

INFO I TESTKLAUSUR

22.02.2000

Name: _____

Vorname: _____

Matrikelnummer: _____

Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Aufgabe 5	Aufgabe 6
max: 12	max: 9	max: 10	max: 10	max:8	max:11

Aufgabe 1

- a. Geben Sie für folgende Sprachen jeweils eine Grammatik an. Ordnen Sie die angegebenen Grammatiken in die Chomsky-Hierarchie ein. (6 P)

Es gilt jeweils: $G_x = \{ S, N, P, S \}$ mit

$S = \{a, b, c\}$

$N = \{ S, A, B \}$

S ist Startsymbol

$$L_1 = ab^*c$$

$$L_2 = (ab)^n c^{n-1}, n > 1$$

- b. Geben Sie zu folgenden Grammatiken jeweils die erzeugte Sprache an. Geben Sie desweiteren zu jeder Grammatik und zu jeder Sprache den Chomsky Typ an (6 P):

Für alle Grammatiken gilt $G_x = \{ S, N, P, S \}$ mit

$S = \{a, b, c\}$

$N = \{ S, A, B \}$

S ist Startsymbol

$$G_1: P = \{ S \rightarrow A \mid B \mid AB \mid \varepsilon \\ A \rightarrow aA \mid aB \mid a \\ B \rightarrow Bb \mid Ab \mid b \}$$

$$G_2: P = \{ S \rightarrow aB \mid bA \mid a \mid b \\ B \rightarrow bA \mid b \\ A \rightarrow aB \mid a \}$$

Aufgabe 2

Gegeben ist folgendes Datenbankschema, hier dargestellt in Tabellen:

Relation Artikel

Artikelnummer	Bezeichnung	auf Lager
0001	Pudelmütze grün	nein
0002	Pudelmütze rot	ja
0003	Kittelschürze geblümt	nein
0004	Kittelschürze gestreift	ja
0005	Schnürsenkel	ja
0006	Einmachgurken	nein

Relation Bestellung

Nummer	Artikel	Kunde
0001	0001	0001
0002	0003	0001
0003	0005	0002
0004	0001	0001

Relation Kunden

Kundennummer	Name
0001	Lieschen Müller
0002	Sebastian Abeck
0003	Donald Duck

- Geben Sie einen Datenbankausdruck (relationale Algebra) an, um eine komplette Artikelliste zu erzeugen. (3 P)
- Geben Sie einen SQL-Ausdruck an, um zu herauszufinden, welcher Kunde welche Artikel bestellt hat. (3 P)
- Eine Datenbankanfrage stellt fest, welche Artikel bestellt wurden, aber nicht im Lager verfügbar sind, und daher nur verzögert ausgeliefert werden können. Zeichnen Sie das Ergebnis dieser Anfrage als Tabelle. (1 P)

Aufgabe 3

	wahr	falsch
D/A-Wandler arbeiten mittels inverser Fourier-Transformation		
In einer von-Neumann Architektur bezeichnet man den Datenpfad zwischen Speicher und Prozessor als "Flaschenhals".		
Die Laufzeit bei Sortieralgorithmen hängt linear von der Problemgröße ab.		
In einem Baum gibt es immer einen oder mehrere Knoten mit Ausgangsgrad 0		
Auf das Ergebnis eines Programmes, hat die verwendete Bindungsvariante für Variable (statisch/dynamisch) keinen Einfluss.		
Ein Regelkreis beinhaltet immer eine Steuerung.		
Sei A Modell von P_1 und P_2 , und B Modell von P_1 , dann folgt daraus, dass B auch Modell von P_2 sein muss.		
Die Gesamtsumme von Marken in einem Petrinetz ist immer konstant.		
SQL steht für "Synchronous Question Lineup"		
Pseudocode-Compiler erzeugen zumeist schnellere Programme als andere Compiler.		

Aufgabe 4

Gefordert ist eine Funktion $produkt(u,o)$, die das Produkt über einer Folge von natürlichen Zahlen berechnet. Die Zahlenfolge wird spezifiziert, durch eine untere Grenze und eine obere Grenze. Die dazwischen liegenden Folgeglieder unterscheiden sich jeweils um den Wert eins, die Folge ist ansteigend. Gehen Sie davon aus, dass die untere Grenze immer kleiner oder gleich der oberen Grenze ist.

