Fakultät IV

Klausur Informatik 1
25. Juli 2002

Jähnichen / Flik
Jucknath / Neunast

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 geschrieben werden, können nicht gewertet werden!**
- Schreiben Sie Ihre Lösungen auf das Aufgabenblatt der jeweiligen Aufgabe. Verwenden Sie auch die Rückseiten. **Schreiben Sie keine Lösungen einer Aufgabe auf ein Blatt, das nicht zu dieser Aufgabe gehört!** Wenn Sie zusätzliche, von uns ausgegebene Blätter verwenden, geben Sie unbedingt an, zu welcher Aufgabe die Lösung gehört!
- Schreiben Sie deutlich! Doppelte, unleserliche oder mehrdeutige Lösungen können nicht gewertet werden!
- Schreiben Sie nur in **blau** oder **schwarz**. Lösungen, die mit Bleistift geschrieben sind, können nicht gewertet werden!
- 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`, `Real`, `Seq` oder den Strukturen für die Listenfunktionale) 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	9	
2	10	
3	14	
4	2	
5	5	
6	6	
7	3	

1. Aufgabe (9 Punkte): Abstrakte Datentypen

```
DATA person == person (name    : denotation,
                       alter   : nat,
                       student : bool)
```

1.1. Induzierte Signatur (2 Punkte) Geben Sie die induzierte Signatur des Datentyps `person` an. Um welche Art von Datentyp handelt es sich bei `person`? Wie nennt man die einzelnen induzierten Elemente?

1.2. Klausurliste (3 Punkte) Abhängig von ihrer Fachrichtung gelten für die einzelnen Studenten unterschiedliche Voraussetzungen bei der Klausurteilnahme. So benötigen Informatiker einen Übungsschein, technische Informatiker (TI) dagegen nicht. Andere Studiengänge haben eine spezielle Regelung ihres Prüfungsamtes. Der folgende Datentyp `student` beschreibt einen Teilnehmer der Veranstaltung Info1, wobei dieser entweder Informatiker, bzw. TI-ler ist, oder einer anderen Studienrichtung angehört.

```
DATA student == informatiker(name: denotation, matrNr: nat, tests: seq[nat], schein: bool)
                TI(name: denotation, matrNr: nat)
                andere(name: denotation, matrNr: nat, schein: bool, klausurberechtigung: bool))
```

Deklarieren und definieren Sie die Funktion `anmelden`, die eine Sequenz von `studenten` übergeben bekommt und eine Sequenz von `studenten` zurückgibt, die nur diejenigen Studenten enthält, die an der Klausur teilnehmen dürfen. Dieses sind alle TI-ler, alle Informatiker mit einem Übungsschein und alle anderen Studenten mit einer Klausurberechtigung.

1.3. Scheinvergabe (4 Punkte) Deklarieren und definieren Sie die Funktion `scheinvergabe`, die eine Sequenz von Studenten übergeben bekommt und eine Sequenz von Studenten zurückgibt. Dabei soll `scheinvergabe` für alle Studenten, die mindestens zwei Tests mit einer Punktzahl größer als 5 bestanden haben, die Scheinbedingung setzen.

2. Aufgabe (10 Punkte): Such- und Sortieralgorithmen

Gegeben sei der Datentyp `student` aus der Aufgabe 1.2.

2.1. Sortieralgorithmen (5 Punkte) Deklarieren und definieren Sie die Funktion `sortieren`, die eine Sequenz von Studenten bekommt und eine nach der Matrikelnummer sortierte Sequenz von Studenten zurückgibt. Die Wahl des Sortieralgorithmus ist Ihnen dabei freigestellt.

2.2. Suchalgorithmen (5 Punkte) In einer eingeschränkten Sichtweise besteht ein Studiengang aus einer Sequenz von Studenten und eine Fakultät aus einer Sequenz von Studiengängen.
Deklarieren und definieren Sie die Funktion `mehrfachBelegt`, die eine Fakultät übergeben bekommt und eine Sequenz von Studenten zurückgibt, welche alle Studenten enthält, die in mehreren Studiengängen dieser Fakultät eingeschrieben sind. Diese Sequenz sollte keine Studenten doppelt enthalten.

Hinweis: Jeder Student ist über die Matrikelnummer eindeutig zu identifizieren. Kein Student kann sich mehrfach bei dem gleichen Studiengang immatrikulieren.

3. Aufgabe (14 Punkte): Funktionen höherer Ordnung

3.1. Ersatzfunktion (8 Punkte) Deklarieren und definieren Sie die Funktion `repeat`, die eine Funktion f und eine natürliche Zahl n übergeben bekommt und f genau n mal auf einen übergebenen Wert x anwendet.

3.2. Ableitung eines Polynoms (6 Punkte) Deklarieren und definieren Sie die Funktion `max`, die eine Funktion f und eine Funktion g übergeben bekommt, und eine Funktion h zurückgibt, welche für jeden Funktionswert n den jeweils größeren Funktionswert (also g(n) oder f(n)) berechnet.

