Fachbereich
Informatik

Klausur Informatik 1
12. Oktober 2000

Mahr / Lemke
Weisweber / Tschirley

Name: .....

Vorname: .....

Matr.-Nr.: .....

Bearbeitungszeit: 120 Minuten

- Benutzen Sie für die Lösung der Aufgaben nur das mit diesem Deckblatt ausgeteilte Papier. **Lösungen, die auf anderem Papier gerschrieben werden, können nicht gewertet werden!**
- Schreiben Sie Ihre Lösungen auf das Aufgabenblatt der jeweiligen Aufgabe. Verwenden Sie auch die Rückseite. **Schreiben Sie keine Lösungen einer Aufgabe auf ein Blatt, das nicht zu dieser Aufgabe gehört!**
- Schreiben Sie deutlich! Doppelte, unleserliche oder mehrdeutige Lösungen können nicht gewertet werden.
- Schreiben Sie nur in **blau** oder **schwarz**. Schreiben Sie nicht mit Bleistift!
- Bei der Implementation von OPAL-Funktionen brauchen *keine* Importe mit angegeben zu werden, wenn Funktionen aus dem gegebenen Teil der Bibliotheca Opalica (oder aus der Struktur `Nat`) oder sonstigen (in den Aufgaben) gegebenen Strukturen benutzt werden!
- Die OPAL-Implementierungen müssen nicht dokumentiert werden.
- Erscheint Ihnen eine Aufgabe mehrdeutig, wenden Sie sich an die Betreuer.
- Tragen Sie jetzt (vor Beginn der Bearbeitungszeit) auf *allen* Blättern Ihren Namen und Ihre Matrikelnummer ein.

Aufgabe	Punkte	erreicht
1	6	
2	6	
3	7	
4	7	
5	7	
6	6	
7	10	

1. Aufgabe (6 Punkte): Softwareentwicklungszyklus

Beschreiben Sie den grundlegenden Softwareentwicklungszyklus

a) unter Verwendung eines Compilers und

b) unter Verwendung eines Interpretierers

in Form eines Ablaufdiagramms oder in Worten. Geben Sie dabei die verwendeten Werkzeuge und deren Ein- und Ausgabedaten an.

2. Aufgabe (6 Punkte): Abstrakte Datentypen

Definieren Sie einen abstrakten Datentypen `grad` mit passendem Signatur- und Implementierungsteil, der Berechnungen und Konvertierungen in die drei Temperatureinheiten Grad Celsius (`inC`), Kelvin (`inK`) und Grad Fahrenheit (`inF`) ermöglicht. Dabei ist 0 Grad Kelvin der absolute Nullpunkt und entspricht -273,16 Grad Celsius. 32 Grad Fahrenheit ist der Gefrierpunkt, d.h. entspricht 0 Grad Celsius und 212 Grad Fahrenheit der Siedepunkt, d.h. 100 Grad Celsius. Die Umrechnungsformel lautet also $c = (f - 32)/1.8$, wenn c die Temperatur in Grad Celsius und f die in Grad Fahrenheit ist. Jede Temperatureinheit soll darstellbar sein (z.B. `25 C`, `28 K` oder `31 F`) und in die jeweils anderen Einheiten konvertierbar sein. Als Operationen auf den Temperatureinheiten sollen `+`, `/` und `durchschnitt` realisiert werden, wobei `+` zwei Temperatureinheiten als Argumente nimmt und `/` eine Temperatureinheit als erstes Argument und einen **nat**-Wert als zweites. Die Funktion `durchschnitt` berechnet die Durschnittstemperatur aus einer vorzugebenden Liste von Temperaturen.

Hinweise:

- Als Datentyp `grad` soll ein Produktyp definiert werden, der die Temperatur in Grad Celsius angibt und **kein Summentyp**, der aus allen drei Temperatureinheiten besteht. Grad Kelvin und Grad Fahrenheit sind dann als einstellige Funktionen `K` bzw. `F` von `real` nach `grad` zu realisieren.
- Die Funktion `#` aus der Struktur `Seq[alpha]` mit der Signatur `FUN # : seq[alpha] -> nat` kann zur Bestimmung der Länge einer Liste verwendet werden.
- Die Funktion `asReal` aus der Struktur `NatConv` mit der Signatur `FUN asReal : nat -> real` kann zur Konvertierung von `nat` nach `real` verwendet werden.

3. Aufgabe (7 Punkte): Datentypen und Sortieren

3.1. Definition (2 Punkte) Definieren Sie einen neuen Datentypen `persdaten`, der insgesamt Informationen über den Vornamen (`denotation`), Nachnamen (`denotation`), Straße (`denotation`), Hausnummer (`nat`), Postleitzahl (`nat`) und den Wohnort (`denotation`) einer Person aufnehmen kann. Dabei sollen Vor- und Nachname sowie die Adressdaten (Straße, Hausnummer, Postleitzahl und Wohnort) jeweils zu eigenen Datentypen `name` bzw. `adresse` zusammengefaßt werden.

3.2. Induzierte Signatur (3 Punkte) Geben Sie die vollständigen Signaturen an, die durch die Deklaration der Datentypen `persdaten`, `name` und `adresse` induziert werden und geben Sie die Kategorien der einzelnen Funktionen an (Konstruktor, Selektor, Diskriminator).

