Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: ______ Kennwort: ______ Arbeitsplatz-Nr.: ______	**Frühjahr 1998**	**66112**

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: Informatik (vertieft studiert)

Einzelprüfung: Automatentheorie, Komplexität, Algorith.

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 6

Bitte wenden!

Thema Nr. 1

Sämtliche Teilaufgaben sind zu bearbeiten!

1. Man beweise oder widerlege für Sprachen L_1, L_2 über einem Alphabet Σ:

   a) $L_1 (L_2 L_1)^* = (L_1 L_2)^* L_1$
   b) $(L_1 \cap L_2)^* = L_1^* \cap L_2^*$
   c) $(L_1^*)^* = L_1^*$

2. Skizzieren Sie Aufbau und Eigenschaften eines Parsers.

3. Für welche Sprachenklassen der Chomsky-Hierarchie ist das Leerheitsproblem entscheidbar? (Begründungen!)

4. Was versteht man unter 'lazy evaluation'?

5. Zeigen Sie, daß $\langle x, y \mapsto 2^x (2y + 1)-1\rangle$ eine primitiv-rekursive Paarfunktion ist.

6. Geben Sie einen Datentyp für Warteschlangen mit Prioritäten an, und beschreiben Sie diesen umgangssprachlich.

7. Welche der folgenden Eigenschaften sind für beliebige WHILE-Programme P entscheidbar? (Begründung!)

   a) P terminiert bei Eingabe 0 (Null).
   b) P berechnet eine primitiv rekursive Funktion.
   c) P enthält höchstens 3 ineinander geschachtelte WHILE-Schleifen.
   d) Zu P gibt es ein äquivalentes Programm P' mit nur einer WHILE-Schleife.

Fortsetzung nächste Seite!

8. Aktualisieren einer Flug-Informationstafel (nach H. Partsch). (Abbildung Beispiel)

flight	destination	scheduled	departure	status
BA 947	London	7.55	10.55	delayed
AZ 435	Mailand	9.10	10.50	delayed
SR 551	Zürich	9.30	10.50	delayed
LH 131	Bombay	10.00	10.45	delayed
RQ 501	Bern	10.00	10.30	delayed
NS 425	Erfurt	10.20	10.45	delayed
LH 2531	Berlin	10.30	10.30	
AY 826	Bangkok	10.30	10.30	
LH 115	Frankfurt	10.30	10.30	
LH 4271	Bari	10.40	10.40	
LH 947	Bremen	10.45	10.45	

Gegeben ist eine Liste von Einträgen. Ein Eintrag kann eine Leerzeile oder eine Flug-Info-Zeile sein. Die Flug-Info-Zeilen sind nach Abflugzeiten geordnet.
Die Aktualisierung soll alle Flug-Info-Zeilen (wiederum geordnet) an den Anfang der Liste und alle Leerzeilen ans Ende bringen.

Konstruieren Sie ein Programm, das die Aktualisierungsaufgabe löst.

Geben Sie insbesondere eine formale Spezifikation, geeignete Datenstrukturen, sowie zur Verifikation geeignete Invarianten an.

Thema Nr. 2

Sämtliche Teilaufgaben sind zu bearbeiten!

Aufgabe 1:

Gegeben sei das Alphabet $\Sigma = \{0,1\}$, die Menge

$$L = \{\, 0^n 1^m \mid n \ge 0,\ m \ge 0, \text{ und } n+m \text{ ist ungerade} \,\}$$

von Zeichenreihen über Σ sowie die Grammatik Γ mit Σ als Menge der Terminalzeichen, den Nichtterminalzeichen Z und A, dem Axiom Z und den Produktionsregeln

$$\begin{aligned}
&Z \to 0\\
&Z \to 1\\
&Z \to 0A\\
&Z \to 1A\\
&Z \to 00Z\\
&A \to 11\\
&A \to 11A\,.
\end{aligned}$$

a) Ist Γ mehrdeutig? Beweisen Sie Ihre Behauptung.

