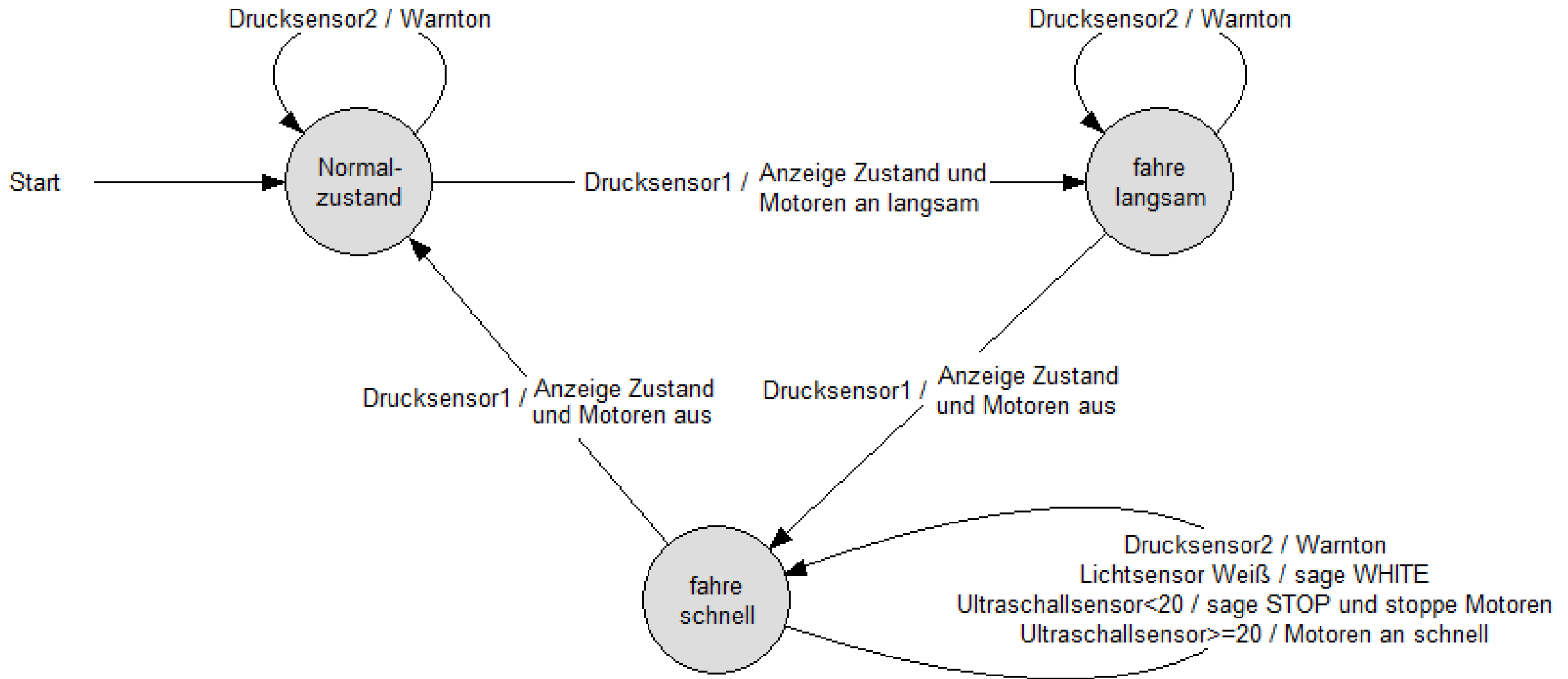
Beispiel – der automatisierte Rollator



Funktionen

Unser automatisierter Rollator kann folgende Probleme älterer Menschen beim Umgang mit Gehhilfen lösen:

- Bergauf stört das Eigengewicht des Rollators, der den Berg hinaufgeschoben werden muss, dies könnte mit einer motorisierten Unterstützung verbessert werden.
- Aufgrund unterschiedlicher Gehgeschwindigkeiten unterschiedlich Tempri einstellbar.
- Sehbeeinträchtigte Personen können vor Hindernissen gewarnt werden (z.B. unter Nutzung des Ultraschallsensors) oder z.B. auf Bahnsteigen auf die weißen Markierungen mit Hilfe entsprechender Lichtsensoren hingewiesen werden.
- Eine Hupe könnte integriert werden
- ...



