



Name: _____

Beispielaufgabe

Informatik, Grundkurs

Aufgabenstellung

Im englischen Sprachraum gibt es, wie auch in der deutschen Sprache, das Bestreben, eine geschlechtergerechte Sprache zu benutzen. Ein typisches Verfahren ist, das Teilwort man durch das Wort person zu ersetzen, so dass zum Beispiel aus dem Wort salesman das Wort salesperson wird.

Im Folgenden sollen englischsprachige Texte auf den Gebrauch von geschlechtergerechter Sprache rein syntaktisch untersucht werden. In einem ersten Ansatz sollen dazu die Wörter herausgefiltert werden, die als Teilwort die Zeichenfolge man haben.

Der folgende endliche Automat A, der durch einen Übergangsgraphen in Abbildung 1 angegeben ist, wird entworfen, um einzelne Wörter darauf zu untersuchen. Dabei wird davon ausgegangen, dass nur kleine Buchstaben in einem Wort vorkommen.

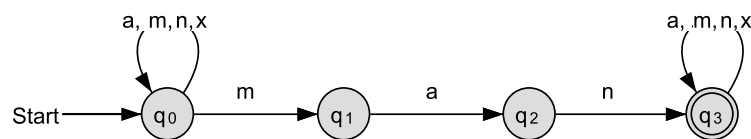


Abbildung 1: Zustandsübergangsgraph des Automaten

Das Zeichen x steht für einen beliebigen Kleinbuchstaben des Alphabets mit Ausnahme von a, m, n.

a) Überführen Sie den Übergangsgraphen aus Abbildung 1 in eine Zustandsübergangstabelle.

Begründen Sie, dass es sich nicht um einen deterministischen endlichen Automaten handelt.

Beschreiben Sie, wie man feststellen kann, ob ein Wort von diesem Automaten akzeptiert wird.

Zeigen Sie, dass die Wörter salesman und command zur Sprache des Automaten gehören.

(8 Punkte)

b) Der Automat, der durch den Übergangsgraphen in Abbildung 1 gegeben ist, soll erweitert und in einen deterministischen Automaten umgewandelt werden, der zusätzlich das Teilwort men akzeptiert.

Entwickeln Sie den Zustandsübergangsgraphen eines deterministischen endlichen Automaten, der diesen Anforderungen entspricht.

(9 Punkte)



Name: _____

- c) Zu Testzwecken sollen Wörter erzeugt werden, die ein bestimmtes Teilwort enthalten. Die folgende Grammatik G erzeugt entsprechende Wörter.

Das Zeichen x steht hier für einen beliebigen Kleinbuchstaben des Alphabets $a \dots z$ mit Ausnahme von n, o, s .

Startsymbol: A

Menge der Nichtterminalsymbole: $N = \{A, B, C, D, E\}$

Menge der Terminalsymbole: $T = \{n, o, s, x\}$

Menge der Produktionen:

$P = \{$

$A \rightarrow BC$

$B \rightarrow \varepsilon \mid nB \mid oB \mid sB \mid xB$

$C \rightarrow sD$

$D \rightarrow oE$

$E \rightarrow n$

$\}$

Zeigen Sie, dass das Wort $ingason$ zu der Sprache gehört, die von der Grammatik G erzeugt wird.

Bestimmen Sie das kürzeste Wort, das zu der Sprache gehört, die von der Grammatik G erzeugt wird.

Erläutern Sie, dass die Grammatik G nicht regulär ist.

Entwickeln Sie eine reguläre Grammatik, die die gleiche Sprache erzeugt.

Beschreiben Sie, welche Wörter durch die Grammatik G erzeugt werden.

(14 Punkte)

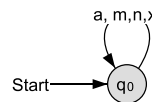


Name: _____

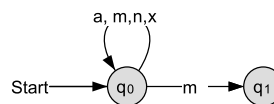
Für ein spezielles Teilwort w der Länge n kann in der oben dargestellten Form ein **nichtdeterministischer** endlicher Automat mit $n+1$ Zuständen konstruiert werden, der erkennt, ob das Teilwort in einem beliebigen Wort enthalten ist:

(Das Verfahren wird grafisch am Beispiel aus Abbildung 1 dargestellt.)

