Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: ______ Kennwort: ______ Arbeitsplatz-Nr.: ______	**Herbst 2000**	**46113**

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

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

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 5

Bitte wenden!

Thema Nr. 1

Aufgabe 1

Zur Darstellung einer Gleitkommazahl arbeiten wir mit dem Alphabet S = {´+´, ´−´, ´.´, ´E´, ´0´,´1´,´2´,´3´,´4´,´5´,´6´,´7´,´8´,´9´}. Daraus bilden wir Zeichenketten A, .. ,G $\in S^*$, die wir schließlich mit Hilfe der Verkettungsoperation ´o´ zu Gleitkommazahlen Z kombinieren: Z = A o B o C o D o E o F o G. Für die Zeichenfolgen A, .. , G gelten dabei die folgenden Bedingungen:

Zeichenreihe	Länge l	darf folgende Zeichen enthalten	Einschränkungen
A	$0 \leq l_A \leq 1$	´+´, ´−´	
B	$l_B \geq 1$	´0´,´1´,´2´,´3´,´4´,´5´,´6´,´7´,´8´,´9´	
C	$0 \leq l_C \leq 1$	´.´	$l_C = 1 \Leftrightarrow l_D \geq 1$
D	$l_D \geq 0$	´0´,´1´,´2´,´3´,´4´,´5´,´6´,´7´,´8´,´9´	
E	$0 \leq l_E \leq 1$	´E´	$l_E = 1 \Leftrightarrow l_G \geq 1$
F	$0 \leq l_F \leq 1$	´+´, ´−´	$l_F = 1 \Rightarrow l_G \geq 1$
G	$l_G \geq 0$	´0´,´1´,´2´,´3´,´4´,´5´,´6´,´7´,´8´,´9´	

Zulässige Gleitkommazahlen wären damit etwa „+5“, „9.123“, „1298.222E+2“, „-2E2“. Nicht zugelassen sind dagegen z.B. „.12“, „E12“ oder „1E+“. Verwenden Sie in den folgenden Aufgaben als Abkürzung das Symbol t mit t $\in$ {´0´, .. , ´9´}!

a) $L_Z \subset S^*$ sei die Sprache, die man durch Bildung aller möglichen Gleitkommazahlen gemäß der in obiger Tabelle angegebenen Regeln erhält.
Zeigen Sie, dass L_Z vom Typ Chomsky-3 ist, indem Sie eine erzeugende rechtslineare Grammatik G dafür angeben!

b) Zeichnen Sie das Zustands-Übergangsdiagramm eines endlichen Automaten, der L_Z akzeptiert!

c) Geben Sie einen regulären Ausdruck für L_Z an!

Fortsetzung nächste Seite!

Aufgabe 2

Wir betrachten folgendes Programmfragment P (in Pascal-ähnlicher Notation):

```
i := 1; f := 1;
while i ≤ n do
        begin
        f := f*i;
        i := i+1;
        end;
```

a) Welche Funktion f wird durch dieses Programmfragment berechnet?

b) Bezüglich der Berechenbarkeit von Funktionen durch imperative Programme unterscheidet man zwischen FOR- (oder LOOP-) und WHILE-Berechenbarkeit.
Charakterisieren Sie kurz diese beiden Berechenbarkeitsbegriffe und ihre Beziehung zueinander!

c) Zu welcher der beiden in b) genannten Berechenbarkeitsklassen gehört die Funktion f aus Teilaufgabe a)?
Begründen Sie Ihre Antwort!

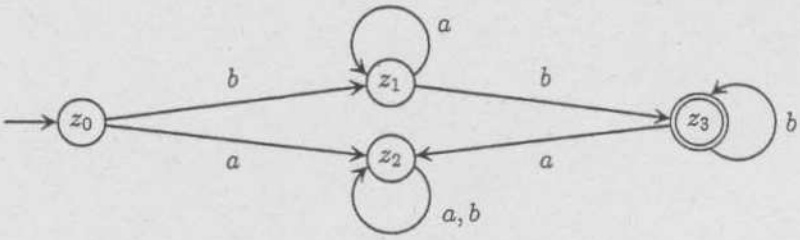
Aufgabe 3

a) Geben Sie eine Turing-Maschine TM zur Berechnung der Vorgängerfunktion *pred* einer natürlichen Zahl $n \geq 1$ an! Dabei soll die Zahl n in Binärform (höchstwertiges Bit links) ohne führende Nullen auf dem Band von TM dargestellt sein und der Schreib-Lesekopf anfangs auf dem höchstwertigen Bit stehen. Nach der Berechnung dürfen führende Nullen auftreten. Außerhalb der Darstellung von n steht an allen anderen Stellen des Bandes das Leerzeichen *´#´*.

