

Klausur Informatik I

20.02.2004

Prof. Dr. G. Goos
Dipl.-Inform. T. Gelhausen

Vorname:	<input type="text"/>
Nachname:	<input type="text"/>
Matrikelnummer:	<input type="text"/>

Zur Klausur sind keine Hilfsmittel und kein eigenes Papier zugelassen. Die Bearbeitungszeit beträgt 60 Minuten.

Die Klausur ist komplett und geheftet abzugeben. Nur Blätter, die Namen und Matrikelnummer tragen, gehen in die Bewertung ein.

Sie dürfen die Rückseite der Aufgabenblätter als Konzeptpapier benutzen. Nur Lösungen, die sich auf dem entsprechenden Aufgabenblatt oder seiner Rückseite befinden, gehen in die Bewertung ein.

Aufgabe	1	2	3	4	5	6	7	Σ
Maximal	8	8	6	9	6	11	12	60
K1								
K2								
K3								

Punkte:

Note:

Aufgabe 1: Wissensfragen (5+3=8P)

a.) Kreuzen Sie an, ob die Aussage Wahr oder Falsch ist. (5P)

Hinweis: Jedes korrekte Kreuz zählt 0,5 Punkte, jedes falsche Kreuz bewirkt 0,5 Punkte Abzug! Die Teilaufgabe wird mindestens mit 0 Punkten bewertet.

WAHR	FALSCH	Aussage
		$g \in o(f(n)) \Leftrightarrow g$ wächst mindestens so schnell wie f
		Eine Transition eines Petri-Netzes kann mehrere Stellen als Nachfolger haben.
		In Prolog gibt es Fakten, Köpfe, Regeln, Klausuren und Körper.
		Das Diamantenlemma macht eine Aussage über die Erfüllbarkeit einer prädikatenlogischen Formel.
		Ein Semi-Thue-System ist ein endliches, deterministisches Textersetzungssystem.
		Mit einer 32-Bit Ganzzahldarstellung lassen sich mehr unterschiedliche Zahlenwerte darstellen, als mit einer Gleitpunktzahl einfacher Länge nach IEEE 754-1985.
		Im λ -Kalkül werden Wahrheitswerte durch Funktionen ausgedrückt.
		Die Suche eines Elementes in der Datenstruktur Keller hat den Aufwand $O(n)$.
		Die funktionale Sprache Haskell verwendet faule, die funktionale Sprache ML strikte Auswertung.
		Eine Audio-CD enthält transiente, digitale Signale und die Signale, die aus den Lautsprechern kommen, sind persistent und analog.

b.) Kreuzen Sie an, welche Eigenschaft von welcher Struktur erfüllt werden muss. (3P)

Eigenschaft	abelsche Halbgruppe	Monoid	Boolesche Algebra
Algebraische Abgeschlossenheit			
Assoziativitätsgesetz			
Distributivgesetz			
Dreiecksungleichung			
Einselement			
Kommutativgesetz			
Modularität			

Aufgabe 2: Endliche Automaten (1+4+1+2=8P)

Gegeben seien zwei positive Binärzahlen. Die Darstellungen seien gleich lang, die Größe der Zahlen selbst beliebig, jede Zahl ist mit mindestens einer führenden Null dargestellt.

- a.) Welche Sorte vollständiger, deterministischer, endlicher Automaten benötigen Sie zum Berechnen der Summe beliebig langer Binärzahlen? (1P)

- b.) Entwerfen Sie einen solchen Automaten, der die Summe der beiden Binärzahlen berechnet. Als Eingabe erhält er zuerst die niederwertigsten Bits der beiden Zahlen, dann die zweit-niederwertigsten, und so weiter. (4P)

Beispiel: $z_1=0011$, $z_2=0110$, ergibt als Eingabe [(1,0), (1,1), (0,1), (0,0)]

- c.) Welchen Aufwand hat die Berechnung mit dem obigen Automaten? Wovon hängt die Berechnungszeit ab? (1P)

- d.) Der bei Teilaufgabe c.) angegebene Aufwand bezieht sich auf die Addition beliebig langer Zahlen. Wie groß ist dementsprechend der Aufwand für den darauf aufbauenden Multiplikationsalgorithmus für Dualzahlen, wenn man annimmt, dass alle Zahlen gleichverteilt als Operanden auftreten (mit Begründung)? (2P)

Aufgabe 3: Abstrakte Datentypen (2+2+1+1=6P)