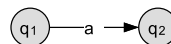
- q_0 sei der Startzustand, der unter jedem Eingabezeichen in sich selbst übergeht. Dies stellt zunächst sicher, dass jedes beliebige Wort aus dem Alphabet eingegeben werden kann und vom Automaten nicht akzeptiert wird, da der Startzustand kein Endzustand ist.



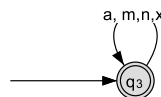
- Für die Zustände q_0 bis q_{n-1} gilt: q_i geht unter dem $(i+1)$ -ten Zeichen des gesuchten Teilworts w in q_{i+1} über:
 q_0 geht unter dem ersten Zeichen des Teilworts in q_1 über,



q_1 geht unter dem zweiten Zeichen des Teilworts in q_2 über und so fort.



- q_n ist der einzige Endzustand und geht unter jedem Eingabezeichen in sich selbst über.



- d) Ein Schüler behauptet, dass ein spezielles Teilwortproblem nur dann von einem **deterministischen** endlichen Automaten mit $n+1$ Zuständen gelöst wird, wenn kein Zeichen im Teilwort doppelt vorkommt. Wenn ein Zeichen zum zweiten Mal gelesen werde, könnte der Automat nicht wissen, ob es sich um das erste oder das zweite Vorkommen des Zeichens im Teilwort handele.

Diese Aussage soll anhand des folgenden Beispiels untersucht werden: Ein Automat soll überprüfen, ob ein Eingabewort das Teilwort *mommy* enthält.

*Entwickeln Sie zunächst nach dem oben beschriebenen Verfahren einen nicht-deterministischen endlichen Automaten, der erkennt, ob das Teilwort *mommy* in einem Eingabewort enthalten ist.*

Begründen Sie, dass die Behauptung falsch ist.

(10 Punkte)



Name: _____

e) Um Automaten in konkreten Kontexten zu implementieren, kann man eine abstrakte Klasse Zustand verwenden, aus der für jeden Zustand eine eigene Klasse abgeleitet wird. Im Folgenden wird eine Teilmodellierung des Automaten, dessen Zustandsübergangsgraph in Abbildung 1 zu angegeben ist, in Form eines Implementationsdiagramms dargestellt.

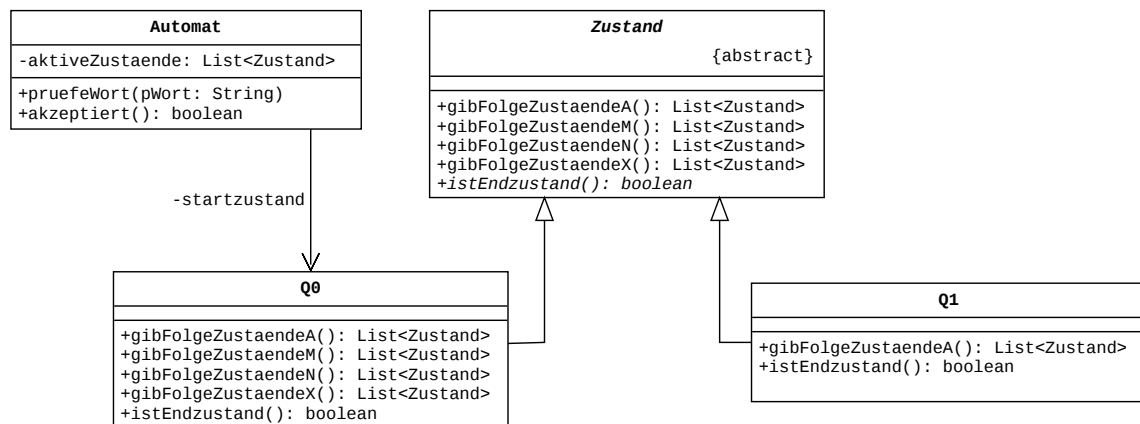


Abbildung 2: Teilmodell in Form eines Implementationsdiagramms

Das Ergebnis eines Zustandsübergangs wird bestimmt, indem eine entsprechende Methode des Objekts einer Klasse aufruft. Beispielsweise wird bei der Eingabe des Zeichens 'a' im Zustand q_3 die Methode `gibFolgeZustaendeA` aufgerufen. Die Methoden liefern als Ergebnis jeweils die Liste der Folgezustände, wenn das im Methodennamen angegebene Eingabezeichen gelesen wird. Diese Liste wird dem Attribut mit dem Bezeichner `aktiveZustaende` der Klasse `Automat` hinzugefügt.

