| Prüfungsteilnehmer | Prüfungstermin | Einzelprüfungsnummer |
| --- | --- | --- |
| Kennzahl: ______ | Herbst 1994 | 66110 |
| Kennwort: ______ | | |
| Arbeitsplatz-Nr.: ______ | | |

**Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen**

**- Prüfungsaufgaben -**

**Fach:** **Informatik (vertieft studiert)**

**Einzelprüfung:** **Automatentheorie, Algorithm. Sprachen**

**Anzahl der gestellten Themen (Aufgaben): 1**

**Anzahl der Druckseiten dieser Vorlage: 4**

Bitte wenden!

Sämtliche Aufgaben sind zu bearbeiten!

**Aufgabe 1**

$M$ sei die Menge aller Zeichenreihen über dem Alphabet $\{0,1\}$, die mindestens so viele Zeichen 1 wie Zeichen 0 enthalten.

Geben Sie eine (deterministische) Turingmaschine $T$ an, die außer einem Leerzeichen nur die Zeichen aus $\{0,1\}$ verwendet und $M$ in folgendem Sinne akzeptiert: Ein Wort $w \in \{0,1\}^*$ steht auf dem ansonsten leeren Band! Angesetzt auf das erste Zeichen von $w$ (bzw. auf ein Leerzeichen, falls $w$ das leere Wort ist), erreicht $T$ genau dann nach endlich vielen Schritten einen Endzustand, wenn $w \in M$ ist.

**Aufgabe 2**

Gegeben sei die Grammatik $\Gamma_1$ mit $\{a,b\}$ als Menge der Terminalzeichen, den Nichtterminalzeichen Z,A,B, dem Axiom Z und den Produktionsregeln

$$
\begin{aligned}
Z &\to aA \\
A &\to b \\
A &\to bB \\
B &\to a \\
B &\to b \\
B &\to bB
\end{aligned}
$$

Die Grammatik $\Gamma_2$ entstehe aus $\Gamma_1$ dadurch, daß zu diesen Produktionsregeln noch die weitere Regel

$$B \to BA$$

hinzugenommen wird.

Der jeweilige Sprachschatz von $\Gamma_1$ und $\Gamma_2$ sei mit $\mathcal{L}(\Gamma_1)$ bzw. $\mathcal{L}(\Gamma_2)$ bezeichnet.

a) Beweisen Sie: $\mathcal{L}(\Gamma_1) = \{ab^n \mid n \in \mathbb{N}\} \cup \{ab^na \mid n \in \mathbb{N}\}$.

b) Geben Sie einen deterministischen endlichen Automaten an, der genau die Zeichenreihen von $\mathcal{L}(\Gamma_1)$ akzeptiert.

c) Beweisen Sie: $\Gamma_2$ ist mehrdeutig.

d) Beweisen Sie: $\mathcal{L}(\Gamma_1) \neq \mathcal{L}(\Gamma_2)$.

e) Überführen Sie $\Gamma_2$ in Greibach-Normalform.

**Aufgabe 3**

Sei **int** die Menge der ganzen Zahlen und **sequ int** die Menge aller endlichen Folgen ganzer Zahlen. Für **sequ int** seien als Grundoperationen verfügbar:

Fortsetzung nächste Seite!

| | |
|---|---|
| *empty*: → **sequ** int, | *empty* = leere Folge |
| *isempty*: **sequ** int → boolean, | *isempty*($s$) = *true* ⇔ $s$ ist leer |
| *first*: **sequ** int → int, | *first*: $(s_1,\ldots,s_n) \mapsto s_1$ |
| *rest*: **sequ** int → **sequ** int, | *rest*: $(s_1,s_2\ldots,s_n) \mapsto (s_2,\ldots,s_n)$ |
| *prefix*: int × **sequ** int → **sequ** int, | *prefix*: $(x,(s_1,\ldots,s_n)) \mapsto (x,s_1,\ldots,s_n)$ |

(Für $s$ = *empty* sind *first*($s$) und *rest*($s$) nicht definiert.)