b) Beweisen Sie: Für den Sprachschatz $\mathcal{L}(\Gamma)$ von Γ gilt: $\mathcal{L}(\Gamma) = L$.

c) Konstruieren Sie direkt aus Γ einen nicht-deterministischen endlichen Automaten, der genau die Zeichenreihen von L akzeptiert.

d) Geben Sie eine (deterministische) Turingmaschine M an, die außer einem Leerzeichen nur die Zeichen aus Σ verwendet und L in folgendem Sinne akzeptiert: Angesetzt auf das erste Zeichen eines auf dem ansonsten leeren Band stehenden Wortes $w \in \Sigma^*$ erreicht M genau dann nach endlich vielen Schritten einen Endzustand, wenn $w \in L$ ist.

Aufgabe 2:

Mit **sequ** integer sei im folgenden die Menge aller endlichen Folgen ganzer Zahlen bezeichnet. Für **sequ** integer seien als Grundoperationen verfügbar:

empty: $\to$ **sequ** integer,	*empty* = leere Folge
isempty: **sequ** integer $\to$ boolean	*isempty*(s) = *true* $\Leftrightarrow$ s ist leer
first: **sequ** integer $\to$ integer,	*first*: $(s_1,\dots,s_n) \mapsto s_1$
rest: **sequ** integer $\to$ **sequ** integer,	*rest*: $(s_1,s_2,\dots,s_n) \mapsto (s_2,\dots,s_n)$
prefix: integer $\times$ **sequ** integer $\to$ **sequ** integer,	*prefix*: $(x,(s_1,\dots,s_n)) \mapsto (x,s_1,\dots,s_n)$
postfix: **sequ** integer $\times$ integer $\to$ **sequ** integer,	*postfix*: $((s_1,\dots,s_n),x) \mapsto (s_1,\dots,s_n,x)$
$\circ$: **sequ** integer $\times$ **sequ** integer $\to$ **sequ** integer,	$\circ$: $((s_1,\dots,s_n),(t_1,\dots,t_m)) \mapsto (s_1,\dots,s_n,t_1,\dots,t_m)$

(Für s = *empty* sind *first*(s) und *rest*(s) nicht definiert. Die "Konkatenation" $\circ$ wird im folgenden in Infixschreibweise verwendet. $\circ$ ist assoziativ, d.h. es gilt $u \circ (v \circ w) = (u \circ v) \circ w$ für alle $u,v,w \in$ **sequ** integer.)

Fortsetzung nächste Seite!

a) Beweisen Sie, daß für beliebige $x \in$ integer und $s,t \in$ **sequ** integer gilt:

a1) $prefix(x, s \circ t) = prefix(x,s) \circ t$.

a2) $postfix(s,x) \circ t = s \circ prefix(x,t)$.

Sei nun weiter $a > 0$ eine fest vorgegebene ganze Zahl. Für $i = 1,2,3$ seien die Funktionen $teil_i$: **sequ** integer → **sequ** integer definiert durch:

$$teil_i(s) = \begin{cases} empty, & \text{falls } s = empty, \\ prefix(first(s), teil_i(rest(s)), & \text{falls } s \neq empty \text{ und } p_i(first(s)) = true, \\ teil_i(rest(s)) & \text{sonst,} \end{cases}$$

wobei gilt: $p_1(x) = true \Leftrightarrow x < -a$, $p_2(x) = true \Leftrightarrow -a \leq x \leq a$, $p_3(x) = true \Leftrightarrow x > a$.

Die Funktion *ordnen*: **sequ** integer → **sequ** integer sei definiert durch:

$$ordnen(s) = teil_1(s) \circ teil_2(s) \circ teil_3(s).$$

Außerdem sei folgende Funktionsvereinbarung gegeben:

```
function ORDALL(s,v,w: sequ integer) sequ integer:
  if isempty(s) then v∘w
  else if first(s) < -a then prefix(first(s),ORDALL(rest(s),v,w))
       □ -a ≤ first(s) ∧ first(s) ≤ a then ORDALL(rest(s),postfix(v,first(s)),w)
       else ORDALL(rest(s),v,postfix(w,first(s)))
       endif
  endif
```

b) Bestimmen Sie die Werte von $teil_1(s)$, $teil_2(s)$, $teil_3(s)$, $ordnen(s)$ und $ORDALL(s,v,w)$ für $a = 5$, $s = (2,7,-6,9,-3,-8,7)$, $v = (0,0)$, $w = (1,1)$.