Beispiel: Der Aufruf $produkt(1,4)$ liefert als Ergebnis 24 ($1*2*3*4=24$)
der Aufruf $produkt(4,5)$ liefert das Ergebnis 20 ($4*5$)

- a.) Schreiben Sie die geforderte Funktion iterativ in JAVA (5 P)
- b.) Schreiben Sie die geforderte Funktion rekursiv in GOFER (5 P)

Aufgabe 5

Zeigen Sie durch Umformungen, die Äquivalenz folgender Ausdrücke: (5 P)

$$(a \Rightarrow b) \vee (c \Rightarrow a) \stackrel{!}{=} \neg [(a \wedge \neg b) \wedge c] \vee [a \vee (b \vee \neg a)]$$

Bringen Sie folgenden Ausdruck durch Umformung in DNF: (3 P)

$$a \Leftrightarrow b$$

Aufgabe 6

Um einen Klammerausdruck A auf Korrektheit zu überprüfen, wird dieser Zeichenweise von links nach rechts ausgewertet. Zur Überprüfung der Korrektheit, wird ein Keller benutzt. Sie bekommen ein Programm vorgegeben, bei dem eine Methode *nächstesZeichen(char c)* existiert, die nacheinander mit jedem Zeichen aus dem Klammerausdruck aufgerufen wird. Für die Variable c gilt: `c = '[' | ']' | '(' | ')'`

a. Beschreiben Sie informell ein Verfahren um mit einem Keller einen Klammerausdruck zu verifizieren, in dem verschiedene Klammerarten (rund, eckig,...) vorkommen. (3 P)

b. Folgende Tabelle beschreibt, das Verfahren formal. In der ersten Spalte, steht das jeweilige Eingabezeichen (das Zeichen, das als Parameter übergeben wird), in der zweiten Spalte die auszuführende Kelleroperation, in der letzten Spalte eine Abbruchbedingung. Tritt die Abbruchbedingung ein, so wird der gesamte Klammerausdruck als ungültig gewertet. Sollte ein Kellerunterlauf auftreten (Versuch, ein Element von Keller zu lesen, wenn der Keller leer ist) tritt ein Fehlerfall ein, und der Ausdruck wird ebenfalls als ungültig gewertet. Tritt nie eine Abbruchbedingung oder ein Fehlerfall ein, so wird der Ausdruck als korrekt gewertet. (Terminierung bei Ende der Eingabezeichenkette, Keller muss dann leer sein).

Beispiel:

Eingabe	auszuführende Operation	Abbruchbedingung
'x'	top()	top() == 'a'

Erläuterung: Bei Eingabe 'x' wird die Operation top() ausgeführt. Sollte top() das Zeichen 'a' liefern, wird die Ausführung abgebrochen, der Ausdruck als ungültig gewertet.

Bitte ausfüllen (8 P):

Eingabe	auszuführende Operation	Abbruchbedingung

INFO I TESTKLAUSUR

Lösungsvorschlag

22.02.2000

Name: _____

Vorname: _____

Matrikelnummer: _____

Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Aufgabe 5	Aufgabe 6
max: 12	max: 9	max: 10	max: 10	max:8	max:11

Aufgabe 1

a. Geben Sie für folgende Sprachen jeweils eine Grammatik an. Ordnen Sie die angegebenen Grammatiken in die Chomsky-Hierarchie ein. (6 P)

Es gilt jeweils: $G_x = \{ S, N, P, S \}$ mit

$S = \{a, b, c\}$
 $N = \{ S, A, B \}$
S ist Startsymbol

$$L_1 = ab^*c$$

$G_1 : P = \{ S \rightarrow aB \mid B \rightarrow bB \mid c \}$

CH-3

$$L_2 = (ab)^n c^{n-1}, n > 1$$

$G_2 : P = \{ S \rightarrow abAC \mid A \rightarrow AC \mid ab \mid C \rightarrow c \}$

CH-2

b. Geben Sie zu folgenden Grammatiken jeweils die erzeugte Sprache an. Geben Sie desweiteren zu jeder Grammatik und zu jeder Sprache den Chomsky Typ an (6 P):

