



Name: \_\_\_\_\_

## Beispielaufgabe

### Informatik, Leistungskurs

#### Aufgabenstellung

Zur Vorbereitung auf die praktische 5000m-Laufprüfung im Abitur im Fach Sport bekommen Schülerinnen und Schüler von ihrer Sportlehrerin einen Trainingsplan, dessen einzelne Trainingseinheiten auf der Methode des Intervalltrainings beruhen. Eine Trainingseinheit enthält mehrere Intervalle. Jedes Intervall soll aus einer oder zwei Minute(n) schnellen Laufens mit gleichhoher Intensität (Belastungsphase) und einer oder mehreren Minute(n) ruhigen Laufens (Ruhephase) bestehen.

Die Sportlehrerin hat die folgenden vier möglichen Trainingsintensitäten vorgegeben:

Für die Belastungsphase sind es

1. extensiv (e) – 70 - 80% der maximalen Herzfrequenz,
2. intensiv (i) – 80 - 90% der maximalen Herzfrequenz,
3. hochintensiv (h) – 90 - 100% der maximalen Herzfrequenz.

Für die Ruhephase ist es

4. ruhig (r) – 50 - 60% der maximalen Herzfrequenz.

Damit die Sportlehrerin die von den Schülerinnen und Schülern absolvierten Trainingseinheiten schnell auf Trainingseffektivität überprüfen kann, hat sie die Informatiklehrerin der Schule gebeten, ein Überprüfungssystem zu entwickeln. Diese schlägt vor, einen endlichen Automaten zu konstruieren, der dieses leistet. Dieser endliche Automat, dargestellt in dem folgenden Zustandsübergangsgraphen, modelliert die Vorgaben der Sportlehrerin. (Dabei entspricht ein Zustandsübergang im Automat einer Länge von einer Minute im Training.)

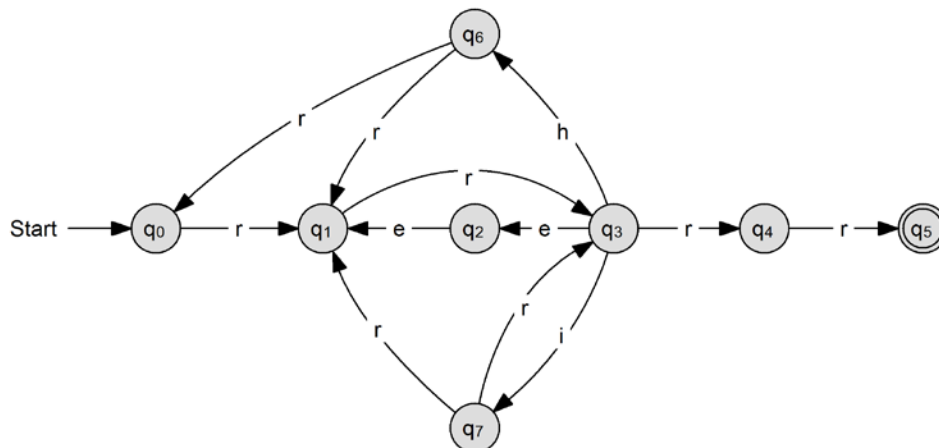


Abbildung 1: Zustandsübergangsgraph zum Prüfen eines Trainingsprogramms

Im Folgenden werden Trainingseinheiten durch Folgen ihrer Intensitätskürzel angegeben.



Name: \_\_\_\_\_

- a) Erläutern Sie, warum der endliche Automat zum Zustandsübergangsgraph in Abbildung 1 nichtdeterministisch ist.

Geben Sie für die Trainingseinheit  $rrrhrreerrr$  die Zustandsfolge des Automaten an.

Zeigen Sie, dass der Automat die Trainingseinheit  $rrhrrirhr$  nicht akzeptiert.

Beurteilen Sie, ob es im Sinne des Automaten eine genau 15 Minuten lange Trainingseinheit geben kann, in der die Belastungsphasen der Intervalle ausschließlich mit extensiver Trainingsintensität (e) gelaufen werden können.

(12 Punkte)

- b) Die Schülerinnen und Schüler sollen selbst effektive Trainingseinheiten konzipieren können.

Entwerfen Sie dazu eine reguläre Grammatik, deren Sprache genau die Wörter enthält, welche der Automat aus Abbildung 1 akzeptiert.

