Wenn es keine Übereinstimmung mit einer case-Konstanten gibt, kann default als Sprungmarke eingesetzt werden.

Gibt es einen Treffer, so werden alle folgenden Anweisungen ausgeführt, bis ein (optionales) break die Abarbeitung beendet.

Fall-Through:
Ohne break geht die Ausführung im nächsten case-Block automatisch weiter.
Falls das gewollt ist, durch Kommentar bestätigen!
It’s not a bug, it’s a feature!

Implementiere den Lachautomaten:

- Was sollte im Konstruktor festgelegt werden?
- Implementiere eine Methode
  boolean wortUntersuchen(String wort),
  die den entsprechenden Wahrheitswert zurück gibt, je nachdem, ob das Wort zur akzeptierten Sprache gehört.
  Tipps:
  wort.length() gibt die Länge des Wortes zurück.
  wort.charAt(i) gibt den i-ten Buchstaben von wort zurück.
- Implementiere eine Methode
  zustandWechseln(char eingabe)

```
public class Lachautomat{
    private int zustand;

    public Lachautomat(){
        zustand = 0; //Anfangszustand z0
    }

    public boolean wortUntersuchen(String wort){
        zustand = 0;
        boolean akzeptiert = false;
        for (int i=0;i<wort.length(); i++) {
            zustandWechseln(wort.charAt(i));
        }
        if (zustand == 3) {
            akzeptiert = true;
        }
        return akzeptiert;
    }
```

```
public void zustandWechseln(char eingabe){
    switch(zustand){
        case 0: {
            switch(eingabe) {
                case 'h': {zustand = 1;} break;
                case 'a': {zustand = 4;} break;
                case '!': {zustand = 4;} break;
            }
        } break;
        case 1: {
            switch(eingabe) {
                case 'h': {zustand = 4;} break;
                case 'a': {zustand = 2;} break;
                case '!': {zustand = 4;} break;
            }
        } break;
```

```
case 2: {
            switch(eingabe) {
                case 'h': {zustand = 1;} break;
                case 'a': {zustand = 4;} break;
                case '!': {zustand = 3;} break;
            }
        } break;
        case 3: {
            switch(eingabe) {
                case 'h': {zustand = 4;} break;
                case 'a': {zustand = 4;} break;
                case '!': {zustand = 4;} break;
            }
        }break;
```

```
        case 4: {
            switch(eingabe) {
                case 'h': {zustand = 4;} break;
                case 'a': {zustand = 4;} break;
                case '!': {zustand = 4;} break;
            }
        }break;
    }
}
```

S.32/1 ein Automat für besondere Zahlen

a) $Z = \{z0, z1, z2, z3\}$

$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$S = z0$

$E = \{z3\}$

Übergangsfunktion:

δ	z0	z1	z2	z3
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				

S.32/1 ein Automat für besondere Zahlen

a) Z = {z0, z1, z2, z3}

Σ = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

S = z0

E = {z3}

Übergangsfunktion:

δ	**z0**	**z1**	**z2**	**z3**
0	z0	z1	z2	z3
1	z1	z2	z3	z1
2	z2	z3	z1	z2
3	z3	z1	z2	z3
4	z1	z2	z3	z1
5	z2	z3	z1	z2
6	z3	z1	z2	z3
7	z1	z2	z3	z1
8	z2	z3	z1	z2
9	z3	z1	z2	z3

```
private int zustand;

  public Zahlenautomat(){
     zustand = 0; //Anfangszustand z0
  }

  public boolean wortUntersuchen(String wort){
     zustand = 0;
     boolean akzeptiert = false;
     for (int i=0;i<wort.length(); i++) {
        zustandWechseln(wort.charAt(i));
     }
     if (zustand == 3) {
        akzeptiert = true;
     }
    return akzeptiert;
  }
```

```
public void zustandWechseln(char eingabe){
        switch(zustand){
            case 0: {
                switch(eingabe) {
                    case '0': {zustand = 0;} break;
                    case '1':
                    case '4':
                    case '7': {zustand = 1;} break;
                    case '2':
                    case '5':
                    case '8': {zustand = 2;} break;
                    case '3':
                    case '6':
                    case '9': {zustand = 3;} break;
                }
            } break;
```

```
 case 1: {
   switch(eingabe) {
       case '0':
       case '3':
       case '6':
       case '9': {zustand = 1;} break;
       case '1':
       case '4':
       case '7': {zustand = 2;} break;
       case '2':
       case '5':
       case '8': {zustand = 3;} break;
   }
} break;
```

```
 case 2: {
   switch(eingabe) {
       case '0':
       case '3':
       case '6':
       case '9': {zustand = 2;} break;
       case '1':
       case '4':
       case '7': {zustand = 3;} break;
       case '2':
       case '5':
       case '8': {zustand = 1;} break;
   }
} break;
```

```
        case 3: {
          switch(eingabe) {
              case '0':
              case '3':
              case '6':
              case '9': {zustand = 3;} break;
              case '1':
              case '4':
              case '7': {zustand = 1;} break;
              case '2':
              case '5':
              case '8': {zustand = 2;} break;
          }
        } break;
    }
  }
}
```