Die Methoden der abstrakten Klasse `Zustand` liefern jeweils eine leere Liste. Mit der Methode `istEndzustand` kann überprüft werden, ob der Zustand ein Endzustand ist.

Die Methode `pruefeWort` der Klasse `Automat` ermöglicht die Eingabe des vom Automaten zu prüfenden Wortes, die Methode `akzeptiert` liefert, ob das eingegebene Wort vom Automaten akzeptiert wurde.

Erweitern Sie das Implementationsdiagramm um die Klassen Q_2 und Q_3 .

Hinweis: Verwenden Sie dazu die Vorlage in der Anlage.

Erläutern Sie, dass diese Modellierung sowohl für deterministische als auch für nichtdeterministische endliche Automaten geeignet ist.

(9 Punkte)

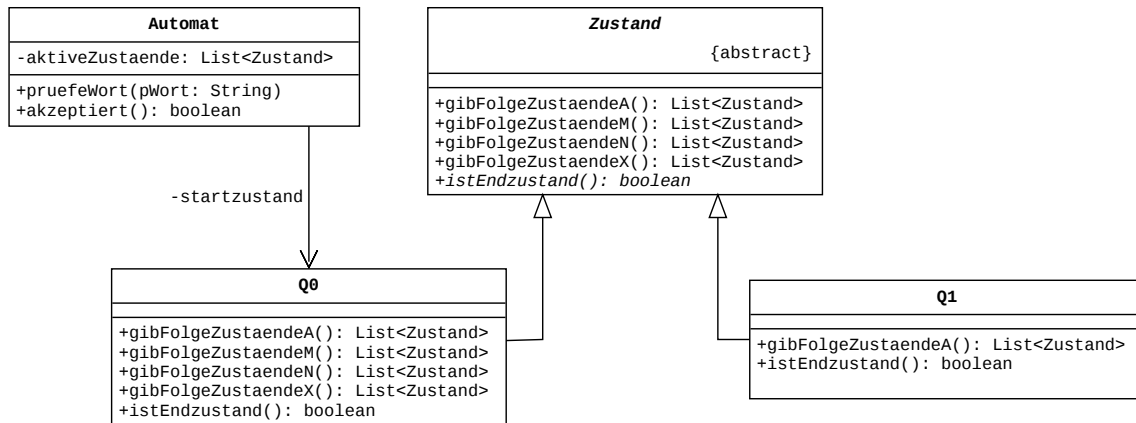
Zugelassene Hilfsmittel

- Wörterbuch zur deutschen Rechtschreibung
- GTR (graphikfähiger Taschenrechner) oder CAS (Computer-Algebra-System)



Name: _____

Anlage: Vorlage für die Bearbeitung von Aufgabenteil e)



Unterlagen für die Lehrkraft

Beispielaufgabe

Informatik, Grundkurs

1. Aufgabenart

Analyse, Modellierung und Implementation von kontextbezogenen Problemstellungen mit Schwerpunkt auf den Inhaltsfeldern Daten und ihre Strukturierung 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

Daten und ihre Strukturierung:

- Objekte und Klassen
 - Implementationsdiagramme

Formale Sprachen und Automaten:

- Endliche Automaten
 - Deterministische endliche Automaten
 - Nichtdeterministische endliche Automaten
- Grammatiken regulärer Sprachen
 - Produktionen mit ϵ
- Möglichkeiten und Grenzen von Automaten und formalen Sprachen

2. Medien/Materialien

entfällt

5. Zugelassene Hilfsmittel

- Wörterbuch zur deutschen Rechtschreibung
- GTR (graphikfähiger Taschenrechner) oder CAS (Computer-Algebra-System)

6.1 Hinweise zur Beispielklausur

Das Ziel dieser Beispielaufgabe ist es, eine mögliche Verknüpfung der Inhaltsfelder „Daten und ihre Strukturierung“ und „Formale Sprachen und Automaten“ (deterministische endliche Automaten, nichtdeterministische endliche Automaten) für eine inhaltsfeldübergreifende Grundkursklausur zu zeigen.

