



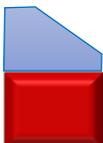
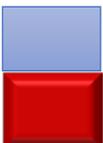
Name: _____

Beispielaufgabe

Informatik, Leistungskurs

Aufgabenstellung

Eine Firma konstruiert aus Fertigteilen verschiedene Spielzeugautos. Folgende Bauteile stehen dafür zur Verfügung:

Bauteil	Bezeichnung	Kurzbezeichnung
	block	bl
	blockmitrad	br
	frontscheibe	fs
	heckscheibe	hs
	tuer	tu
	fenster	fe
	spoiler	sp



Name: _____

Die zu konstruierenden Spielzeugautos werden durch eine formale Sprache $L(G)$ beschrieben, die durch eine Grammatik $G = \{T, N, P, S\}$ definiert wird.

$T = \{bl, br, fs, hs, tu, fe, sp\}$

$N = \{A, B, C, D, E, F\}$

$S = A$

$P = \{$

$A \rightarrow blA \mid brB \mid hsc,$

$B \rightarrow brB \mid blB \mid hsc \mid feC \mid tuC \mid fsD,$

$C \rightarrow feC \mid tuC \mid fsD,$

$D \rightarrow blD \mid brE,$

$E \rightarrow blE \mid brE \mid spF \mid \varepsilon,$

$F \rightarrow \varepsilon$

$\}$

Beispiel:

Das Wort $blhsfetufsb r sp$ ist der Code für folgendes Spielzeugauto:

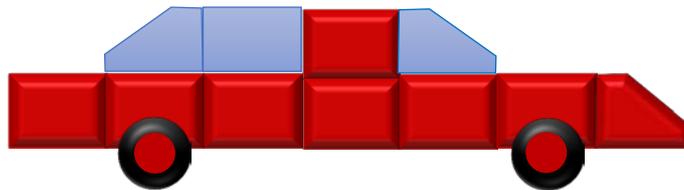


Abbildung 1: Beispiel für ein Spielzeugauto mit dem Code $blhsfetufsb r sp$

a)

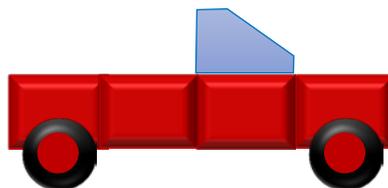


Abbildung 2: Spielzeugauto für Aufgabenteil a)

Geben Sie zu dem in Abbildung 2 dargestellten Auto das entsprechende Wort der Autosprache $L(G)$ an.

Geben Sie ein Wort der Autosprache $L(G)$ für ein Auto an, das aus neun Bauteilen besteht.

Bestimmen Sie die beiden Wörter der Autosprache $L(G)$, die Autos aus genau drei Bauteilen beschreiben.

(8 Punkte)



Name: _____

- b) Zeigen Sie anhand der Grammatik G , dass das Wort $brbrblfsfsbr$ nicht zur Autosprache $L(G)$ gehört.

Zeigen Sie mithilfe der Grammatik G , dass das Wort $hstufefsbrsp$ zur Autosprache gehört.

(10 Punkte)

- c) Begründen Sie, dass die Grammatik G der Autosprache $L(G)$ regulär ist.

Entwickeln Sie einen endlichen Automaten, der alle Wörter der Autosprache akzeptiert.

(12 Punkte)

Die Firma gibt eine Software in Auftrag, mit der interaktiv Automodelle konstruiert werden können.

Nach Eingabe eines Wortes der Autosprache prüft ein Parser, ob das zugehörige Automodell entsprechend der Grammatik $L(G)$ zulässig ist, und zeigt dann eine Abbildung des Modells an oder gibt eine Fehlermeldung aus.

Bevor der Parser die Syntax des Wortes der Autosprache prüft, zerlegt der Scanner das zu parsende Wort in eine Liste der im Wort enthaltenen Terminale der Autosprache $L(G)$.