(8 Punkte)

Eine variable Form des Intervalltrainings soll das bisherige „klassische“ Training der Schülerinnen und Schüler ergänzen. Ein deterministischer Kellerautomat, dargestellt in dem folgenden Zustandsübergangsgraphen, modelliert die veränderten Vorgaben der Sportlehrerin.

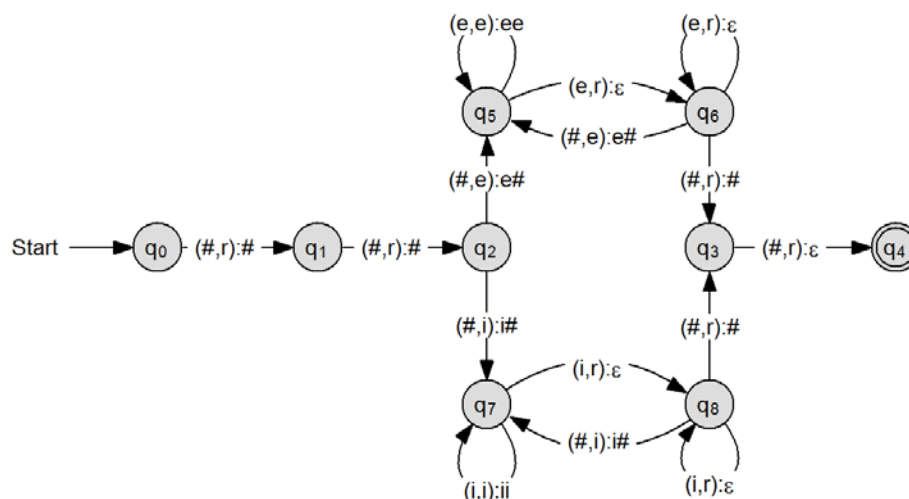


Abbildung 2: Zustandsübergangsgraph zum Prüfen eines zweiten Trainingsprogramms



Name: \_\_\_\_\_

- c) Stellen Sie für die folgende Trainingseinheit die Zustandsfolge des durch Abbildung 2 gegebenen Automaten dar.

rririirrrr

*Erläutern Sie im Sachkontext den Aufbau einer im Sinne des Zustandsübergangsgraphen aus Abbildung 2 korrekten Trainingseinheit und nennen Sie die Veränderungen gegenüber dem Aufbau einer im Sinne des Zustandsübergangsgraphen aus Abbildung 1 korrekten Trainingseinheit.*

*Erweitern Sie den Zustandsübergangsgraphen des Automaten aus Abbildung 2 in der gegebenen Anlage so, dass sich Belastungsphasen mit extensiver und intensiver Trainingsintensität abwechseln können.*

(11 Punkte)

- d) Der folgende Quelltext zeigt eine Klasse TrainingsApp mit einem Ausschnitt der Methode `pruefe`, die auf die Zustände  $q_2$  bis  $q_6$  des Automaten aus Abbildung 2 sowie deren Übergänge reduziert ist und diese simulieren soll.



Name: \_\_\_\_\_

```
1 public class TrainingsApp {
2     Stack<Character> pruefStack;

3     public TrainingsApp() {
4         pruefStack = new Stack<Character>();
5     }

6     private int zustand2(char pAktZeichen) {
7         int naechsterZustand = -1;
8         if (pruefStack.isEmpty() && pAktZeichen == 'e') {
9             pruefStack.push(new Character('e'));
10            naechsterZustand = 5;
11        }
12        return naechsterZustand;
13    }

14    private int zustand3(char pAktZeichen) {...}
15    private int zustand4(char pAktZeichen) {...}
16    private int zustand5(char pAktZeichen) {...}
17    private int zustand6(char pAktZeichen) {...}

18    public boolean pruefe(String pFolge) {
19        // Der reduzierte Automat startet im Zustand 2.
20        int zustand = 2;
21        int position = 0;
22        char aktZeichen = ' ';

23        while (zustand != -1 && position < pFolge.length()) {
24            aktZeichen = pFolge.charAt(position);
25            switch (zustand) {
26                case 2: zustand = zustand2(aktZeichen); break;
27                case 3: zustand = zustand3(aktZeichen); break;
28                case 4: zustand = zustand4(aktZeichen); break;
29                case 5: zustand = zustand5(aktZeichen); break;
30                case 6: zustand = zustand6(aktZeichen); break;
31            }
32            position = position + 1;
33        }
34        return (zustand == 4);
35    }
36 }
```

Die Methode `zustand6` (siehe Zeile 17) soll die Zustandsübergänge ausgehend vom Zustand  $q_6$  simulieren.