beim Start

ändere zustand ▼ auf 1

show string "Normalzustand" at line 1

forever

während zustand ▼ = ▼ 1

mache wenn touch 1 ▼ is pressed dann

pause until nicht touch 1 ▼ is pressed

ändere zustand ▼ auf 2

show string "fahre langsam" at line 1

run large motor ▼ A ▼ at 5 % ⊕

run large motor ▼ D ▼ at 5 % ⊕

wenn touch 2 ▼ is pressed dann

play tone at Middle C for 1/2 ▼ beat

```

forever
  während zustand = 2
  mache
    wenn touch 1 is pressed dann
      pause until nicht touch 1 is pressed
      ändere zustand auf 3
      show string " fahre schnell " at line 1
      stop all motors
    +
    wenn touch 2 is pressed dann
      play tone at Middle C for 1/2 beat
    +
  +

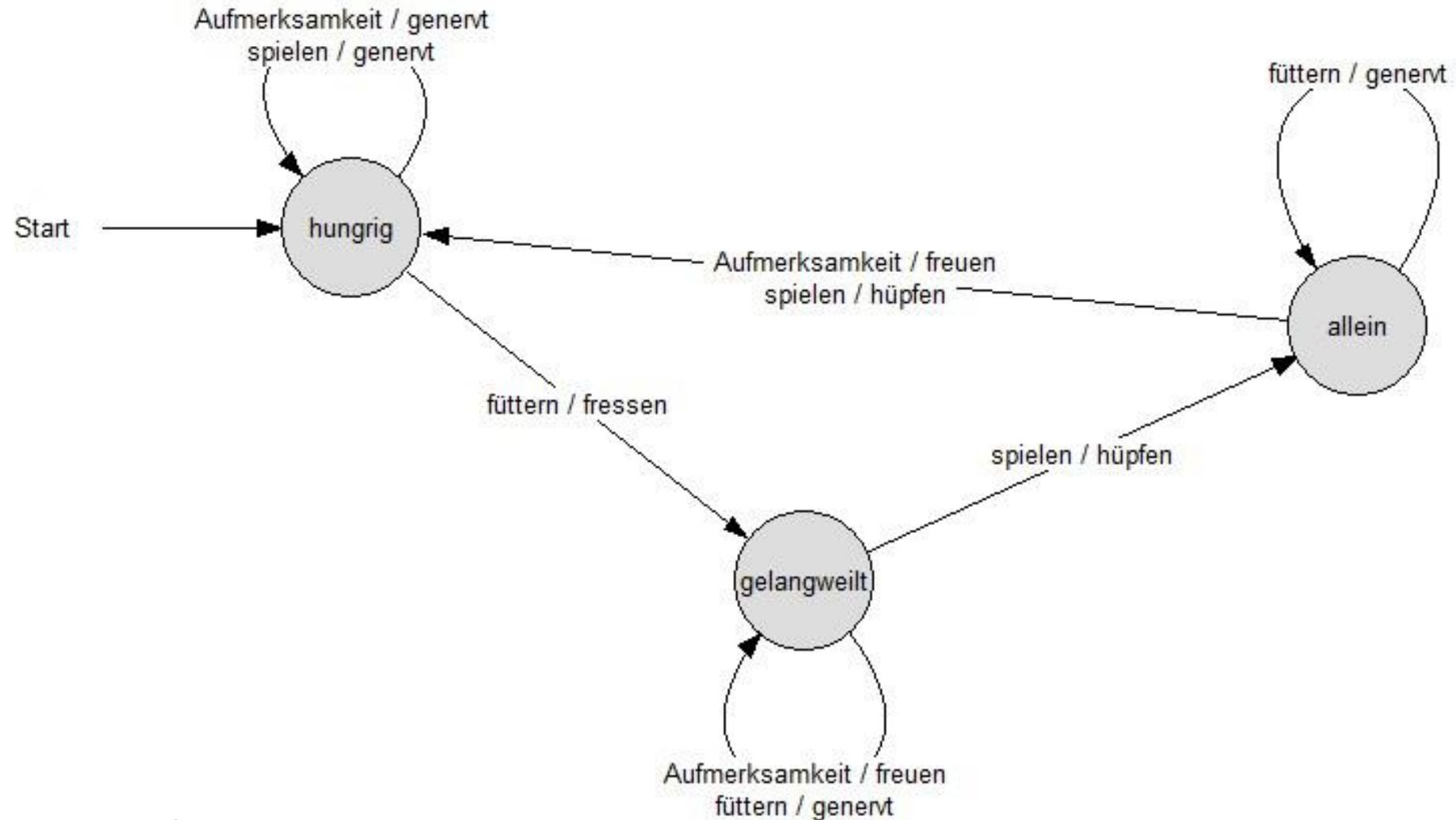
```