Diese Beispielaufgabe orientiert sich an den Abiturvorgaben für 2021.

Ab dem Abiturjahrgang 2021 ist für eine Grundkursklausur, die aus zwei Aufgaben besteht, eine Bearbeitungszeit von 3 Stunden und 45 Minuten vorgesehen. Diese Beispielaufgabe stellt nur eine von zwei Aufgaben dar, die zusammen eine vollständige Grundkursklausur ergeben.

6.2 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)

Zustandsübergangstabelle

	a	m	n	x
q ₀	{q ₀ }	{q ₀ , q ₁ }	{q ₀ }	{q ₀ }
q ₁	{q ₂ }	{}	{}	{}
q ₂	{}	{}	{q ₃ }	{}
q ₃	{q ₃ }	{q ₃ }	{q ₃ }	{q ₃ }

Der Automat ist kein deterministischer endlicher Automat, weil es vom Zustand q₀ mit dem Eingabezeichen m zwei Zustandsübergänge zu unterschiedlichen Folgezuständen gibt.

Ein Wort wird von dem Automaten akzeptiert, wenn es beginnend mit den Startzustand q₀ eine Zustandsfolge gibt, die im Endzustand q₃ endet, wobei jedes Zeichen des Wortes einen Zustandsübergang bewirkt.

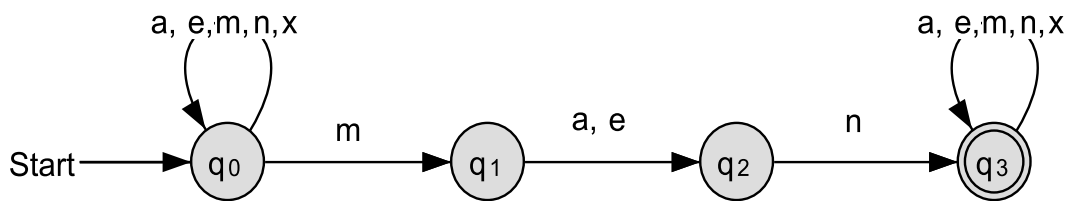
$$q_0 \xrightarrow{s} q_0 \xrightarrow{a} q_0 \xrightarrow{l} q_0 \xrightarrow{e} q_0 \xrightarrow{s} q_0 \xrightarrow{m} q_1 \xrightarrow{a} q_2 \xrightarrow{n} q_3$$

$$q_0 \xrightarrow{c} q_0 \xrightarrow{o} q_0 \xrightarrow{m} q_0 \xrightarrow{m} q_1 \xrightarrow{a} q_2 \xrightarrow{n} q_3 \xrightarrow{d} q_3$$

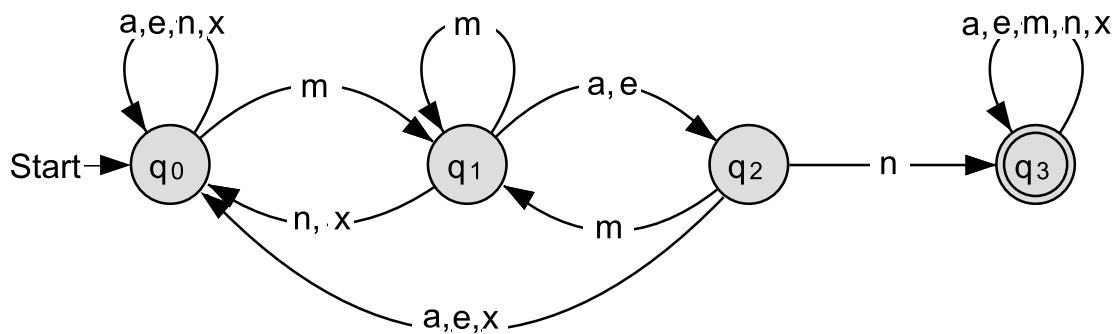
Teilaufgabe b)

Das Zeichen x steht in dieser Teilaufgabe für einen beliebigen Kleinbuchstaben des Alphabets mit Ausnahme von a, e, m, n.