*Implementieren Sie die Methode `zustand6`.*

(8 Punkte)



Name: \_\_\_\_\_

e) Im Folgenden soll die Belastungs- und die dazugehörige Ruhephase eines Intervalls gleich lang sein. Die Belastungsphase setzt sich wiederum selbst wie folgt aus drei gleichlangen Blöcken zusammen: Jeder Block wird mit nur einer Trainingsintensität (also nur e oder nur i oder nur h) gelaufen. Diese drei Blöcke müssen in belastungsansteigender Reihenfolge angeordnet sein.

Ein Beispiel für ein korrektes Intervall ist: eihrrr

Die folgende Methode `trainingKorrekt` überprüft eine Trainingseinheit auf Korrektheit. Die Trainingseinheit wird als Parameter übergeben und besteht, wie beschrieben, aus Intensitätskürzeln.

```
1 public boolean trainingKorrekt(String pFolge) {
2     boolean korrekt = true;
3     int pos = 0;
4     int anzahlE, anzahlI, anzahlH, anzahl;

5     while (korrekt && pos < pFolge.length()) {
6         anzahlE = 0;
7         anzahlI = 0;
8         anzahlH = 0;

10        while (pos < pFolge.length() && pFolge.charAt(pos) == 'e') {
11            anzahlE = anzahlE + 1;
12            pos = pos + 1;
13        }

14        while (pos < pFolge.length() && pFolge.charAt(pos) == 'i') {
15            anzahlI = anzahlI + 1;
16            pos = pos + 1;
17        }

18        while (pos < pFolge.length() && pFolge.charAt(pos) == 'h') {
19            anzahlH = anzahlH + 1;
20            pos = pos + 1;
21        }

23        anzahl = anzahlE + anzahlI + anzahlH;
24        while (pos < pFolge.length() && pFolge.charAt(pos) == 'r') {
25            anzahl = anzahl - 1;
26            pos = pos + 1;
27        }

28        if (anzahlE != anzahlI || anzahlE != anzahlH || anzahl != 0) {
29            korrekt = false;
30        }
31    }
32    return korrekt;
33 }
```



Name: \_\_\_\_\_

*Zeigen Sie anhand des Quelltextes, dass die folgende Trainingseinheit korrekt ist:*

eeeiihhrrrrrrrrrr

*Zeigen Sie anhand des Quelltextes, dass die folgende Trainingseinheit nicht korrekt ist und geben Sie konkret an, welche Teile der Bedingung in Zeile 28 erfüllt und welche nicht erfüllt sind.*

eeihrrrrrr

*Entwickeln Sie zwei Folgen von Intensitätskürzeln, die unterschiedliche Teile der Bedingung nicht erfüllen.*

(10 Punkte)

**Zugelassene Hilfsmittel:**

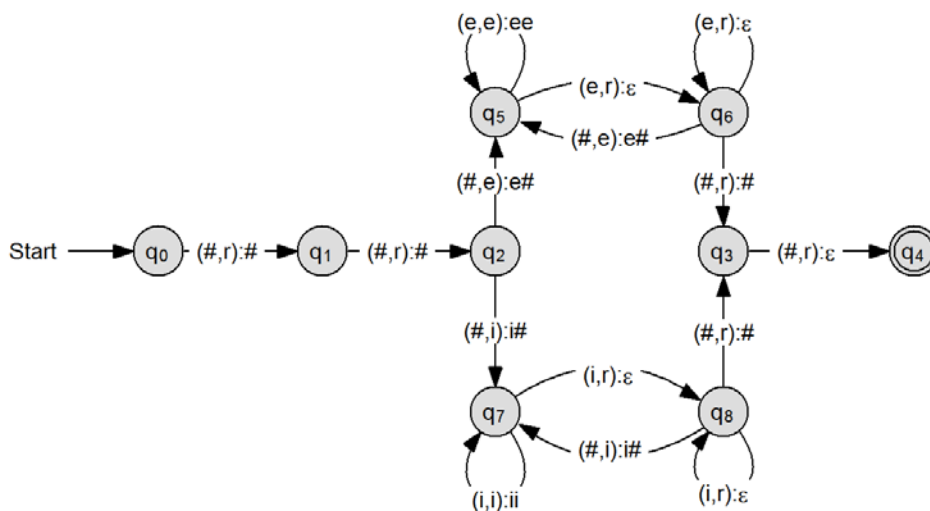
- Wörterbuch zur deutschen Rechtschreibung
- Taschenrechner (grafikfähiger Taschenrechner / CAS-Rechner)



