Universität Karlsruhe

Informatik 1 WS 00/01

Institut für Telematik, Forschungsgruppe C&M
Prof. Dr. S. Abeck, R. Scholderer

Abgabe bis: 12.12.2000 14:00 Uhr

Übungsblatt 7

Aufgabe

1. Relationale Algebra - Entwicklung von SQL-Anfragen (0,5 Punkte)

Gegeben ist der folgende Ausschnitt aus einem Studierenden-Datenbestand:

Relation:**Studierende**

Name	**Vorname**	**Mat-Nr**	**Fach**	**Semester**
Schulz	Karin	123451	Inf	1
Mülle	Susanne	123452	Inf	1
Schmidt	Dieter	123453	Inf	2
Maier	Klaus	123454	Inf	3
Hofmann	Peter	123457	Inf	4
Müller	Peter	234571	Bio	5
Wehner	Chris	245487	Bio	7
Wolf	Kirsten	245874	WiWi	9

a) Entwerfen Sie SQL-Anfragen, die als Ergebnis die folgenden Zielrelationen haben:

1.

Name	**Mat-Nr**	**Semester**
Schulz	123451	1
Mülle	123452	1
Schmidt	123453	2
Wolf	245874	9

2.

Name	Vorname
Schulz	Karin
Mülle	Susanne
Schmidt	Dieter
Maier	Klaus
Wehner	Chris

b) Geben Sie das Schema zur Relation Studierende an.

Aufgabe

2. Term der booleschen Algebra (1,5 Punkte)

Gegeben ist die in der Vorlesung eingeführte Boolesche Algebra

B = *B*(, ⊤ , ⊥ , ☾ , ∨ , ∧) und folgender Term: ((N ∧ ⊤) ∨ ⊥) ∧ ☾ (P ∨ ☾ W).

a. Was bedeutet $\Sigma^{(0)}$? Welche Elemente aus $\Sigma^{(0)}$ kommen im gegebenen Term vor?
b. Zeigen Sie, daß der gegebene Term korrekt ist, also den in der Vorlesung eingeführten Bedingungen für korrekte Terme entspricht.
c. Geben Sie den Kantorovic-Baum für den Term an.

Aufgabe

3. Boolesche Algebra (1 Punkt)

Gegeben sind zwei Formeln:

1) (((a ∧ b) ∨ (☾ a ∧ ☾ b)) ∧ c) ∨ (☾ ((a ∧ b) ∨ (☾ a ∧ ☾ b)) ∧ ☾ c)

2) (a ∧ b ∧ c) ∨ (☾ a ∧ ☾ b ∧ c) ∨ (☾ a ∧ b ∧ ☾ c) ∨ (a ∧ ☾ b ∧ ☾ c)

a. Zeigen Sie durch Umformung mit den Gesetzen der Booleschen Algebra, dass die erste Formel gleichbedeutend mit der Formel ist.
b. Zeigen Sie die Gleichheit beider Formeln durch eine Wertetabelle.

Aufgabe

4. Keller - TripelKeller (1 Punkt)

Entwerfen Sie eine abstrakte Algebra TripelKeller, die wie ein Keller arbeitet mit dem Unterschied, dass jeweils ein Tripel von Elementen, auf den TripelKeller geschrieben (push), gelesen (top) bzw. gelöscht (pop) wird.
Geben Sie zu der Algebra die Signatur $\Sigma_{\text{TripelKeller}}$ und die Axiome an.
Geben Sie den Ausdruck an, der folgenden graphisch skizzierten Wertverlauf eines TripelKellers beschreibt.

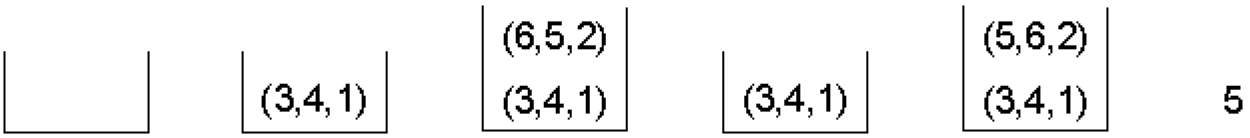

Abbildung zu Aufgabe: *TripelKeller*

Hinweis: Definieren Sie für anstatt einer top-Funktion die drei Funktionen top1, top2 und top3. Dabei soll top1 das erste Element des Tripels, top2 das zweite und top3 das dritte Element des Tripels ausgeben.

Aufgabe

5. Keller - Auswertung von Kellern (1 Punkt)

Gegeben sind unterschiedliche Keller.

a) Von welchem Typ sind die Keller, die von folgenden Kellerausdrücken erzeugt werden?

> (Mit "Typ" ist hier der Typ der Kellerelemente gemeint. Wie in einer Programmiersprache können Typen z.B. Integer, Character, Double sein.)

```
  I.   top(push(push(CreateStack,1),2))
 II.   empty(push(CreateStack,false))
III.
       push(push(pop(push(push(CreateStack,"Hof"),"Andi")),"Joa"),"H
       an")
```

b) Gehen Sie jetzt von einem Keller vom Typ Integer Keller aus. Vereinfachen Sie folgende Ausdrücke unter Verwendung der aus der Vorlesung bekannten Axiome soweit wie möglich.

```
  I.   empty(pop(push(push(pop(push(CreateStack,2)),99),1)))
 II.   top(push(push(push(push(CreateStack,1),7),6),2))
III.
       push(push(push(CreateStack,2),top(pop(push(push(CreateStack,1
       ),3)))),1)
```

Die Ausdrücke aus b) lassen sich auch mit dem Multimediawerkzeug "Stack" (auf der Startseite des Lernsystems unter Animationen zu finden) auswerten. Dabei wird die Funktionsweise eines Kellers deutlich.

c) Gibt es bei b) I) nur eine Möglichkeit, den Kellerausdruck auszuwerten? Wenn nein, geben Sie eine weitere an.

d) Geben Sie ein Auswertungsverfahren, welches für Kellerausdrücke, bei denen die Operationen top oder empty ganz außen stehen, eine möglichst schnelle Auswertung durchführt. Formulieren Sie ihr Verfahren als Pseudocode.