Gegeben sei ein Modul Zoo durch

```
module Zoo where
data Katze a = Hund | Maus (Katze a) a (Katze a) deriving (Eq, Show)
loewe (Maus flo reh ziege) = flo
elefant Hund = True
ratte (Maus pferd schwein gans) = gans
vogel (Maus schaaf otter rind) = otter
elefant (Maus spinne fisch wolf) = False
```

a.) Geben Sie die Signaturen von loewe, ratte, vogel und elefant an. (2P)

b.) Reduzieren Sie folgende Ausdrücke mit Hilfe der Axiome (2P)

*loewe (Maus (Maus Hund "Kalb" Hund) "Frosch" Hund)**vogel (ratte (Maus (Maus Hund "Gnu" Hund) "Pfau" (Maus Hund "Kara" Hund)))*

c.) Welche Tiere exportiert der Modul? (1P)

d.) Welchem bekannten ADT entspricht der Typ Katze? (1P)

Aufgabe 4: Chomsky (1+6+1+1=9P)

Gegeben sei eine Grammatik $G=(\Sigma, N, P, S)$ mit folgenden sieben Produktionen P:

1. $S \rightarrow B \mid aBDc$
2. $B \rightarrow bbEbb \mid aBDc$
3. $Dc \rightarrow cD$
4. $bD \rightarrow b$
5. $bcD \rightarrow bc$
6. $ccD \rightarrow cc$
7. $E \rightarrow b \mid bE \mid Eb$

- a.) Von welchem CH-Typ ist die Grammatik (mit Begründung)? (Geben Sie den höchstmöglichen Typ an!) (1P)

- b.) Geben Sie die erzeugte Sprache $\mathcal{L}(G)$ an. Von welchem Chomsky-Typ ist sie? (Geben Sie den höchstmöglichen Typ an!) (6P)

$\mathcal{L}(G) =$

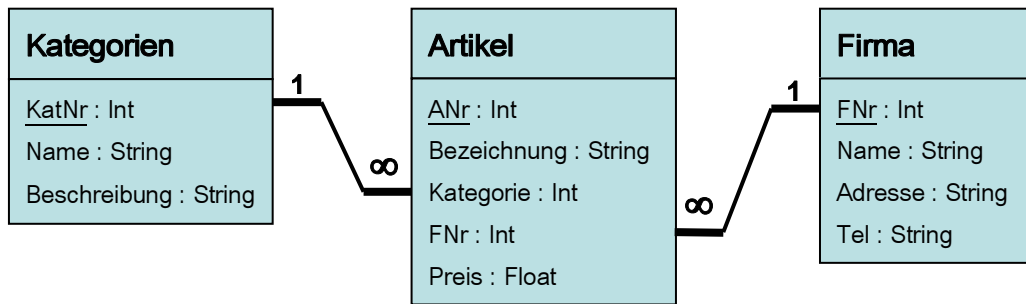
Die Sprache ist vom Typ:

- c.) Geben Sie die Menge Q der Regeln an (*Angabe der Indices*), die verhindern, dass G vom CH-Typ ist, der dem Chomsky-Typ von $\mathcal{L}(G)$ entspricht. (1P)

Q =

- d.) Sei $P' = P \setminus Q$. Erweitern Sie P' so zu P'' , dass $\mathcal{L}((\Sigma, N, P'', S)) = \mathcal{L}(G)$. (1P)

$P'' = P' \cup$

Aufgabe 5: Relationale Algebra und SQL (2+2+2=6P)

Gegeben sei obiges Schema und folgender Ausdruck in relationaler Algebra:

$\pi_{\text{Bezeichnung, Preis}}(\sigma_{(\text{Kategorien.Name} = \text{'Stifte'} \wedge \text{Preis} < 10.00 \wedge \text{Kategorien.KatNr} = \text{Artikel.Kategorie})}(\text{Artikel} \times \text{Kategorien}))$

a.) Geben Sie zu dem gegebenen Ausdruck einen äquivalenten Ausdruck in SQL an. (2P)

b.) Geben Sie zu dem gegebenen Ausdruck den Anfrage-Baum an. (2P)

Name: _____

Matrikelnummer: _____

- c.) Aus welchen Gründen kann folgende Anfrage nicht in einen äquivalenten Ausdruck der relationalen Algebra umgeformt werden (mit Begründung)? (2P)

```
SELECT *
FROM (
    SELECT a AS p, b AS q, c AS r
    FROM m
    WHERE a=3
    AND b=4
    UNION ALL
    SELECT x AS p, y AS q, z AS r
    FROM n
    WHERE x=3
    AND y=4
)
ORDER BY p, q, r;
```