S. 33/5 Schafkopf

a) Grammatik G = (V, Σ, P, S)

mit Σ = { 7, 8, 9, 10, Unter, Ober, König, As, Herz,
Schellen, Gras, Eichel}

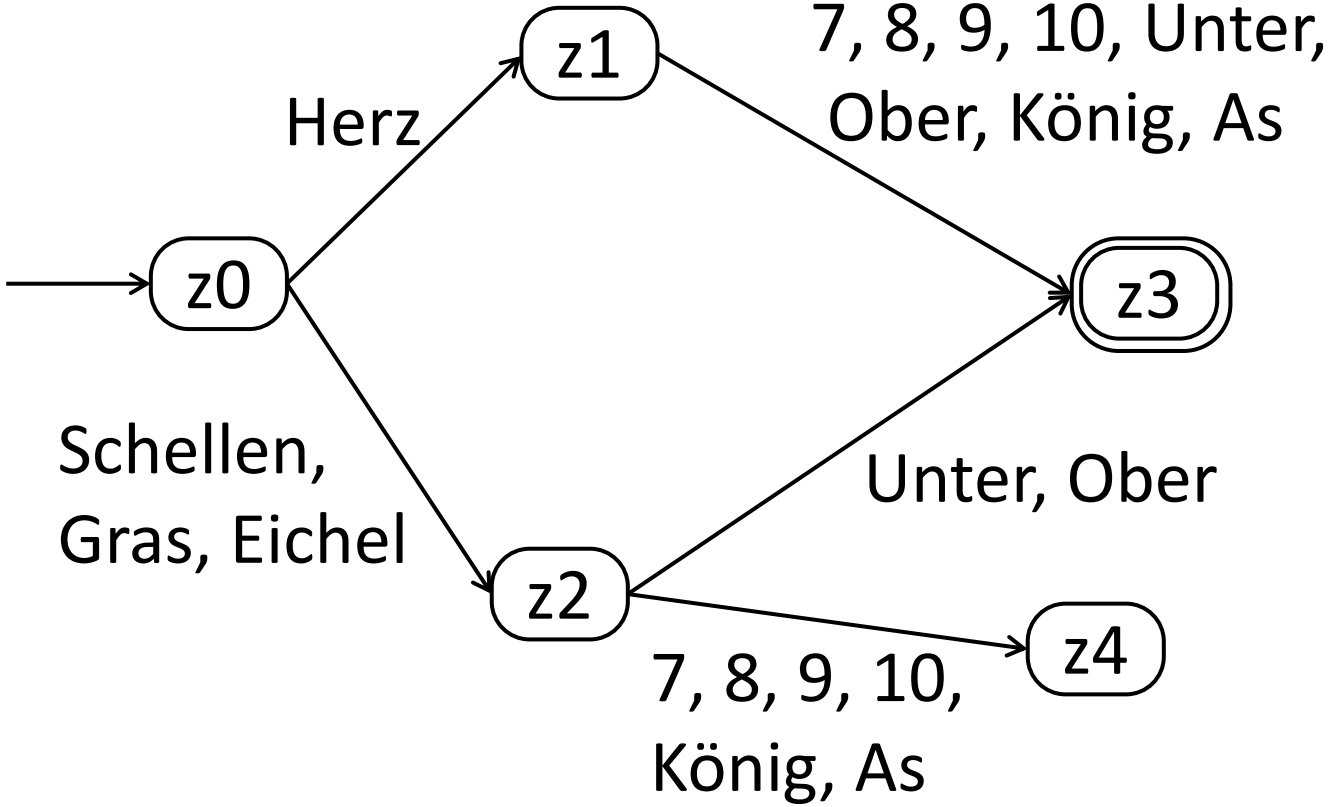
V = {<spielkarte>, <farbe>, <wert>}

P: <spielkarte> → <farbe> <wert>

<farbe> → 'Herz' | 'Schellen' | 'Gras' | 'Eichel'

<wert> → '7'|'8'|'9'|'10'|'Unter'|'Ober'|'König'|'As'

b)



c) $$\frac{\binom{4}{4}\binom{28}{4}}{\binom{32}{8}}$$

S. 33/6

Implementiere die Automaten von S. 27/1