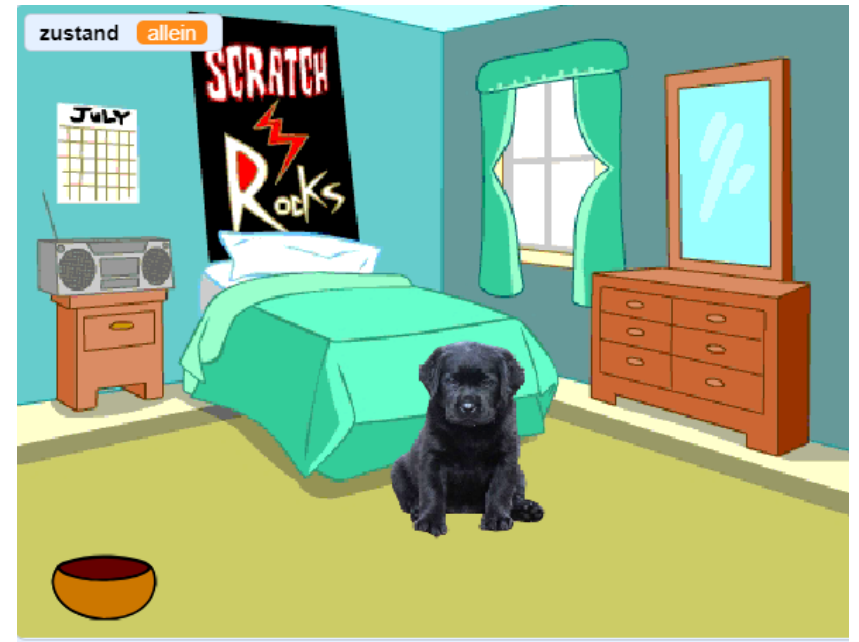
```

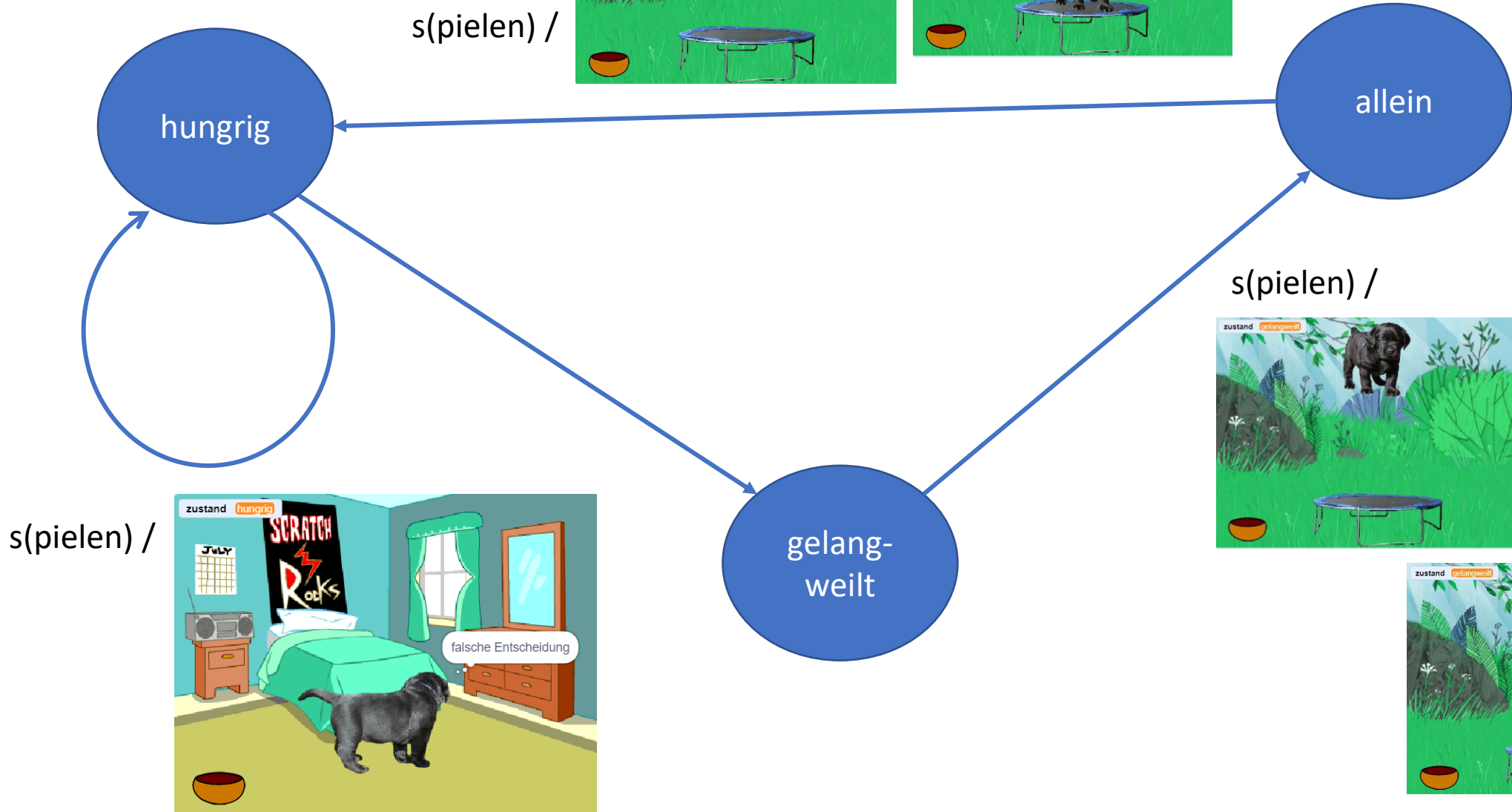
forever
  während zustand = 3
  mache
    wenn nicht ultrasonic 4 distance < 20 dann
      run large motor A at 10 %
      run large motor D at 10 %
    +
    wenn ultrasonic 4 distance < 20 dann
      play sound effect information stop until done
      stop all motors
    +
    wenn is color sensor 3 detected  dann
      play sound effect colors white until done
    +
    wenn touch 1 is pressed dann
      pause until nicht touch 1 is pressed
      ändere zustand auf 1
      show string " Normalzustand " at line 1
      stop all motors
    +
    wenn touch 2 is pressed dann
      play tone at Middle C for 1/2 beat
    +
  +