Für alle Grammatiken gilt $G_x = \{ S, N, P, S \}$ mit

$S = \{a, b, c\}$
 $N = \{ S, A, B \}$
S ist Startsymbol

$G_1 : P = \{ S \rightarrow A \mid B \mid AB \mid \epsilon \mid A \rightarrow aA \mid aB \mid a \mid B \rightarrow Bb \mid Ab \mid b \}$

$$L_1 = (a^*b^*)$$

G1: CH-2

L1: CH-3

$G_2: P = \{S \rightarrow aB \mid bA \mid a \mid b$
 $B \rightarrow bA \mid b$
 $A \rightarrow aB \mid a \}$

$L_2 = \{b\} (ab)^* \{a\}$

$G_2: CH-3$

$L_2: CH-3$

Aufgabe 2

Gegeben ist folgendes Datenbankschema, hier dargestellt in Tabellen:

Relation Artikel

Artikelnummer	Bezeichnung	auf Lager
0001	Pudelmütze grün	nein
0002	Pudelmütze rot	ja
0003	Kittelschürze geblümt	nein
0004	Kittelschürze gestreift	ja
0005	Schnürsenkel	ja
0006	Einmachgurken	nein

Relation Bestellung

Nummer	Artikel	Kunde
0001	0001	0001
0002	0003	0001
0003	0005	0002
0004	0001	0001

Relation Kunden

Kundennummer	Name
0001	Lieschen Müller
0002	Sebastian Abeck
0003	Donald Duck

- a. Geben Sie einen Datenbankausdruck (relationale Algebra) an, um eine komplette Artikelliste zu erzeugen. (3 P)

$P_A: \Gamma(\text{Artikel}) \rightarrow \Gamma(A)$

mit $A \subseteq \text{Artikel}$

$A = \{\text{Bezeichnung: String}\}$

- b. Geben Sie einen SQL-Ausdruck an, um zu herauszufinden, welcher Kunde welche Artikel bestellt hat. (3 P)

```
SELECT Name, Bezeichnung
FROM Kunden, Artikel, Bestellung
WHERE Kundennummer=Kunde AND Artikel=Artikelnummer;
```

c. Eine Datenbankanfrage stellt fest, welche Artikel bestellt wurden, aber nicht im Lager verfügbar sind, und daher nur verzögert ausgeliefert werden können. Zeichnen Sie das Ergebnis dieser Anfrage als Tabelle. (1 P)

Artikelnummer	Bezeichnung	auf Lager
0001	Pudelmütze grün	nein
0003	Kittelschürze geblümt	nein

Aufgabe 3

	wahr	falsch
D/A-Wandler arbeiten mittels inverser Fourier-Transformation		X
In einer von-Neumann Architektur bezeichnet man den Datenpfad zwischen Speicher und Prozessor als "Flaschenhals".	X	
Die Laufzeit bei Sortieralgorithmen hängt linear von der Problemgröße ab.		X
In einem Baum gibt es immer einen oder mehrere Knoten mit Ausgangsgrad 0	X	
Auf das Ergebnis eines Programmes, hat die verwendete Bindungsvariante für Variable (statisch/dynamisch) keinen Einfluss.		X
Ein Regelkreis beinhaltet immer eine Steuerung.	X	
Sei A Modell von P_1 und P_2 , und B Modell von P_1 , dann folgt daraus, dass B auch Modell von P_2 sein muss.		X
Die Gesamtsumme von Marken in einem Petrinetz ist immer konstant.		X
SQL steht für "Synchronous Question Lineup"		X
Pseudocode-Compiler erzeugen zumeist schnellere Programme als andere Compiler.		X

Aufgabe 4

Gefordert ist eine Funktion $produkt(u,o)$, die das Produkt über einer Folge von natürlichen Zahlen berechnet. Die Zahlenfolge wird spezifiziert, durch eine untere Grenze und eine obere Grenze. Die dazwischen liegenden Folgeglieder unterscheiden sich jeweils um den Wert eins, die Folge ist ansteigend. Gehen Sie davon aus, dass die untere Grenze immer kleiner oder gleich der oberen Grenze ist.