b) Wenden Sie Ihre Turingmaschine zur Berechnung von *pred(24)* an!
Geben Sie dabei für alle Zwischenzustände jeweils die Bandbelegung und den Zustand von TM an!

Thema Nr. 2

Aufgabe 1



Welche der folgenden regulären Ausdrücke α_i beschreiben genau die Sprache, die der endliche Automat mit obigem Zustandsgraph akzeptiert? (Es kann mehrere solche Ausdrücke geben.)

α_1 = b(a)*b
α_2 = b(a)*b(b)*
α_3 = (b(a)*b(b)* | a(a| b)*a(b)*)
α_4 = b(a)*(b)*
α_5 = b(a)*(b)*b

Aufgabe 2

2.1 „Eine Sprache L ist genau dann regulär, wenn *Bedingung*."
Geben Sie vier verschiedene Bedingungen an, für die diese Aussage korrekt ist!

2.2 Gibt es drei Sprachen L_1, L_2, L_3 mit $L_1 \subseteq L_2 \subseteq L_3$, so dass L_1 und L_3 vom Typ 2 (kontextfrei) sind, aber L_2 nicht?

Geben Sie entweder drei solche Sprachen an oder einen Beweis, dass dies nicht möglich ist!

Fortsetzung nächste Seite!

Aufgabe 3

Ein Palindrom ist ein Wort, das vorwärts und rückwärts gelesen gleich ist, z.B. *anna* oder *uhu*. Das leere Wort ε ist auch ein Palindrom.

Sei $\Sigma = \{a, b\}$ und sei L die Menge aller Wörter über Σ, die Palindrome sind.

3.1 Geben Sie eine Grammatik $G = (V, \Sigma, P, S)$ an mit $L = L(G)$!

3.2 Beweisen Sie, dass L nicht regulär ist!

3.3 Geben Sie eine deterministische Turingmaschine $M = (Z, \Sigma, \Gamma, \delta, s, \_, \{e\})$ an, die genau die Palindrome über Σ akzeptiert, für die also $T(M) = L$ ist!

Dabei ist Z die Zustandsmenge, $\Sigma = \{a, b\}$ das Eingabealphabet, $\Gamma = \{a, b, \_\}$ das Bandalphabet, _ das Leerzeichen, $s \in Z$ der Anfangszustand, $e \in Z$ der Endzustand (hier reicht ein einziger) und δ die Überführungsfunktion.

Geben Sie zu jedem von Ihnen verwendeten Zustand in Z eine informelle Erläuterung, welchem Zweck der Zustand dienen soll, etwa in folgender Form: „Zustand r_a: nach rechts laufen bis Wortende, anschließend prüfen, ob Wort mit a endet“!

3.4 Geben Sie die Folge der Konfigurationen an, die M jeweils durchläuft für das Eingabewort *aba*, für das Eingabewort *baab* und für das Eingabewort *abaa*!

Aufgabe 4

Sei Σ ein Alphabet.

4.1 Geben Sie die Definition dafür an, dass eine Menge $L \subseteq \Sigma^*$ rekursiv aufzählbar ist!

4.2 Geben Sie die Definition dafür an, dass eine Menge $L \subseteq \Sigma^*$ semi-entscheidbar ist!

4.3 Wie hängen die in den Teilaufgaben 4.1 und 4.2 definierten Eigenschaften zusammen?

4.4 Beweisen Sie: Sind L_1 und L_2 rekursiv aufzählbar, dann ist auch $L_1 \cup L_2$ rekursiv aufzählbar!

4.5 Beweisen Sie: Sind L_1 und L_2 rekursiv aufzählbar, dann ist auch $L_1 \cap L_2$ rekursiv aufzählbar!

Hinweis: Benutzen Sie den Zusammenhang zwischen rekursiv aufzählbar und semi-entscheidbar!