Name: | Vorname: | Matr.–Nr.:

Technische Universität München
Fakultät für Informatik
Prof. Dr. M. Broy

WS 2001/2002
9. Februar 2002

Klausur zu Einführung in die Informatik I

(Gruppe B)

Hinweis: Das Aufgabenblatt muss abgegeben werden !

Aufgabe 1 Boolesche Algebra (circa 6 Punkte)

Seien r, s, t freie Identifikatoren.

a) Zeigen Sie unter ausschließlicher Benutzung der Involutionsgesetze, Kommutativgesetze, Assoziativgesetze, Distributivgesetze, Gesetze von de Morgan, Neutralitätsgesetze bzw. der Definition der Implikation, dass folgende Terme jeweils äquivalent sind und geben Sie bei jedem Rechenschritt an, welches Gesetz Sie angewendet haben!

(i) Term 1: $\neg r \vee (s \wedge r)$ Term 2: $r \Rightarrow s$
(ii) Term 1: $\neg(\neg r \vee s) \wedge (s \vee t)$ Term 2: $r \wedge (\neg s \wedge t)$

b) Zeigen oder widerlegen Sie:

Der Term $(r \vee \neg s) \wedge \neg(s \Rightarrow t)$ ist erfüllbar.

Aufgabe 2 Textersetzungssystem (circa 10 Punkte)

Gesucht ist ein Textersetzungssystem, das natürliche Zahlen $n \geq 0$ in Strichzahldarstellung in Binärdarstellung, d. h. Wörter der Form <$b_k \ldots b_1$> mit $b_i \in \{O, L\}$, $1 \leq i \leq k$, konvertiert. Beispiel: Bei Eingabe <||||||> terminiert die Berechnung mit der Ausgabe <*LLO*>.

a) Geben Sie einen geeigneten Zeichenvorrat für das in Teilaufgabe b) zu entwickelnde Textersetzungssystem an.

b) Ergänzen Sie auf dem Angabenblatt die rechten Seiten der Regelmenge des Textersetzungssystems derart, dass das Textersetzungssystem diese Konvertierung durchführt.

Die Zeichen „<“ und „>“ markieren Anfang und Ende des Eingabeworts (bzw. des Ausgabeworts nach der Berechnung). Mit dem „Schiffchen“ r wird die Strichzahl ganzzahlig durch 2 dividiert und der Rest als L bzw. O vermerkt. Das „Schiffchen“ s dient der Trennung von Strichzahl und Binärzahl.

$T = \{$

$<> \rightarrow$ ☐ , (1)

$<| \rightarrow$ ☐ , (2)

$r|| \rightarrow$ ☐ , (3)

$r|> \rightarrow$ ☐ , (4)

$r> \rightarrow$ ☐ , (5)

$r|s \rightarrow$ ☐ , (6)

$rs \rightarrow$ ☐ , (7)

$<s \rightarrow$ ☐ , (8)

$\}$

c) Testen Sie (unabhängig von der Korrektheit Ihrer Lösung zu b) Ihr Textersetzungssystem mit dem Eingabewort <|||> . Geben Sie für jeden Berechnungsschritt die Nummer der angewendeten Regel, und das daraus resultierende Wort an.

d) Beantworten Sie folgende Fragen und geben Sie jeweils eine kurze Begründung an.

(i) Gibt es eine Strichzahl n, für die sich die möglichen Berechnungen dieses Textersetzungssystems gegenüber der Berechnung des Textersetzungssystems gemäß der Markovstrategie unterscheiden?

(ii) Ergeben sich daraus Auswirkungen auf das Berechnungsergebnis?

Aufgabe 3 BNF (circa 9 Punkte)

Die Teilmenge OT der booleschen Terme mit vereinfachter Klammerung, die nur über der Variable x und dem Operator $\vee$ aufgebaut sind ("Oder-Terme") wird durch folgende BNF beschrieben:

$$< ot >::= x|(< ot > \{\vee < ot >\}^+)$$

Die maximale Zahl von ineinander geschachtelten Klammern eines Oder-Terms nennt man seine Schachtelungstiefe. OT_i sei die Menge aller Oder-Terme mit Schachtelungstiefe $i \in IN$, z.B. $(x \vee x) \in OT_1$, $(x \vee (x \vee x)) \in OT_2$.

a) Geben Sie aus OT_1 und OT_2 jeweils zwei andere Beispiele an.

b) Geben Sie die Mengen OT_0, OT_1 und OT_2 jeweils in Mengenschreibweise an.

c) Beschreiben Sie die Mengen OT_0 und OT_1 jeweils durch einen regulären Ausdruck.

d) Modifizieren Sie die gegebene BNF so, daß als Variablenbezeichner alle endlichen Zeichenreihen über dem Alphabet $\{a, b, c, \ldots, z, 0, 1, \ldots, 9\}$ zulässig sind, die mit einem Buchstaben beginnen.

Aufgabe 4 Programmierung (circa 8 Punkte)

Hinweis: Die in dieser Aufgabe zu formulierenden Rechenvorschriften können in Gofer oder in Vorlesungsnotation angegeben werden!

Die Rechenstruktur **SEQ** sei wie in der Vorlesung definiert.

a) Es bezeichne `rs` eine Sequenz von Sequenzen der Sorte **m**, d.h. `rs` ist von der Sorte **seq seq m**. Die Funktion `addFirst` fügt ein Element `y` der Sorte **m** am Anfang jedes Elements von `rs` an. Formulieren Sie unter Verwendung applikativer Programmiertechniken eine Rechenvorschrift, die die Funktion `addFirst` implementiert!

Gegeben sei eine Sequenz `ys` von paarweise verschiedenen Elementen der Sorte **m**. Gesucht ist eine Sequenz der Sorte **seq seq m**, die alle Untersequenzen von `ys` enthält (**Bemerkung:** Eine Untersequenz entsteht aus einer Sequenz ys durch Streichen beliebig vieler Elemente).

b) Formulieren Sie kurz die Grundidee Ihres Algorithmus!

c) Geben Sie unter Verwendung applikativer Programmiertechniken eine Rechenvorschrift an, die die Sequenz aller Untersequenzen von `ys` berechnet! (Hinweis: Der Einsatz der Funktion `addFirst` aus Teilaufgabe a) ist bei der Lösung dieser Aufgabe hilfreich!)

Aufgabe 5 Induktion (circa 7 Punkte)

Die Rechenstruktur **SEQ** sei wie in der Vorlesung definiert. Gegeben sei eine Rechenvorschrift zur Revertierung einer Sequenz `rs`:

fct rev = (**seq char** rs) **seq char**:
  if rs $\stackrel{?}{=}$ empty **then** empty
    else rev (rest (rs)) ∘ <first (rs)>
  fi

Außerdem sei das Prädikat

$$\text{rev}\ (rs \circ ts) = \text{rev}\ (ts) \circ \text{rev}\ (rs) \qquad (*)$$

gegeben, wobei `rs` und `ts` von der Sorte **seq char** sind.

a) Zeigen Sie, dass das Prädikat $(*)$ für alle Sequenzen `ts` der Sorte **seq char** und für rs_0 = `empty` bzw. rs_1 = <t>, wobei `t` ein Element der Sorte **char** ist, den Wert `true` annimmt!

b) Zeigen Sie durch strukturelle Induktion, dass das Prädikat $(*)$ für alle Sequenzen `rs` und `ts` gilt; als Induktionsanfang können Sie dabei die Resultate aus Teilaufgabe a) verwenden!