Name: \_\_\_\_\_

**Anlage zu Aufgabe c)**

Zu erweiternder Automat:





Name: \_\_\_\_\_

## Anhang:

### Die generische Klasse `Stack<ContentType>`

Objekte der generischen Klasse **Stack** (Keller, Stapel) verwalten beliebige Objekte vom Typ `ContentType` nach dem Last-In-First-Out-Prinzip, d.h., das zuletzt abgelegte Objekt wird als erstes wieder entnommen. Alle Methoden haben eine konstante Laufzeit, unabhängig von der Anzahl der verwalteten Objekte.

### Dokumentation der generischen Klasse `Stack<ContentType>`

#### `Stack()`

Ein leerer Stapel wird erzeugt. Objekte, die in diesem Stapel verwaltet werden, müssen vom Typ `ContentType` sein.

#### `boolean isEmpty()`

Die Anfrage liefert den Wert `true`, wenn der Stapel keine Objekte enthält, sonst liefert sie den Wert `false`.

#### `void push(ContentType pContent)`

Das Objekt `pContent` wird oben auf den Stapel gelegt. Falls `pContent` gleich `null` ist, bleibt der Stapel unverändert.

#### `void pop()`

Das zuletzt eingefügte Objekt wird von dem Stapel entfernt. Falls der Stapel leer ist, bleibt er unverändert.

#### `ContentType top()`

Die Anfrage liefert das oberste Stapelobjekt. Der Stapel bleibt unverändert. Falls der Stapel leer ist, wird `null` zurückgegeben.



## Unterlagen für die Lehrkraft

# Beispielaufgabe

## Informatik, Leistungskurs

---

### 1. Aufgabenart

Analyse, Modellierung und Implementation von kontextbezogenen Problemstellungen mit Schwerpunkt auf den Inhaltsfeldern Algorithmen und formale Sprachen und Automaten

### 2. Aufgabenstellung

siehe Prüfungsaufgabe

### 3. Materialgrundlage

entfällt

### 4. Bezüge zu dem Kernlehrplan und zu den Vorgaben

Die Aufgaben weisen vielfältige Bezüge zu den Kompetenzerwartungen und Inhaltsfeldern des Kernlehrplans bzw. zu den in den Vorgaben ausgewiesenen Fokussierungen auf. Im Folgenden wird auf Bezüge von zentraler Bedeutung hingewiesen.

#### 1. Inhaltsfelder und inhaltliche Schwerpunkte

Algorithmen

- Analyse, Entwurf und Implementierung von Algorithmen

Formale Sprachen und Automaten:

- Syntax und Semantik einer Programmiersprache (Java)
- Endliche Automaten und Kellerautomaten
- Grammatiken regulärer Sprachen
- Scanner, Parser und Interpreter für eine reguläre Sprache
- Möglichkeiten und Grenzen von Automaten und formalen Sprachen

#### 2. Medien/Materialien

– keine –

### 5. Zugelassene Hilfsmittel

- Wörterbuch zur deutschen Rechtschreibung
- Taschenrechner (grafikfähiger Taschenrechner / CAS-Rechner)

## 6. Modelllösungen

**Die Modelllösung stellt eine mögliche Lösung bzw. Lösungsskizze dar. Der gewählte Lösungsansatz und –weg der Schülerinnen und Schüler muss nicht identisch mit dem der Modelllösung sein. Sachlich richtige Alternativen werden mit entsprechender Punktzahl bewertet (Bewertungsbogen: Zeile „Sachlich richtige Lösungsalternative zur Modelllösung“).**

### Teilaufgabe a)

Vom Zustand  $q_6$  existieren zwei Übergänge mit dem Zeichen  $r$  nach  $q_0$  bzw.  $q_1$  und von  $q_7$  nach  $q_1$  und  $q_3$  ebenfalls zwei Übergänge mit dem Zeichen  $r$ . Damit ist der zum Zustandsübergangsgraph in Abbildung 1 gehörende Automat nichtdeterministisch.