```

Beispiel „Tamagotchi“







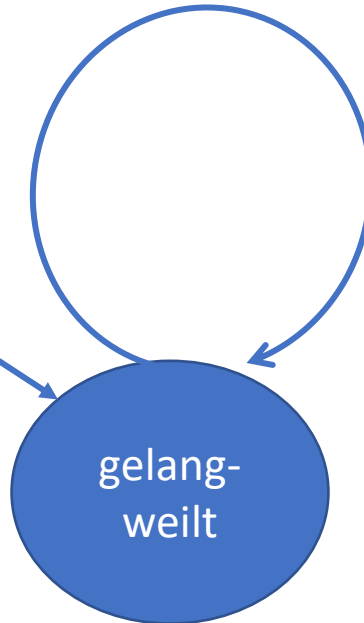
hungrig

f(üttern) /



allein

f(üttern) /



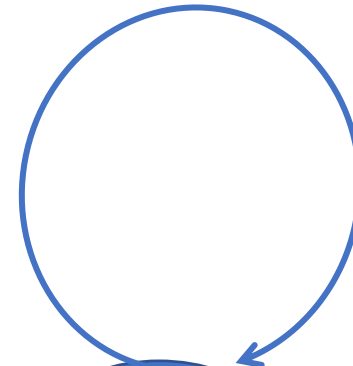
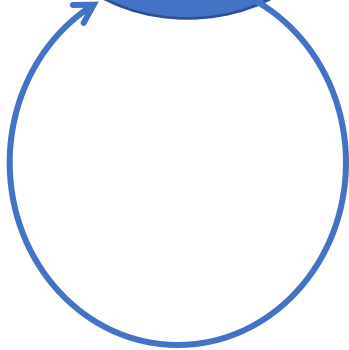
gelangweilt