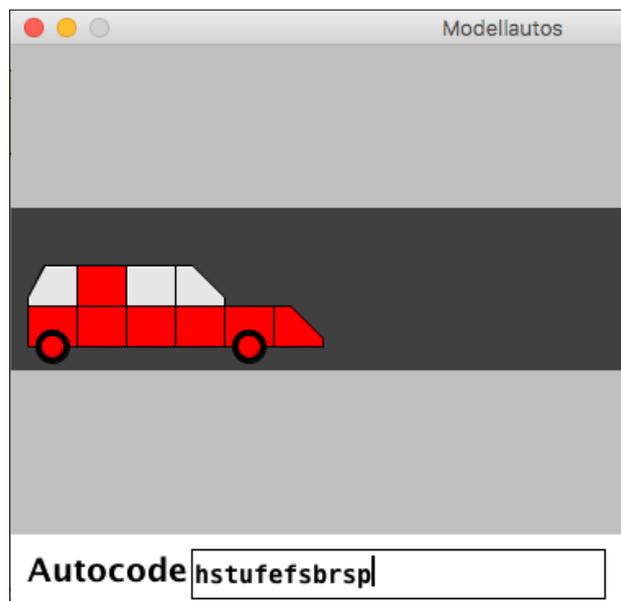


Abbildung 3: Interaktives Erstellen von Automodellen

Für das Beispiel in Abbildung 3 gibt die Methode `scan` mit dem Methodenkopf

```
public List<String> scan(String pWort)
```

die Liste `("hs", "tu", "fe", "fs", "br", "sp")` zurück, wenn ihr das Wort `"hstufefsbrsp"` als Parameter übergeben wurde. Diese Liste wird im Folgenden **Tokenliste** genannt. Falls das Wort nicht ausschließlich aus Terminalen der Autosprache zusammengesetzt ist, wird eine leere Liste zurückgegeben.

- d) Entwickeln Sie einen Algorithmus für das oben beschriebene Zerlegungsproblem.

(6 Punkte)



Name: _____

e) Quellcode 1 ist ein Ausschnitt aus der Klasse AutoParser.

```
1 public class AutoParser {
2     private List<String> tokenListe;
3     private String token;
4     private Scanner scanner;

5     public AutoParser() {
6         scanner = new Scanner();
7     }

8     private String gibNaechstesToken() {
9         if (tokenListe.hasAccess()) {
10            String token = tokenListe.getContent();
11            tokenListe.next();
12            return token;
13        } else {
14            return "";
15        }
16    }

17    public boolean parse(String pWort) {
18        tokenListe = scanner.scan(pWort);
19        if (!tokenListe.isEmpty()) {
20            tokenListe.moveToFirst();
21            token = gibNaechstesToken();
22            return giltA();
23        } else {
24            return false;
25        }
26    }

27    private boolean giltA() {
28        if (token.equals("bl")) {
29            token = gibNaechstesToken();
30            return giltA();
31        } else if (token.equals("br")) {
32            token = gibNaechstesToken();
33            return giltB();
34        } else if (token.equals("hs")) {
35            token = gibNaechstesToken();
36            return giltC();
37        } else {
38            return false;
39        }
40    }

41    private boolean giltB() {...}
42    private boolean giltC() {...}
43    private boolean giltD() {...}
44    private boolean giltE() {...}
45    private boolean giltF() {...}
46 }
47 }
```

Quellcode 1: Ausschnitt des Quellcodes der Klasse AutoParser



Name: _____

Analysieren und erläutern Sie die Methode `parse` und die Methode `giltA` anhand des zu parsenden Wortes "hstufefsbrsp".

Ermitteln Sie unter Zuhilfenahme der Grammatik G der Autosprache $L(G)$ die Methoden, die beim Parsen des Wortes "hsfsbr" aufgerufen werden.

Implementieren Sie die Methode `giltE`.

(14 Punkte)

Zugelassene Hilfsmittel:

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

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 den Vorgaben¹

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
- Grammatiken regulärer Sprachen¹
- 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)

¹ Produktionen mit ϵ

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)

Zum Auto in Abb. 2 gehört das Wort: brblfsbr

Ein mögliches Wort der Autosprache mit 9 Terminalen: brbrhstututufsbrsp

Die beiden einzigen Wörter der Autosprache, die jeweils ein Auto aus drei Bauteilen beschreiben sind: hsfbr und brfsbr

Teilaufgabe b)

Aus dem Versuch, das Wort brbrblfsfsbr mithilfe der Grammatik zu erzeugen, ergibt sich folgende Ableitungskette:

A -> brB -> brbrB -> brbrblB -> brbrblfsD -> Widerspruch!