Eine Zustandsfolge des Automaten zum Zustandsübergangsgraph in Abbildung 1 bei der Trainingseinheit  $rrirhrreerrr$  ist diese:

$$q_0 \xrightarrow{r} q_1 \xrightarrow{r} q_3 \xrightarrow{i} q_7 \xrightarrow{r} q_3 \xrightarrow{h} q_6 \xrightarrow{r} q_1 \xrightarrow{r} q_3 \xrightarrow{e} q_2 \xrightarrow{e} q_1 \xrightarrow{r} q_3 \xrightarrow{r} q_4 \xrightarrow{r} q_5$$

Es gibt nur drei Möglichkeiten, wie der Automat beginnt, die Trainingseinheit  $rrhrrihr$  zu prüfen, ...

1. ... bis zu folgendem Abbruch:

$$q_0 \xrightarrow{r} q_1 \xrightarrow{r} q_3 \xrightarrow{h} q_6 \xrightarrow{r} q_1 \xrightarrow{r} q_3 \xrightarrow{i} q_7 \xrightarrow{r} q_3 \xrightarrow{h} q_6 \xrightarrow{r} q_1 \xrightarrow{r} q_3 \xrightarrow{r} q_4$$

$q_4$  ist aber kein akzeptierender Endzustand.

2. ... bis zu folgendem Abbruch:

$$q_0 \xrightarrow{r} q_1 \xrightarrow{r} q_3 \xrightarrow{h} q_6 \xrightarrow{r} q_0 \xrightarrow{r} q_1$$

$q_1$  ist aber kein akzeptierender Endzustand.

3. ... bis zu folgendem Abbruch:

$$q_0 \xrightarrow{r} q_1 \xrightarrow{r} q_3 \xrightarrow{h} q_6 \xrightarrow{r} q_1 \xrightarrow{r} q_3 \xrightarrow{i} q_7 \xrightarrow{r} q_3 \xrightarrow{h} q_6 \xrightarrow{r} q_0 \xrightarrow{r} q_1 \xrightarrow{r} q_3$$

$q_3$  ist aber kein akzeptierender Endzustand.

Die Trainingseinheit wird nicht vom Automat akzeptiert, also ist sie nicht korrekt aufgebaut.

Es kann keine 15-minütige Trainingseinheit geben, in der die Belastungsphasen der Intervalle mit extensiver Trainingsintensität gelaufen werden können, denn abzüglich den zwei Minuten des Einlaufens und den zwei Minuten des Auslaufens verbleiben elf Minuten für die Intervalle. Ein Intervall in extensiver Intensität ist drei Minuten lang, davon zwei Minuten extensives Laufen und eine Minute im Tempo der Ruhephase. Da drei nicht elf teilt, kann es eine solche Trainingseinheit also nicht geben.

**Teilaufgabe b)**

Die folgende reguläre Grammatik erzeugt Wörter, die korrekten Trainingseinheiten im Sinne des Automaten aus Abbildung 1 entsprechen.

Terminale:  $T = \{e, i, h, r\}$   
 Nicht-Terminale:  $N = \{S, A, B, C, D, E, F, G\}$   
 Startsymbol:  $S$   
 Produktionen:  $P = \{$

$$\begin{aligned}
 & S \rightarrow rA, \\
 & A \rightarrow rB, \\
 & B \rightarrow rC \mid eD \mid hE \mid iF, \\
 & C \rightarrow rG, \\
 & D \rightarrow eA, \\
 & E \rightarrow rA \mid rS, \\
 & F \rightarrow rB \mid rA, \\
 & G \rightarrow \varepsilon
 \end{aligned}$$

$\}$

**Teilaufgabe c)**

Die Zustandsfolge des Automaten zum Zustandsübergangsdiagramm in Abbildung 1 bei der Eingabe der Trainingseinheit  $rririirrrr$  ist die folgende:

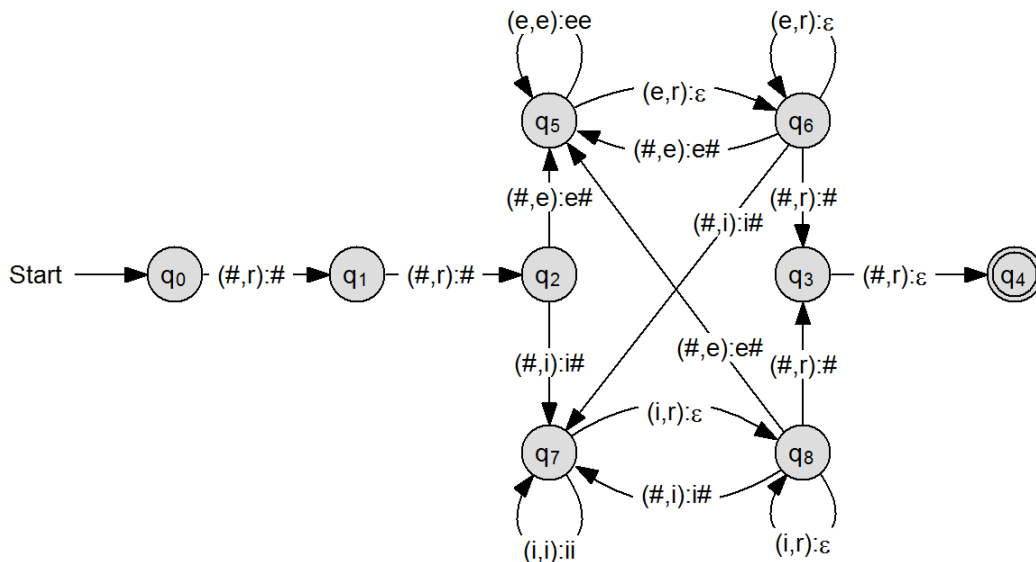
$$\begin{aligned}
 & (q_0, \#) \xrightarrow{r} (q_1, \#) \xrightarrow{r} (q_2, \#) \xrightarrow{i} (q_7, i\#) \xrightarrow{r} (q_8, \#) \xrightarrow{i} (q_7, i\#) \xrightarrow{i} (q_7, ii\#) \xrightarrow{r} (q_8, i\#) \xrightarrow{r} (q_8, \#) \\
 & \xrightarrow{r} (q_3, \#) \xrightarrow{r} (q_4, \#)
 \end{aligned}$$

Eine Trainingseinheit beginnt mit einer zweiminütigen Phase des Warmlaufens. Dann folgen entweder ein oder mehrere Intervalle, in der die Belastungsphase jeweils ausschließlich mit extensiver oder intensiver Intensität gelaufen werden sollen. Jedes einzelne Intervall besteht jeweils aus einer vorher in der Länge nicht festgelegten Belastungsphase und einer gleichlangen Ruhephase, z.B.  $er$ ,  $eerr$ ,  $eeerrr$ , ... oder  $ir$ ,  $iirr$ ,  $iiirrr$ , ... Den Abschluss der Trainingseinheit bildet eine zweiminütige Phase des Auslaufens im Tempo der Ruhephase.

Der Unterschied der Trainingseinheiten zu denen durch den Automaten in Abbildung 1 vorgegebenen Trainingseinheiten besteht aus folgenden Punkten:

- Mischformen der Trainingsintensität (extensiv, intensiv bzw. hochintensiv) innerhalb einer Trainingseinheit sind nicht erlaubt.
- Als Trainingsintensität ist ausschließlich extensiv bzw. intensiv erlaubt.
- Die Belastungs- und die sich anschließende Ruhephase müssen gleich lang sein.

Der ergänzte Zustandsübergangsgraph ist nachfolgend gegeben:



### Teilaufgabe d)

Eine mögliche Implementierung der fehlenden Methode wäre:

```

private int zustand6(char pAktZeichen) {
    int naechsterZustand = -1;
    if (!pruefStack.isEmpty()) {
        //Uebergang von q6 nach q6
        char topSym = pruefStack.top().charValue();
        pruefStack.pop();
        if (topSym == 'e' && pAktZeichen == 'r') {
            naechsterZustand = 6;
        }
    } else {
        if (pAktZeichen == 'r') {
            //Uebergang von q6 nach q3
            naechsterZustand = 3;
        } else if (pAktZeichen == 'e') {
            //Uebergang von q6 nach q5
            pruefStack.push(new Character('e'));
            naechsterZustand = 5;
        }
    }
    return naechsterZustand;
}

```

### Teilaufgabe e)

Die Korrektheit der Folge von Intensitätskürzel eeeiiihhrrrrrrrrrr wird durch die Angabe der Werte der Zähler `anzahlE`, `anzahlI`, `anzahlH` und `pos` nach Durchlaufen der einzelnen Schleifen gezeigt.