$S_3$ sei die Menge aller Folgen aus **sequ** int mit einer durch 3 teilbaren Anzahl von Komponenten.

a) Geben Sie (unter ausschließlicher Verwendung der genannten Grundoperationen) rekursive Funktionsvereinbarungen an für Funktionen *last*, *lead*, *postfix*, *conc* mit folgender Bedeutung (*last*($s$) und *lead*($s$) sind nur für $s \neq$ *empty* definiert):

| | |
|---|---|
| *last*: **sequ** int → int, | *last*: $(s_1,\ldots,s_n) \mapsto s_n$ |
| *lead*: **sequ** int → **sequ** int, | *lead*: $(s_1,\ldots s_{n-1},s_n) \mapsto (s_1,\ldots,s_{n-1})$ |
| *postfix*: **sequ** int × int → **sequ** int, | *postfix*: $((s_1,\ldots,s_n),x) \mapsto (s_1,\ldots,s_n,x)$ |
| *conc*: **sequ** int × **sequ** int → **sequ** int, | *conc*: $((s_1,\ldots,s_n),(t_1,\ldots,t_m)) \mapsto (s_1,\ldots,s_n,t_1,\ldots,t_m)$ |

b) Gegeben sei die Funktion *vorn* durch die Funktionsvereinbarung

```
function vorn(s:sequ int) sequ int:
  (* definiert nur für s ∈ S3 *)
  if isempty(s) then s
  else prefix(first(s), vorn(rest(lead(lead(s))))) endif
```

Beweisen Sie: Für $s \in S_3$ ist *vorn*($s$) das "vordere Drittel von $s$", d.h. für $s = (s_1,\ldots,s_{3k})$, $k \in \mathbb{N}_0$, ist *vorn*$(s) = (s_1,\ldots,s_k)$.

c) Geben Sie analog zur Funktion *vorn* aus Teilaufgabe b) eine rekursive Funktionsvereinbarung für eine Funktion *hinten* an, die nur die genannten Grundoperationen und Funktionen aus Teilaufgabe a) verwendet und die für $s \in S_3$ das hintere Drittel von $s$ berechnet (d.h. *hinten*$(s) = (s_{2k+1},\ldots,s_{3k})$ für $s = (s_1,\ldots,s_{3k})$, $k \in \mathbb{N}_0$).

d) Geben Sie eine rekursive Funktionsvereinbarung für eine Funktion *mitte* an, die außer den genannten Grundoperationen und den Funktionen aus Teilaufgabe a) ausschließlich die Funktion *vorn* aus Teilaufgabe b) verwenden darf und die für $s \in S_3$ das mittlere Drittel von $s$ berechnet (d.h. *mitte*$(s) = (s_{k+1},\ldots,s_{2k})$ für $s = (s_1,\ldots,s_{3k})$, $k \in \mathbb{N}_0$).

e) Die rekursive Funktionsvereinbarung von *vorn* in Teilaufgabe b) soll in systematischer Weise in einen iterativen Algorithmus (mit gleicher Wirkung) überführt werden (der ebenfalls nur die angegebenen Grundoperationen und Funktionen aus Teilaufgabe a) verwendet). Betten Sie dazu *vorn* in einen geeigneten allgemeineren repetitiv rekursiven Algorithmus ein, und entrekursivieren und spezialisieren Sie diesen zu dem gesuchten iterativen Algorithmus.

f) Objekte aus **sequ** int können in Programmiersprachen wie PASCAL, MODULA o.ä. als lineare Listen realisiert werden. Geben Sie (in einer derartigen Programmiersprache) entsprechende Typvereinbarungen und Algorithmen zur Realisierung der angegebenen Grundoperationen an.

Fortsetzung nächste Seite!

g) Geben Sie (in PASCAL, MODULA o.ä.) einen iterativen Algorithmus *laenge* an, der unter Verwendung der Realisierungen der Grundoperationen gemäß Teilaufgabe f) die Anzahl der Komponenten einer als lineare Liste realisierten Folge aus **sequ** int berechnet.

**Aufgabe 4**

Durch die Funktionsvereinbarung

> **function** $f(x,y,z\text{:nat})\text{nat}$:
> **if** $x = y+1$ **then** $z+y$ **else** $f(x+y+1, 2*(y+1), z+y+1)$ **endif**

ist eine Funktion $f: \mathbb{N}_0^3 \rightarrow \mathbb{N}_0$ definiert.

a) Bestimmen Sie $f(6,0,1)$.

b) Beweisen Sie: $f(x,y,z)$ terminiert für alle $x,y,z \in \mathbb{N}_0$ mit $x > y$.

c) Beweisen Sie, daß für alle $x,y,z \in \mathbb{N}_0$ mit $x > y$ gilt: $f(x,y,z)$ ist genau dann eine gerade Zahl, wenn $x + z$ ungerade ist.