4. Aufgabe (2 Punkte): Codesicherung

Überprüfen Sie, ob folgender Datenstrom, der mit 4 CRC-Prüfbits gesichert ist, fehlerfrei empfangen wurde. Das Generatorpolynom ist $x^4 + x^1 + x^0$.

0 0 1 1 0 1 0 1 1 0 1 1 1 1 1 0

5. Aufgabe (5 Punkte): Zahlendarstellungen

Gegeben sind zwei Zahlen in gemischt gebrochener 2-Komplement-Darstellung mit je 4 Vor- und 4 Nachkommastellen, $a = 1110.0101_2$ und $b = 0010.1000_2$

5.1. Umwandlung (1 Punkte) Wandeln Sie die Zahlen in das Dezimalsystem. Der Rechenweg muß erkennbar sein.

5.2. Subtraktion (1 Punkte) Bilden Sie die Subtraktion $a - b$ im oben angegebenen Format. Die Überträge müssen erkennbar sein.

5.3. Umwandlung (1 Punkte) Überführen Sie die Zahlen in normalisierte Gleitpunktzahlen unter Verwendung der verkürzten Darstellung (1-Bit-Vorzeichen, 4-Bit-Exponent, 11-Bit-Mantisse, bias=7).

5.4. Multiplikation (2 Punkte) Multiplizieren Sie die beiden Zahlen in Gleitpunktdarstellung und geben Sie das Produkt ebenfalls in normalisierter verkürzter Darstellung an. Betrachten Sie getrennt das Vorzeichen, die Mantissenbeträge und die Exponenten. Die Rechenschritte müssen erkennbar sein.

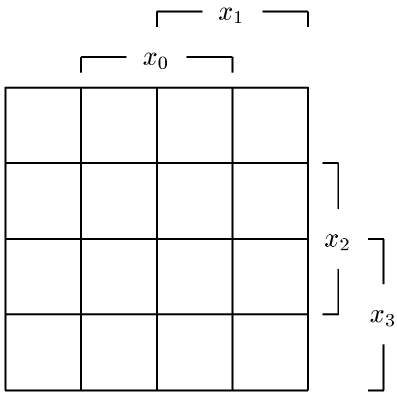
6. Aufgabe (6 Punkte): Schaltnetzentwurf

Gegeben ist folgende Tabelle.

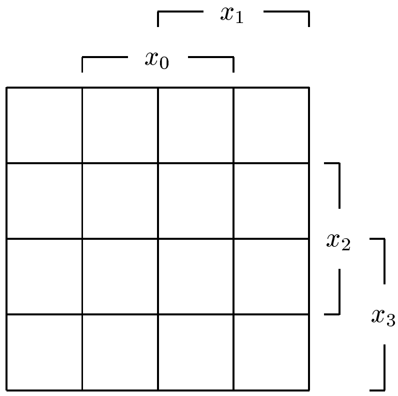
x_3	x_2	x_1	x_0	y_0	y_1
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	1	1
0	1	0	1	1	*
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	1	*	1
1	1	1	1	1	1

6.1. Minimierung (3 Punkte) Bestimmen Sie mit Hilfe der KV-Tafeln eine minimale *disjunktive* Normalform für y_0 und eine minimale *konjunktive* Normalform für y_1. Markieren Sie die zusammengefaßten Bereiche.

y_0:



y_1:



6.2. Umformung (2 Punkte) Formen Sie die Gleichungen so um, daß für die Realisierung von y_0 nur NAND- und für die Realisierung von y_1 nur NOR-Gatter erforderlich sind.

6.3. Schaltung (1 Punkte) Zeichnen Sie eine gemeinsame Schaltung zur Realisierung beider Funktionen mit NAND-bzw. NOR-Gattern unter Verwendung der in der vorangehenden Teilaufgabe ermittelten Funktionsgleichungen.

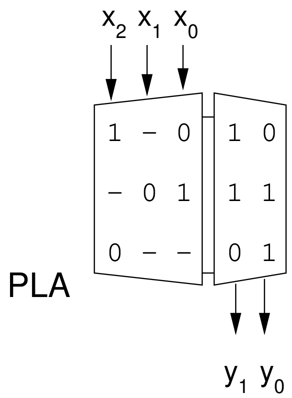
Name: ....

Matr.-Nr. ....

A7

7. Aufgabe (3 Punkte): Schaltnetzanalyse

Gegeben ist folgendes Schaltnetz.



7.1. Normalformen (2 Punkte) Geben Sie die Funktion y_1 in *ausgezeichneter disjunktiver* Normalform und die Funktion y_0 in *ausgezeichneter konjunktiver* Normalform an.

7.2. ROM (1 Punkte) Realisieren Sie die Funktionen y_1 und y_0 mit einem ROM-Baustein unter Verwendung der Vorgabe.