> *Hinweis:* Überlegen Sie, ob eine Auswertung von innen nach außen oder von außen nach innen schneller ist.
> Verwenden Sie die Operationen `auswertbar(kt)`(Prüft ob, auf Operationen der Klammertiefe kt ein Axiom anwendbar ist.) und
> `auswerten(kt)` (Wendet auf Operation der Klammertiefe kt ein Axiom an, sofern es möglich ist.)
> für Ihren Pseudocode.

Aufgabe

6. Sortierverfahren und Aufwandsabschätzung (1,5 Punkte)

Wir gehen von dem in der Vorlesung (Abschnitt 3.1) behandelten Sortierverfahren durch Einsortieren aus und wandeln es folgendermaßen ab: Wähle das kleinste Element im Stapel x und füge es zum Stapel y hinzu.

a. Formulieren Sie den Algorithmus in möglichst strukturiertem Pseudo-Code. (Bei den beiden Stapeln x und y kann das i-te Element durch x[i] bzw y[i] angesprochen werden, x.del(i) bzw. y.del(i) löscht das i-te Element des Stapels x bzw. y.)
b. Zeigen Sie die Arbeitsweise dieses Verfahrens anhand des Stapelinhalts von x = [6, 2, 9, 5] auf (siehe Vorlesung).
c. Geben Sie in einer Tabelle an, wie viele und welche Vergleiche (Vergleiche zwischen Stapelelementen oder von Schleifenvariablen) und Zuweisungen (Zuweisungen für Stapelelemente, Hilfsvariablen und Schleifenvariablen) der Algorithmus benötigt für den Stapel x aus Aufgabe b) benötigt. Zählen Sie Löschoperationen als Zuweisungen.

Aufgabe

7. Java: Sortieren durch Vertauschen (2,5 Punkte)

Wir gehen von dem in der Vorlesung behandelten Sortierverfahren durch Vertauschen aus.

a. Formulieren Sie den Algorithmus in Pseudo-Code. Wie in der vorherigen Aufgabe kann das i-te Element des Stapels x durch x[i] angesprochen werden.
b. Überführen Sie den Algorithmus in ein lauffähiges Java-Programm, welches Stapel der Länge 4 einliest und mit diesem Verfahren sortiert.

   Ergänzen Sie zwei Integer-Variablen, die folgendes zählen:

(1) Anzahl der durchgeführten Vergleiche

(2) Anzahl der durchgeführten Zuweisungen

Ausgegeben werden sollen der sortierte Stapel, sowie die Anzahl von Zuweisungen und Vergleichen.

Testen Sie Ihren Algorithmus mit dem Eingabestapel x=[6, 2, 9, 5]. .

Aufgabe

8. Türme von Hanoi (1 Punkt)

Gegeben seien n Scheiben unterschiedlichen Durchmessers. Die Scheiben seien der Größe nach zu einem Turm (Platz a) geschichtet, die Größte zuunterst.

Die Aufgabe lautet, den Turm vom Platz a auf einen zweiten Platz b zu verlegen. Als Zwischenablage steht ein dritter Platz c zur Verfügung.

Es darf stets nur die oberste Scheibe eines Turms bewegt werden. Zu keiner Zeit darf auf einem der drei Plätze eine kleinere Scheibe unter einer größeren liegen.

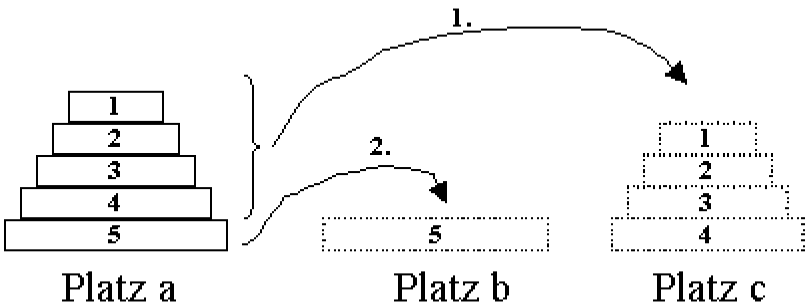

Abbildung zu Aufgabe *Türme von Hanoi*

a. Beschreiben Sie ein Verfahren zur Lösung dieser Aufgabe.

   (*Hinweis:* Man verwende einen Induktiven Ansatz. Unter der Vorraussetzung dass die Lösung des Problems für n-1 Scheiben bekannt ist, gebe man die Lösung für n Scheiben an.)
b. Wieviele elementare Schritte (Verlegen einer Scheibe von einem Platz zum Nächsten) benötigen Sie, um das Problem für n Scheiben zu lösen.

   (*Hinweis:* Stellen Sie eine rekursive Gleichung Anz(n) auf, mit der die Anzahl der elementaren Schritte für n Scheiben mit Kenntnis der Lösung für n-1 Scheiben berechnet werden kann. Suchen Sie eine geschlossene Form für diese Gleichung und beweisen Sie diese mit vollständiger Induktion.)
c. Ist Ihr Verfahren deterministisch?