Der folgende nichtdeterministische Automat akzeptiert die geforderten Wörter.



Durch Änderung der Zustandsübergänge kann man einen deterministischen Automaten konstruieren, der die gleichen Wörter akzeptiert.



Teilaufgabe c)

$A \rightarrow BC \rightarrow iBC \rightarrow inBC \rightarrow ingBC \rightarrow ingaC \rightarrow ingasD \rightarrow ingasoE \rightarrow ingason$

Das kürzeste Wort ergibt sich, wenn die Produktion $B \rightarrow \epsilon$ sofort zur Anwendung kommt.

$A \rightarrow BC \rightarrow C \rightarrow sD \rightarrow soE \rightarrow son$

Eine reguläre Grammatik kann entweder rechtslinear oder linkslinear sein. Bei einer rechtslinearen Grammatik haben alle Produktionen auf der rechten Seite ein Terminalsymbol oder ein Terminalsymbol gefolgt von einem Nichtterminalsymbol. Bei einer linkslinearen Grammatik haben alle Produktionen auf der rechten Seite entweder ein Terminalsymbol oder ein Nichtterminalsymbol gefolgt von einem Terminalsymbol.

Die Produktion $A \rightarrow BC$ verstößt gegen diese Regel, denn auf der rechten Seite stehen zwei Nichtterminalsymbole.

Die folgende reguläre Grammatik erzeugt die gleiche Sprache:

Terminalsymbole: $T = \{n, o, s, x\}$

Nichtterminalsymbole: $N = \{S, T, U\}$

Startsymbol: S

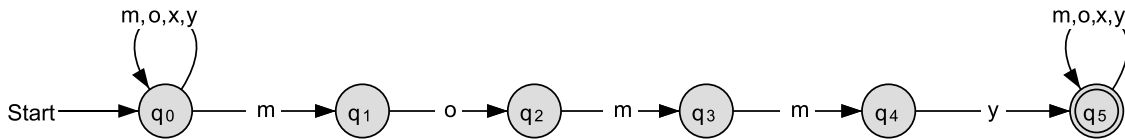
Produktionen: $P = \{$
 $S \rightarrow nS \mid oS \mid sS \mid xS \mid sT,$
 $T \rightarrow oU,$
 $U \rightarrow n$
 $\}$

Es handelt sich bei der oben angegebenen Grammatik um eine rechtslineare Grammatik, die damit regulär ist.

Durch die Grammatik werden alle Wörter bestimmt, die auf son enden.

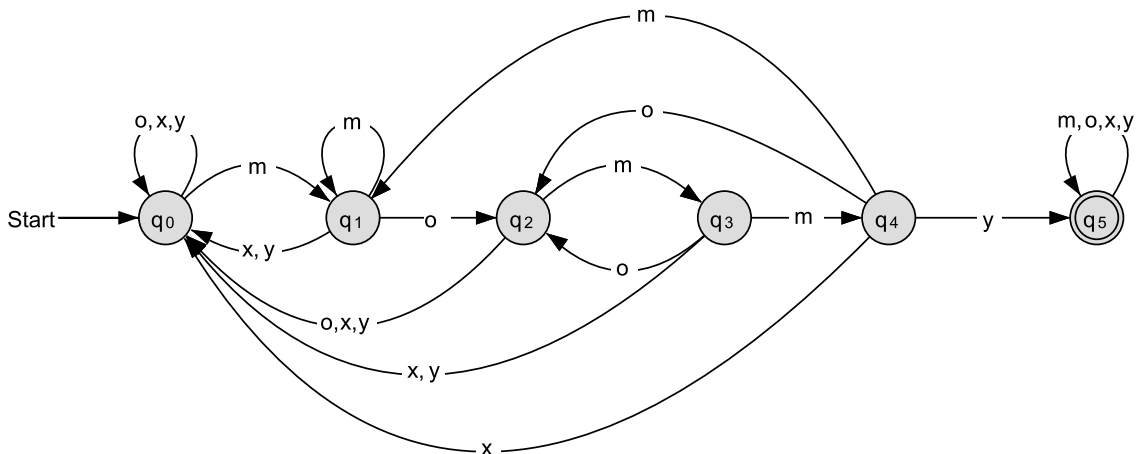
Teilaufgabe d)