Aufgabe 6: Graphen und Relationen (1+4+5+1=11P)

Gegeben sei folgender Graph:

$$E : \{ a, b, c, d, e \}$$

$$K : a \quad [a, b]$$

$$b \quad [b, d, e]$$

$$c \quad [c, d]$$

$$d \quad [d]$$

$$e \quad [a, e]$$

a.) Wie nennt man diese Darstellungsform? (1P)

b.) Es handelt sich um einen (1P) (*bitte Ankreuzen*)

- nicht zusammenhängenden Graphen.
- schwach zusammenhängenden Graphen.
- einseitig zusammenhängenden Graphen.
- stark zusammenhängenden Graphen.

Es lassen sich insgesamt (2P) (*bitte Anzahl eintragen*)

- schwache Zusammenhangskomponenten identifizieren,
- einseitige Zusammenhangskomponenten identifizieren und
- starke Zusammenhangskomponenten mit jeweils mehr als einem Knoten identifizieren.

Der Graph beschreibt eine (1P) (*bitte Ankreuzen*)

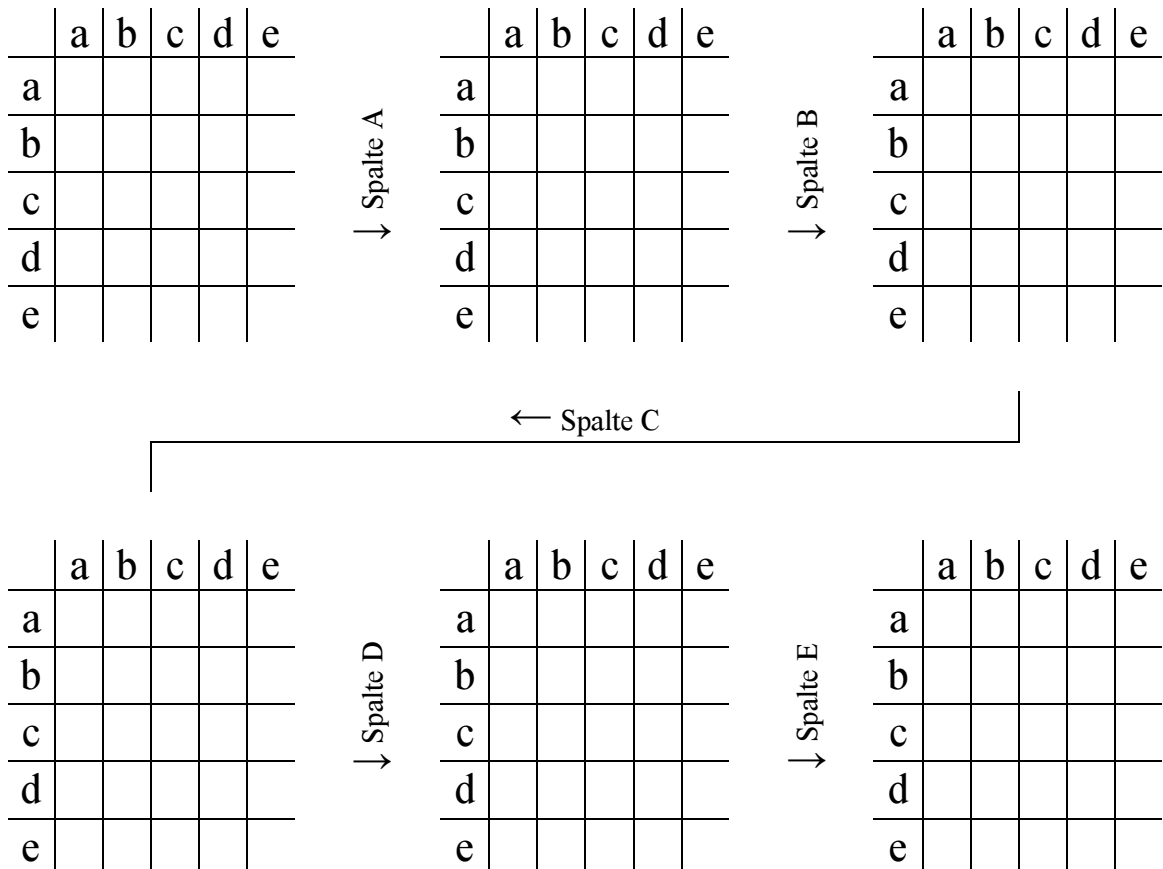
- reflexive, nichttransitive Relation.
- transitive, nichtreflexive Relation.
- reflexive und transitive Relation.
- Relation, die weder reflexiv noch transitiv ist.