Zeile	Nachbedingung der Schleife
9	anzahlE = 3, pos = 3
13	anzahlI = 3, pos = 6
17	anzahlH = 3, pos = 9
22	pos = 18, anzahl = 0

Die Zähler `anzahlE`, `anzahlI` und `anzahlH` haben identische Werte und der Summenzähler `anzahl` ist auf 0 (Bedingung Zeile 28), also ist die Folge von Intensitätskürzeln korrekt.

Selbiger Nachweis wird für die Folge von Intensitätskürzeln `eeihrrrrr` geführt:

Zeile	Nachbedingung der Schleife
9	anzahlE = 2, pos = 2
13	anzahlI = 2, pos = 4
17	anzahlH = 1, pos = 5
22	pos = 10, anzahl = 0

Die Folge ist nicht korrekt, da bereits die Bedingung `anzahlE != anzahlH` erfüllt ist, d.h. die Minutenanzahl mit hochintensiver Trainingsintensität stimmt nicht mit der Minutenanzahl mit extensiver Trainingsintensität überein.

Als eine fehlerhafte Folge von Intensitätskürzeln sei `ehrr` gegeben. Diese Folge erfüllt die Bedingung `anzahlE != anzahlI`, d.h. die Minutenanzahl mit intensiver Trainingsintensität stimmt nicht mit der Minutenanzahl mit extensiver Trainingsintensität überein.

Als eine weitere fehlerhafte Folge von Intensitätskürzeln sei `eihrr` gegeben. Diese Folge erfüllt die Bedingung `anzahl != 0`, d.h. die Minutenanzahl der Belastungsphase des gegebenen Intervalls entspricht nicht der Minutenanzahl der Ruhephase.

**7. Teilleistungen – Kriterien / Bewertungsbogen zur Prüfungsarbeit**

Name des Prüflings: \_\_\_\_\_ Kursbezeichnung: \_\_\_\_\_

Schule: \_\_\_\_\_

**Teilaufgabe a)**

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK <sup>1</sup>	ZK	DK
1	erläutert, warum der Automat nichtdeterministisch ist.	2 (I)			
2	gibt die Zustandsfolge des Automaten für rrrrreerrr an.	3 (I)			
3	zeigt durch die Angabe aller möglichen Zustandsfolgen, dass rrrrrirhrrr nicht akzeptiert wird.	4 (I)			
4	beurteilt, ob es eine entsprechende 15-minütige Trainingseinheit gibt.	3 (III)			
Sachlich richtige Lösungsalternative zur Modelllösung: (12)					
.....					
.....					
<b>Summe Teilaufgabe a)</b>		12			

**Teilaufgabe b)**

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
1	entwirft eine reguläre Grammatik, deren Sprache genau die Worte enthält, welche der Automat aus Abbildung 1 akzeptiert.	8 (II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (8)					
.....					
.....					
<b>Summe Teilaufgabe b)</b>		8			

<sup>1</sup> EK = Erstkorrektur; ZK = Zweitkorrektur; DK = Drittkorrektur

**Teilaufgabe c)**

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
1	stellt für die Trainingseinheit die Zustandsfolge dar.	2 (I)			
2	erläutert im Sachkontext den Aufbau einer korrekten Trainingseinheit im Sinne des Automaten in Abbildung 2.	2 (II)			
3	nennt die Veränderungen gegenüber dem Aufbau einer Trainingseinheit im Sinne von Automat 1.	3 (II)			
4	erweitert den Zustandsübergangsgraphen des Automaten aus Abbildung 2.	4 (II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (11) ..... .....					
<b>Summe Teilaufgabe c)</b>		11			

**Teilaufgabe d)**

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
1	implementiert die Methode zustand6.	8 (II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (8) ..... .....					
<b>Summe Teilaufgabe d)</b>		8			

**Teilaufgabe e)**

<b>Anforderungen</b>		<b>Lösungsqualität</b>			
		maximal erreichbare Punktzahl (AFB)	<b>EK</b>	<b>ZK</b>	<b>DK</b>
	<b>Der Prüfling</b>				
1	zeigt die Korrektheit der durch Intensitätskürzel gegebenen Trainingseinheit.	2 (II)			
2	zeigt die Fehlerhaftigkeit der durch Intensitätskürzel gegebenen Trainingseinheit.	2 (II)			
3	gibt zwei unterschiedlich fehlerhafte Trainingseinheiten in Form der Intensitätskürzel an.	6 (III)			
Sachlich richtige Lösungsalternative zur Modelllösung: (10)					
.....					
.....					
<b>Summe Teilaufgabe e)</b>		10			