Es gibt keine Produktion mit dem Nichtterminal D auf der linken Seite, deren rechte Seite mit dem Terminal fs beginnt. Daher lässt sich das Wort nicht ableiten und gehört somit nicht zur Autosprache.

Das Wort hstufefsbrsp lässt sich mithilfe folgender Ableitungskette aus der Grammatik der Autosprache L(G) ableiten:

A -> hsc -> hstuC -> hstufec -> hstufefsD -> hstufefsbrE -> hstufefsbrspF -> hstufefsbrsp

Teilaufgabe c)

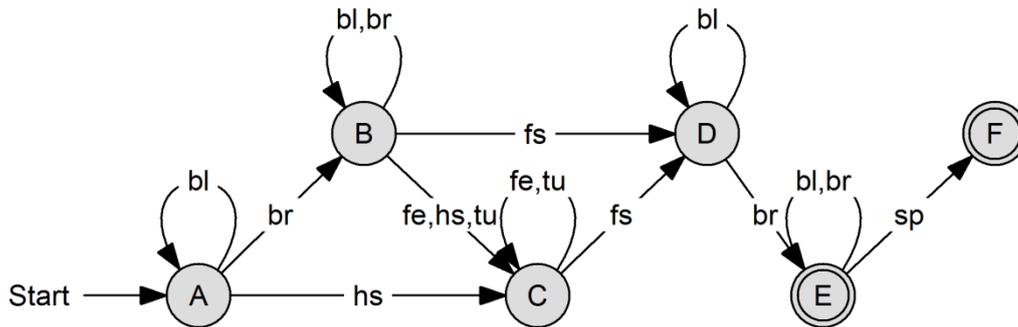
Eine Grammatik ist regulär, wenn sie entweder rechtsregulär oder linksregulär ist.

Eine Grammatik ist **rechtsregulär**, wenn für alle Produktionen der Grammatik folgende Regeln gelten:

1. Die linke Seite einer Produktion besteht nur aus **einem** Nichtterminal.
2. Die rechte Seite einer Produktion enthält das leere Wort, ein Terminal oder ein Terminal gefolgt von einem Nichtterminal.

Die Autogrammatik L(G) erfüllt beide Kriterien für eine rechtsreguläre Grammatik. Auf der linken Seite der Produktionen stehen nur Nichtterminale und auf der rechten Seite entweder das leere Wort oder ein Terminal gefolgt von einem Nichtterminal.

Zu jeder regulären Grammatik gibt es einen äquivalenten endlichen Automaten. Aus der Grammatik $L(G)$ lässt sich folgender endliche Automat konstruieren:



Alle nicht angegebenen Eingaben führen in einen Fehlerzustand.

Teilaufgabe d)

Algorithmus

Wenn das Wort aus einer ungeraden Zahl von Zeichen besteht, liegt ein Fehlerfall vor. Ansonsten wird das zu scannende Wort in Paare von Zeichen zerlegt.

Für jedes Zeichenpaar wird folgender Arbeitsschritt durchgeführt. Wenn das Zeichenpaar Terminal der Grammatik ist, wird es an die anfangs leere Tokenliste angehängt, andernfalls liegt ein Fehlerfall vor.

Die Tokenliste wird zurückgegeben oder im Fehlerfall die leere Liste.

Teilaufgabe e)

Analyse der Methode `parse`:

1. Aufruf der Methode `parse("hstufefsbrsp")`
2. Der Scanner bestimmt die Tokenliste:
`tokenliste = ("hs", "tu", "fe", "fs", "br", "sp")`
3. Da die Tokenliste nicht leer ist, wird das erste Element der Tokenliste zum aktuellen Element.
4. Der Variablen `token` wird durch Aufruf der Methode `gibNaechstesToken()` der Wert "hs" zugewiesen.
5. Das boolesche Ergebnis des Aufrufes der Methode `giltA()` wird zurückgegeben.

Analyse der Methode `giltA()`:

Die Methode `A` ruft in Abhängigkeit des Wertes der Variablen `token` eine Methode mit booleschem Rückgabewert auf und gibt den erhaltenen Wert zurück.

Wenn `token` gleich "bl", wird `giltA()` aufgerufen und der erhaltene boolesche Wert zurückgegeben,

wenn `token` gleich "br", wird `giltB()` aufgerufen und der Wert zurückgegeben,

wenn `token` gleich "hs", wird `giltC()` aufgerufen und der Wert zurückgegeben.