- c.) Führen Sie den Floyd-Warshall-Algorithmus für die durch den Graphen gegebene Relation durch. Tragen Sie dazu die Ursprungsrelation in das erste angegebene Feld ein. Führen Sie dann den Algorithmus der Reihe nach für die Spalten a, b, c, d und e durch und tragen Sie die jeweiligen Zwischenergebnisse in die dafür vorgesehenen Felder ein. Markieren Sie jeweils die hinzugekommene(n) Kante(n) durch Einkreisen! (5P)

Hinweis: Nur mit Angabe der jeweiligen Zwischenschritte gibt es Punkte. Es reicht also nicht, nur das Endergebnis hinzuschreiben!

Hilfe: Um Hin- und Herblättern zu vermeiden, sei der Graph der Aufgabe an dieser Stelle nochmals angegeben:

$E : \{ a, b, c, d, e \}$
 $K : a \quad [a, b]$
 $\quad b \quad [b, d, e]$
 $\quad c \quad [c, d]$
 $\quad d \quad [d]$
 $\quad e \quad [a, e]$



- d.) Welchen Aufwand hat der Floyd-Warshall-Algorithmus? (1P)

Aufgabe 7: Haskell (5+2+5=12P)

Gegeben sei eine Sprache $\mathcal{L}_{m,n}$, deren Wörter immer abwechselnd m gleiche Buchstaben, dann n gleiche (jedoch von den vorhergehenden m verschiedenen) Buchstaben, dann wieder m , dann n und so weiter haben.

Mit anderen Worten die Sprache hat die Form $a^m b^n c^m d^n e^m f^n \dots$ mit $a, b, c, d, e, f, \dots \in \Sigma$, wobei $a \neq b, b \neq c, c \neq d, d \neq e, \dots$, und $m, n \geq 0$.

Beispiel: $a^2 \in \mathcal{L}_{2,5}$, $a^2 b^5 \in \mathcal{L}_{2,5}$, $a^2 b^5 c^2 \in \mathcal{L}_{2,5}$, $a^2 b^5 c^2 a^5 \in \mathcal{L}_{2,5}$,
 $a^2 b^5 c^3 \notin \mathcal{L}_{2,5}$, und insbesondere $a^2 a^5 c^2 a^5 = a^7 c^2 a^5 \notin \mathcal{L}_{2,5}$

Hinweis: Wenn Sie die Signatur einer Funktion angeben, dürfen Sie diese Funktion in den späteren Teilaufgaben verwenden, auch wenn Sie keine Implementierung angeben konnten.

- a.) Schreiben Sie eine Haskell-Funktion `gleicheweg`, die das längste Präfix p einer Zeichenkette xs bestimmt, wobei p ausschließlich aus identischen Buchstaben besteht. Geben Sie die Länge von p sowie den verbleibenden Rest von xs zurück, also beispielsweise für $xs="aaasdf"$ die Werte 3 und "sdf". Die Funktion soll einen maximal polynomialen Aufwand haben. (5P, bei höherem als polynomiellen Aufwand max. 4P)

- b.) Geben Sie eine Haskell-Funktion `laengenliste` an, die mit der Funktion aus Teilaufgabe a.) eine Zeichenkette in eine Liste von Zahlen verwandelt, wobei jeder Eintrag für die Länge einer Folge konsekutiver, gleicher Zeichen steht. Die Reihenfolge der Längeneinträge in der Liste soll der Reihenfolge der zugehörigen Zeichenfolgen entsprechen. (2P)

Name: _____

Matrikelnummer: _____

- c.) Bauen Sie mit Hilfe der zuvor definierten Funktion(en) eine Haskell-Funktion abwechslung, die in Abhängigkeit von der Zugehörigkeit des Eingabewortes zur gegebenen Sprache einen Booleschen Wert zurückgibt; und zwar ‚Wahr‘, wenn das Wort zu der gegebenen Sprache gehört und sonst ‚Falsch‘. (5P)