3.3. Sortieren von Adressen (2 Punkte) Schreiben Sie eine Funktion, die eine Liste von Personen des Typs `persdaten` nach den Nachnamen sortieren kann. Als Sortierverfahren soll der Quick-Sort verwendet werden:

a) Nimm das erste Element e einer Liste.

b) Teile die Restliste r in zwei Listen k und g auf, wobei k sämtliche Elemente enthält, die kleiner oder gleich den Element e sind, und g alle Elemente, die größer sind als e.

c) Sortiere die beiden Listen k und g (natürlich wieder mit Quick-Sort).

d) Die sortierte Gesamtliste enthält man durch Konkatenation der sortierten Liste k, des Elements e und der sortierten Liste g in der angegebenen Reihenfolge.

4. Aufgabe (7 Punkte): Rekursion und Termination

4.1. **(3 Punkte)** In funktionalen Sprachen wie Opal ist die Rekursion die wichtigste Programmiertechnik. Dabei ist besonders auf die Termination zu achten. Bitte geben Sie an, auf was Sie achten müssen, damit ein (funktionales) Programm terminiert.

4.2. **(4 Punkte)** Sehen Sie sich die folgenden beiden Funktionen an:

```
FUN .. : nat ** nat -> seq[nat]
DEF n .. m ==
        IF n > m THEN <>
        ELSE n :: ((n+1) .. m)
        FI

FUN funktion : nat ** seq[nat] -> bool
DEF funktion(_,<>) == false
DEF funktion(el1,el2::rest) ==
        IF el1 = el2 THEN true
        ELSE funktion(el1,rest)
        FI
```

a) Beschreiben Sie, was die Funktionen `..` und `funktion` machen (Semantik).

b) Begründen Sie, ob und, wenn ja, warum die Funktionen terminieren. Der Nachweis kann in eigenen Worten erfolgen. Ein formaler Beweis ist möglich, aber nicht notwendig.

5. Aufgabe (7 Punkte): Funktionen höherer Ordnung

Deklarieren und definieren Sie die folgenden drei Funktionen höherer Ordnung:

5.1. Folgen (2 Punkte) eine Funktion `folge`, die aus einer Funktion f mit Signatur `nat -> nat` eine Folge von $f(0)$ bis $f(n)$ erzeugt. Die Folge soll so lang sein, bis $f(n)$ eine anzugegebende Schranke überschritten hat. Die Funktion soll aufgerufen werden können als `folge(f)(schranke)`.

5.2. Verschiebung einer mathematischen Funktion (2 Punkte) eine Funktion `verschieben`, die eine Funktion f mit Signatur `real -> real` in x- und y-Richtung verschiebt. Die Funktion soll aufgerufen werden können als `verschieben(dx,dy)(f)(x)`, wobei `dx` und `dy` die Verschiebung angeben.

5.3. Listenfunktionale (3 Punkte) eine Funktion `nachfahren`, die für eine Person der Sorte `person` sämtliche Nachfahren (Kinder und Kindeskinder) berechnet. Die Person selbst gehört mit zu dieser Menge. Benutzen Sie zur Lösung Listenfunktionale. `person` sei dabei definiert als:

```
DATA person == person(
        name : denotation,
        kinder : seq[person])
```

6. Aufgabe (6 Punkte): Boolesche Algebra

6.1. **(3 Punkte)** Beweisen oder widerlegen Sie folgende Gleichung durch Anwendung der bekannten Rechenregeln (nicht durch eine Wertetabelle). Sorgen Sie dafür, daß die Umformungen nachvollziehbar sind!

$$(a \cdot b) + (\overline{a} \cdot \overline{b}) \quad = \quad \overline{(\overline{a} \cdot b) + (a \cdot \overline{b})}$$

6.2. **(3 Punkte)** Formen Sie beide Seiten der Gleichung so weit wie möglich um.

$$(a \cdot b \cdot c) + \overline{\overline{(\overline{a} \cdot b)} \cdot \overline{(\overline{a} \cdot \overline{b})}} + a \quad = \quad \overline{\overline{c} + \overline{(a \cdot b \cdot c)} + (c + c \cdot a)}$$

7. Aufgabe (10 Punkte): Zahlendarstellungen

Kommentieren Sie Ihren Lösungsweg – Lösungen, bei denen der Lösungsweg nicht erkennbar ist, können nicht gewertet werden!

7.1. **(2 Punkte)** Berechnen Sie die Lösung von $243_{(5)} + 122_{(3)}$ und geben Sie das Ergebnis im Binärsystem (Basis 2) an.

7.2. **(4 Punkte)** Geben Sie die beiden Dezimalzahlen $a = -77,9375$ und $b = 9,0625$ als binäre Festkommazahlen mit 8 Vor- und 4 Nachkommastellen im Zweierkomplement an.

Berechnen Sie dann *binär*

- $a + b$ und
- $a \cdot b$

und geben Sie das Ergebnis mit 8 Vor- und 4 Nachkommastellen an. Konvertieren Sie das Ergebnis jeweils in das Dezimalsystem. Rechnen Sie ausschließlich mit vorzeichenbehafteten Zahlen! Interpretieren Sie ihr Ergebnis.

7.3. **(4 Punkte)** Gegeben ist die mit 16 Bit gespeicherte Gleitkommazahl 1010 0110 1101 1000, die nach dem in der LV vorgestellten, an dem Gleitkommazahlenformat der IEEE orientierten Schema codiert ist. Berechnen Sie den dargestellten Bruch.

Hinweis: Das Format sieht 1 Vorzeichenbit, 4 Bit für den mit dem Bias 7 versehenen Exponenten und 11 Bit für den Nachkommaanteil der Mantisse in dieser Folge vor.