Beispiel: Der Aufruf $produkt(1,4)$ liefert als Ergebnis 24 ($1*2*3*4=24$)
der Aufruf $produkt(4,5)$ liefert das Ergebnis 20 ($4*5$)

a.) Schreiben Sie die geforderte Funktion iterativ in JAVA (5 P)

```
public int produkt (int u, int o) {  
    int result=u;  
    for (int i=u+1, i<=o, i++) {  
        result=result*i;  
    }  
    return result;  
}
```

b.) Schreiben Sie die geforderte Funktion rekursiv in GOFER (5 P)

```
produkt: Int->Int->Int  
produkt u o = if u == o then u else u*produkt u+1 o
```

Aufgabe 5

Zeigen Sie durch Umformungen, die Äquivalenz folgender Ausdrücke: (5 P)

$$(a \Rightarrow b) \vee (c \Rightarrow a) \quad ! \quad = \quad \neg [(a \wedge \neg b) \wedge c] \vee [a \vee (b \vee \neg a)]$$

$$\begin{aligned} \neg a \vee b \vee \neg c \vee a &= \neg [(a \wedge \neg b) \wedge c] \vee a \vee b \vee \neg a \\ (\neg a \vee a) \vee \neg c \vee b &= \neg [(a \wedge \neg b) \wedge c] \vee b \vee (a \vee \neg a) \\ \text{True} \vee \neg c \vee b &= \neg [(a \wedge \neg b) \wedge c] \vee b \vee \text{True} \\ \text{True} &= \text{True} \end{aligned}$$

Bringen Sie folgenden Ausdruck durch Umformung in DNF: (3 P)

$$a \Leftrightarrow b$$

$$(\neg a \vee b) \wedge (\neg b \vee a)$$

$$[(\neg a \vee b) \wedge \neg b] \vee [(\neg a \vee b) \wedge a]$$

$$[(\neg a \wedge \neg b) \vee (b \wedge \neg b)] \vee [(\neg a \wedge a) \vee (b \wedge a)]$$

$$(\neg a \wedge \neg b) \vee (b \wedge a)$$

Aufgabe 6

Um einen Klammerausdruck A auf Korrektheit zu überprüfen, wird dieser Zeichenweise von links nach rechts ausgewertet. Zur Überprüfung der Korrektheit, wird ein Keller benutzt. Sie bekommen ein Programm vorgegeben, bei dem eine Methode *nächstesZeichen(char c)* existiert, die nacheinander mit jedem Zeichen aus dem Klammerausdruck aufgerufen wird. Für die Variable c gilt: $c = '[' \mid ']' \mid '(' \mid ')'$

- a. Beschreiben Sie informell ein Verfahren um mit einem Keller einen Klammerausdruck zu verifizieren, in dem verschiedene Klammerarten (rund, eckig,...) vorkommen. (3 P)

Lege öffnende Klammern auf den Keller. Bei einer schliessenden Klammer, nehme das oberste Element vom Keller, und vergleiche, ob es eine schliessende Klammer der gleichen Klammerart ist, wie die öffnende Klammer. Falls nicht, oder falls Keller leer ist, breche ab, dann ist der Ausdruck ungültig. Wurde der Ausdruck abgearbeitet, und ist der Keller leer, so ist der Ausdruck gültig.

- b. Folgende Tabelle beschreibt, das Verfahren formal. In der ersten Spalte, steht das jeweilige Eingabezeichen (das Zeichen, das als Parameter übergeben wird), in der zweiten Spalte die auszuführende Kelleroperation, in der letzten Spalte eine Abbruchbedingung. Tritt die Abbruchbedingung ein, so wird der gesamte Klammerausdruck als ungültig gewertet. Sollte ein Kellerunterlauf auftreten (Versuch, ein Element von Keller zu lesen, wenn der Keller leer ist) tritt ein Fehlerfall ein, und der Ausdruck wird ebenfalls als ungültig gewertet. Tritt nie eine Abbruchbedingung oder ein Fehlerfall ein, so wird der Ausdruck als korrekt gewertet. (Terminierung bei Ende der Eingabezeichenkette, Keller muss dann leer sein).

Beispiel:

Eingabe	auszuführende Operation	Abbruchbedingung
'x'	top()	top() == 'a'

Erläuterung: Bei Eingabe 'x' wird die Operation top() ausgeführt. Sollte top() das Zeichen 'a' liefern, wird die Ausführung abgebrochen, der Ausdruck als ungültig gewertet.

Bitte ausfüllen (8 P):

Eingabe	auszuführende Operation	Abbruchbedingung
'('	push('(')	false
'['	push('[')	false
)'	pop()	pop() != '('
]'	pop()	pop() != '['