In allen anderen Fällen gibt die Methode `giltA()` den Wert `false` zurück.

Bei dem vorgegebenen Beispiel hat die Variable `token` den Wert `"hs"` und es wird `giltC()` aufgerufen.

Die Methode `parse(String pWort)` hat die Funktion zu prüfen, ob sein als Parameter übergebener String der Syntax der Autogrammatik entspricht. Für jedes Nichtterminal der Autogrammatik gibt es eine Methode, die jeweils dessen Produktionen prüft und je nach Ergebnis `true` oder `false` zurückgibt. In der Methode `parse` wird die Methode für das Startsymbol `A` der Grammatik aufgerufen. Die rechten Seiten der Produktionen von `A` bestehen aus einem der Terminale `"b1"`, `"br"` oder `"hs"`, gefolgt von einem Nichtterminal. Wenn das aktuelle `token` einen dieser Werte hat, muss die Methode des ihm folgenden Nichtterminals aufgerufen werden, andernfalls wird sofort `false` zurückgegeben. Entsprechend wird weiterverfahren, bis die Tokenliste vollständig abgearbeitet ist. Die Methode `giltA()` gibt `true` zurück, wenn keine der im Laufe der Syntaxanalyse aufgerufenen Methoden `false` zurückgeliefert hat. Andernfalls gibt die Methode `giltA()` `false` zurück und das Wort gehört nicht zur Autosprache.

Beim Aufruf von `parse("hsfsbr")` werden folgende Produktionen „abgearbeitet“

A → hsC
C → fsD
D → brE
E → ε

Beim Parsen des Wortes `"hsfsbr"` werden daher die Methoden `giltA()`, `giltC()`, `giltD()`, und `giltE()` aufgerufen.

Implementation der Methode `giltE`:

```
private boolean giltE() {
    if (token.equals("b1") || token.equals("br")) {
        token = gibNaechstesToken();
        return giltE();
    } else if (token.equals("sp")) {
        token = gibNaechstesToken();
        return giltF();
    } else if (token.equals("")) {
        return true;
    } else {
        return false;
    }
}
```

7. Teilleistungen – Kriterien / Bewertungsbogen zur Prüfungsarbeit

Name des Prüflings: _____ Kursbezeichnung: _____

Schule: _____

Teilaufgabe a)

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK ²	ZK	DK
1	gibt das Wort der Autosprache L(G) für die Abbildung 2 an.	2 (I)			
2	gibt ein Wort der Autosprache für ein Auto mit 9 Bauteilen an.	2 (I)			
3	bestimmt für die beiden zulässigen Autos mit drei Bauteilen die Wörter der Autosprache.	4 (II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (8)					
.....					
.....					
Summe Teilaufgabe a)		8			

Teilaufgabe b)

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
1	zeigt anhand der Grammatik, dass das vorgegebene Wort nicht zur Autosprache gehört.	4 (II)			
2	zeigt mithilfe der Grammatik, dass das vorgegebene Wort zur Autosprache gehört.	6 (II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (10)					
.....					
.....					
Summe Teilaufgabe b)		10			

² EK = Erstkorrektur; ZK = Zweitkorrektur; DK = Drittkorrektur

Teilaufgabe c)

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
1	begründet, dass die Grammatik der Autosprache regulär ist.	4 (II)			
2	entwickelt einen endlichen Automaten, der alle Wörter der Autosprache akzeptiert.	8 (II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (12)					
Summe Teilaufgabe c)		12			

Teilaufgabe d)

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
1	entwickelt einen Algorithmus und berücksichtigt folgende Aspekte: - Zerlegung des Wortes in Paare von Zeichen - Prüfung, ob die Zeichenpaare Terminale der Grammatik sind - Verwaltung der Zeichenpaare in der Tokenliste - Behandlung von Fehlerfällen	6 (II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (6)					
Summe Teilaufgabe d)		6			

Teilaufgabe e)

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
1	analysiert und erläutert die Methode parse.	3 (III)			
2	analysiert und erläutert die Methode giltA.	3 (III)			
3	ermittelt die beim Parsen aufgerufenen Methoden.	3 (II)			
4	implementiert die Methode giltE.	5 (II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (14)					
Summe Teilaufgabe e)		14			