c) Beweisen Sie, daß für alle $s \in$ **sequ** integer gilt:

$$ordnen(s) = ORDALL(s, emty, empty).$$

d) Aus *ORDALL* soll in systematischer Weise ein iterativer Algorithmus zur Berechnung von *ordnen(s)* für alle $s \in$ **sequ** integer entwickelt werden, der nur die angegebenen Grundoperationen verwendet. Betten Sie dazu *ORDALL* in einen geeigneten allgemeineren repetitiv rekursiven Algorithmus ein, und entrekursivieren und spezialisieren Sie diesen zu dem gesuchten iterativen Algorithmus.

Hinweis: Verwenden Sie zur Formulierung der Algorithmen eine programmiersprachenähnliche Notation, wie sie etwa zur Vereinbarung von *ORDALL* verwendet ist.

Fortsetzung nächste Seite!

Aufgabe 3:

Durch die Funktionsvereinbarungen

> **function** $F(n\text{:nat})$boolean:
> **if** $n = 0$ **then** *true*
> ▯ $n = 1 \vee n = 2 \vee n = 3$ **then** *false*
> **else** $G(n\text{-}1) \wedge G(n\text{-}2) \wedge G(n\text{-}3)$ **endif**,
> **function** $G(n\text{:nat})$boolean:
> **if** $n = 0$ **then** *false*
> ▯ $n = 1 \vee n = 2 \vee n = 3$ **then** *true*
> **else** $F(n\text{-}1) \vee F(n\text{-}2) \vee F(n\text{-}3)$ **endif**

sind zwei Funktionen $F, G: \mathbb{N}_0 \to \{true, false\}$ definiert.

Beweisen Sie, daß für beliebiges $n \in \mathbb{N}_0$ gilt:

a) $F(n)$ terminiert.

b) $F(n) = true \Leftrightarrow n$ ist durch 4 teilbar.

Aufgabe 4:

Gegeben sei eine unendliche Teilmenge M von $\mathbb{N}_0$ und eine μ-rekursive ("berechenbare") Funktion $g: \mathbb{N}_0 \to \mathbb{N}_0$, so daß für alle $n \in \mathbb{N}_0$ gilt:

$$n \in M \Leftrightarrow g(n) = 0.$$

Durch die noch unvollständige Definition

$$f(n) = \begin{cases} \mu_x(g(x) = 0), & \text{falls } n = 0, \\ \mu_x(\ldots?\ldots), & \text{falls } n > 0 \end{cases}$$

sei eine weitere Funktion $f: \mathbb{N}_0 \to \mathbb{N}_0$ gegeben.

Bestimmen Sie den unvollständigen Teil ...?... der Definition von f so, daß f folgende drei Eigenschaften besitzt:

i) f ist μ-rekursiv.
ii) Für alle $n \in \mathbb{N}_0$ gilt: $f(n) < f(n+1)$.
iii) Für alle $n \in \mathbb{N}_0$ gilt: $n \in M \Leftrightarrow \exists m(f(m) = n)$.

Begründen Sie Ihre Antwort!

Hinweis: Der Operator μ_x ist für $(k+1)$-stellige $(k \geq 0)$ Prädikate $p(x_1,\ldots,x_k,x)$ definiert durch:

$\mu_x(p(x_1,\ldots,x_k,x)) = y$, falls $p(x_1,\ldots,x_k,y)$ und nicht $p(x_1,\ldots,x_k,z)$ für $0 \leq z < y$.

Gilt $p(x_1,\ldots,x_k,z)$ für kein $z \in \mathbb{N}_0$, so ist $\mu_x(p(x_1,\ldots,x_k,x))$ undefiniert.