f(üttern) /





a(ufermerksamkeit) /



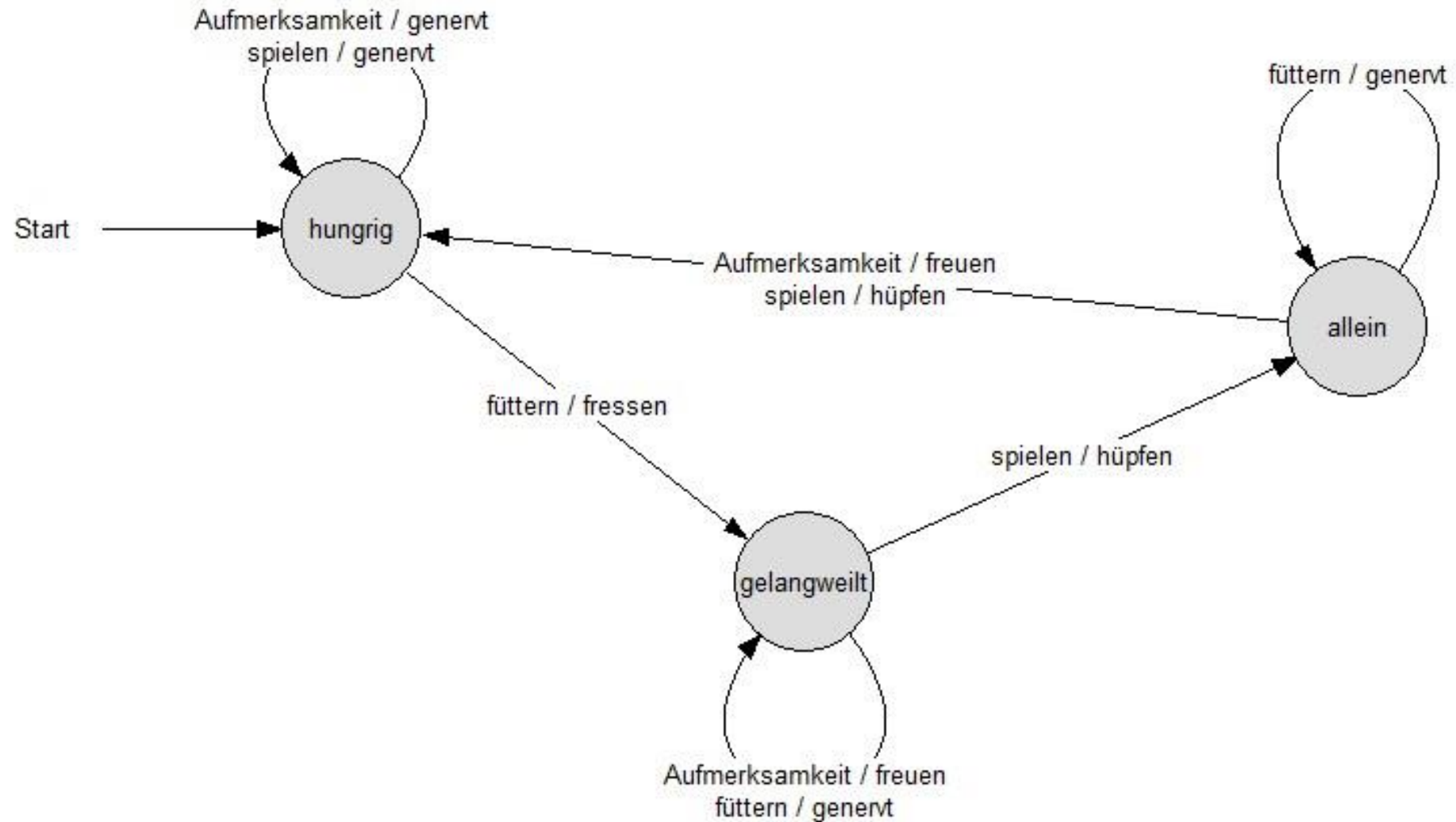
a(ufermerksamkeit) /

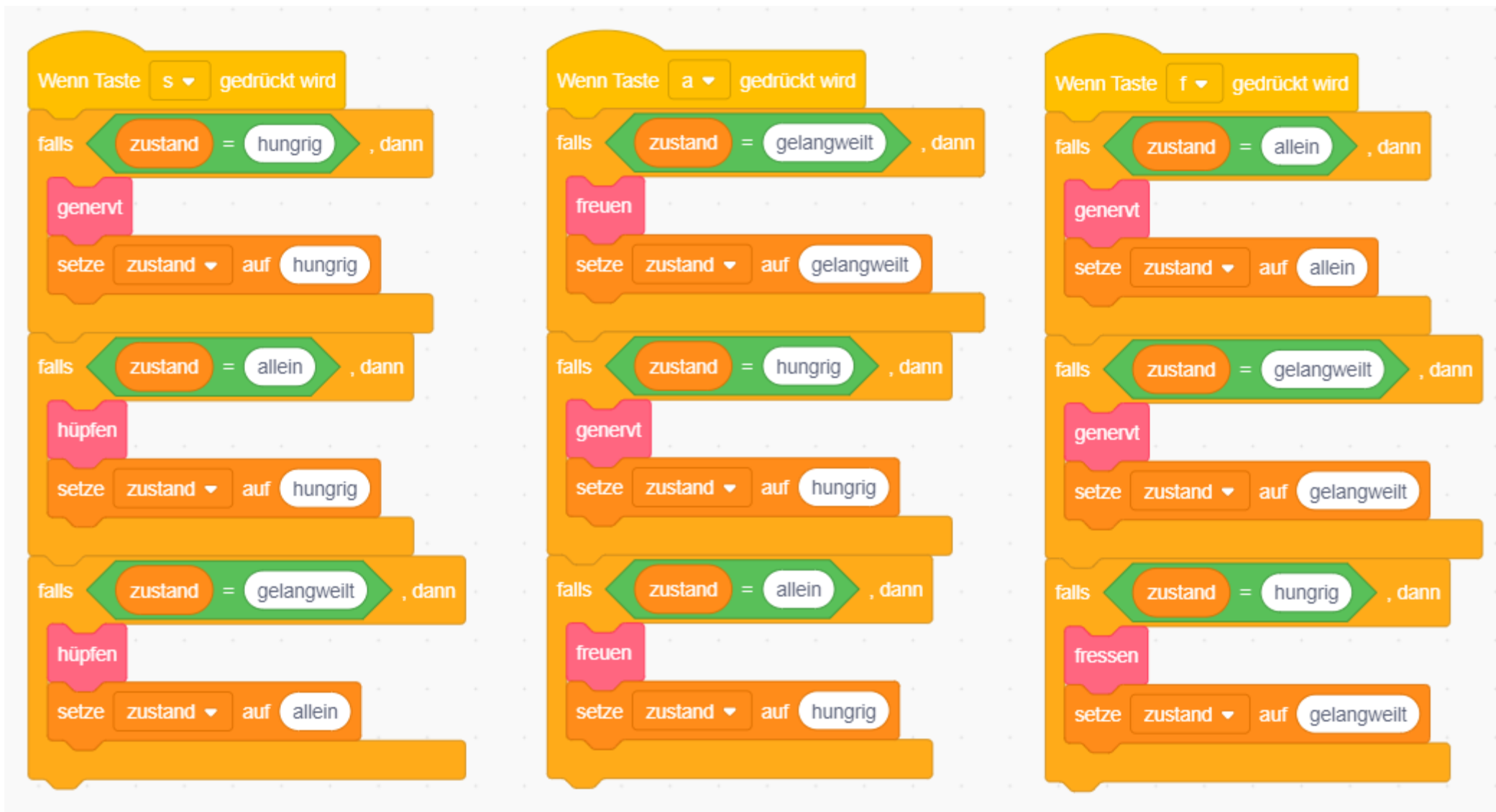


a(ufermerksamkeit) /



„Tamagotchi“





Implementierung

- Siehe Scratch3 -Programm

Grenzen endlicher Automaten

- Ein Palindrom ist ein Wort, das vorwärts und rückwärts gelesen gleich ist, z.B. „RENTNER“. Kann ein endlicher Automat angegeben werden, der Palindrome erkennt?
- Lisa und Paul streiten sich immer, wer öfter Nachtschicht gemacht hat. Deshalb führen sie Buch. Der fortlaufende Eintrag lautet LPLLLPLLPPPPLLP..... Kann ein endlicher Automat angegeben werden, der entscheiden kann, ob Lisa und Paul gleich oft Nachtschicht gemacht haben?
- Unterschied, ob das Wort/die Einträge eine feste Maximallänge haben oder nicht