Der Automat, der gemäß der Anleitung entsteht: Der Startzustand geht unter allen Eingabezeichen in sich selbst über. Außerdem geht es mit den Zeichen m , o , m , m und y vom Startzustand schrittweise in die Zustände q_1 bis q_5 über. Der Zustand q_5 ist der einzige Endzustand und geht unter allen Eingabezeichen in sich selbst über.



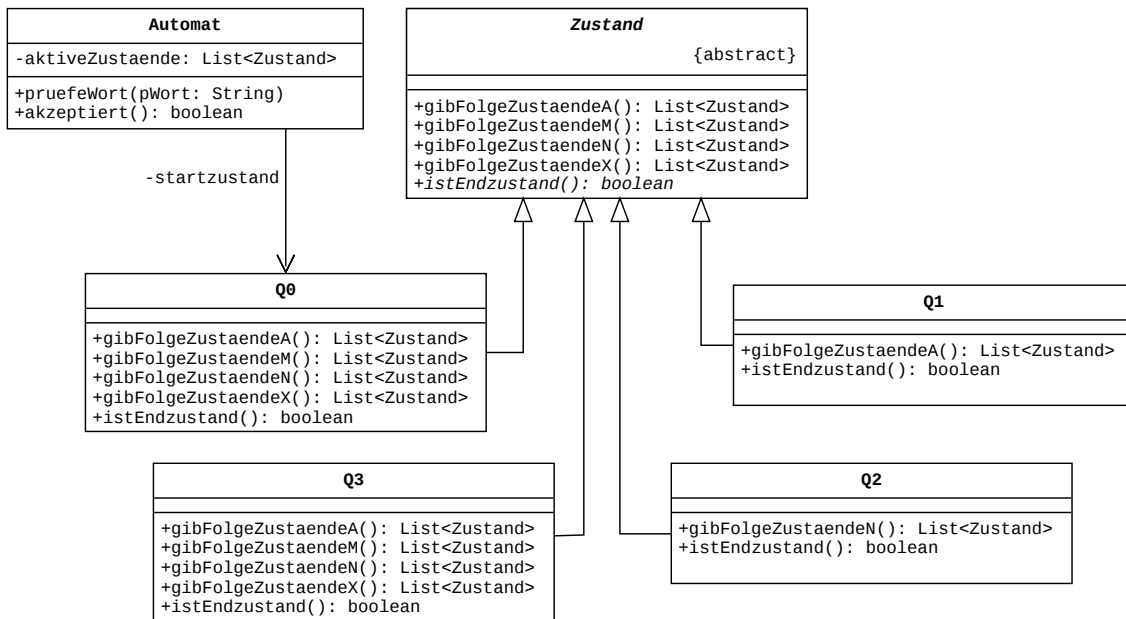
Die Behauptung kann widerlegt werden, indem der nicht-deterministischer Automat in einen deterministischen endlichen Automaten überführt wird, ohne dass sich dabei die Anzahl der Zustände erhöht.

Veränderung der Zustandsübergänge, um den nichtdeterministischen endlichen Automaten in einen deterministischen endlichen Automaten zu überführen:



Damit existiert ein Automat mit der genannten Anzahl von Zuständen, der die gleiche Sprache erkennt.

Teilaufgabe e)



Für einen nicht-deterministischen Automaten gilt, dass er unter dem gleichen Eingabezeichen und aus dem gleichen Zustand in verschiedene Zustände wechseln kann, während ein deterministischer endlicher Automat in maximal einen neuen Zustand wechseln kann.

Die Übergänge zwischen den Zuständen werden im Modell durch Methodenaufrufe dargestellt. Die Rückgabe der Methodenrufe ist jeweils eine Liste aller möglichen Folgezustände. Dies ermöglicht, dass in einem deterministischen Programm der Nicht-Determinismus dadurch abgebildet wird, dass alle möglichen Folgezustände gleichzeitig betrachtet werden; der Automat befindet sich sozusagen in einem Meta-Zustand, der einer Sammlung aller möglichen Folgezustände entspricht. Dieser Meta-Zustand wird durch das Attribut `aktiveZustaende` verwaltet. Aus dieser Liste der aktiven Zustände können bei folgenden Eingaben wiederum alle möglichen Folgezustände ermittelt werden.

Der Automat akzeptiert ein Wort, wenn nach Abarbeitung des Wortes die aktuelle Liste der aktiven Zustände mindestens einen Endzustand enthält.

Bei einem deterministischen endlichen Automaten enthält die Liste die Rückgabeliste der Methoden und damit auch das Attribut `aktiveZustaende` stets maximal einen Zustand.

7. Teilleistungen – Kriterien / Bewertungsbogen zur Prüfungsarbeit

Name des Prüflings: _____ Kursbezeichnung: _____

Schule: _____

Teilaufgabe a)

Anforderungen		Lösungsqualität			
		maximal erreichbare Punktzahl (AFB)	EK ¹	ZK	DK
	Der Prüfling				
1	überführt den Übergangsgraphen in eine Zustandsübergangstabelle.	2 (II)			
2	begründet, dass es sich nicht um einen deterministischen Automaten handelt.	2 (II)			
3	beschreibt, wie man feststellen kann, ob ein Wort akzeptiert wird.	2 (I)			
4	zeigt, dass die Wörter salesman und command zur Sprache es Automaten gehören.	2 (II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (8)					
Summe Teilaufgabe a)		8			

Teilaufgabe b)

Anforderungen		Lösungsqualität			
		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
	Der Prüfling				
1	entwickelt einen Zustandsübergangsgraph eines deterministischen endlichen Automaten, der den beschriebenen Anforderungen genügt.	9 (II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (9)					
Summe Teilaufgabe b)		9			

¹ EK = Erstkorrektur; ZK = Zweitkorrektur; DK = Drittkorrektur

Teilaufgabe c)

Anforderungen		Lösungsqualität			
		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
	Der Prüfling				
1	zeigt, dass das Wort <i>ingason</i> zu der von der Grammatik G erzeugten Sprache gehört.	3 (II)			
2	bestimmt das kürzeste Wort, das zu der von der Grammatik G erzeugten Sprache gehört.	2 (II)			
3	erläutert, dass die Grammatik nicht regulär ist.	3 (II)			
4	entwickelt eine reguläre Grammatik, die die gleiche Sprache erzeugt.	4 (II)			
5	beschreibt im Sachzusammenhang die Wörter, die durch die Grammatik erzeugt werden.	2 (I)			
Sachlich richtige Lösungsalternative zur Modelllösung: (14)					
Summe Teilaufgabe c)		14			

Teilaufgabe d)

Anforderungen		Lösungsqualität			
		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
	Der Prüfling				
1	entwickelt nach dem beschriebenen Verfahren einen nicht-deterministischen endlichen Automaten, der erkennt, ob das angegebene Teilwort in einem Eingabewort enthalten ist.	4 (II)			
2	begründet, dass die Aussage falsch ist.	6 (III)			
Sachlich richtige Lösungsalternative zur Modelllösung: (10)					
Summe Teilaufgabe c)		10			

Teilaufgabe e)

Anforderungen		Lösungsqualität			
		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
	Der Prüfling				
1	erweitert das Implementationsdiagramm um die Klassen Q2 und Q3.	5			
2	erläutert, dass diese Modellierung sowohl für deterministische als auch für nicht-deterministische Automaten geeignet ist.	4			
Sachlich richtige Lösungsalternative zur Modelllösung: (9)					
Summe Teilaufgabe e)		9			