Übungs-Termine (je 10.15–11.45 Uhr, Raum 1.035):
Gruppe 1 und 3: Di. 14.7., Gruppe 2: Mi. 15.7.

Abgabe der Übungsblätter am 10.7. vor der Vorlesung oder im Foyer, 1. Etage.

Aufgabe 1: (2+3 Punkte)

In der Vorlesung wurde eine Turingmaschine definiert, die die Funktion $h_1 : \{|\}^* \to \{|\}^*$ mit $h_1(|^m) = |^{m+1}$ berechnet und eine dazu analoge Turingmaschine, die dieses für Binärzahlen tut.

(a) Konstruieren Sie eine Turingmaschine mit Eingabealphabet $\Sigma = \{|, +\}$, die die Funktion

$$h_2 : \Sigma^* \to \Gamma^* \text{ mit } h_2(|^m + |^n) \;=\; |^m + |^n \equiv |^{m+n}$$

berechnet. Wählen Sie ein geeignetes Γ.

(b) Konstruieren Sie eine Turingmaschine mit Eingabealphabet $\Sigma' = \{0, 1, +\}$, die Zahlen in Binärdarstellung addieren kann, d. h. die Funktion

$h_3 : \Sigma^* \to \Gamma^*,\ h_3(u{+}v) \;=\; u{+}v \equiv w,$ mit $u, v, w \in \{0,1\}^*$ und $u{+}v = w$

berechnet für die übliche Interpretation der u, v, w als Binärzahlen.

Geben Sie zunächst eine umgangssprachliche Beschreibung der Arbeitsweisen Ihrer Turingmaschinen an.

Aufgabe 2: (3+1 Punkte)

Gegeben sei folgendes Programm, das aus einer Eingabe $x \in \mathbb{N}$ die Ausgabe $y \in \mathbb{N}$ berechnet.

```
read(x);
y:=1; z:=2;
while z≤x do
   z:=z+z; y:=y+1
od;
write(y)
```

(a) Entwerfen Sie eine Turingmaschine für obiges Programm. Die Ein- und Ausgabe soll dabei unär geschehen, d. h. Eingabealphabet $\Sigma = \{|\}$. Beschreiben Sie zunächst umgangssprachlich ihre Turingmaschine und wie Sie die drei Variablen x, y und z verwalten.

(b) Welche Funktion berechnet das Programm?

Aufgabe 3: (2+1 Punkte)

(a) Entwerfen Sie eine Turingmaschine, die zunächst die unäre Eingabe $|^n$, $n \geq 0$ in eine Binärzahl ohne führende Nullen wandelt und dann die Ziffern 0 und 1 der Binärzahl durch | ersetzt. Die Eingabe ||||| soll so z. B. zunächst in 101 und dann in ||| gewandelt werden.

(b) Welche Funktion berechnet das Programm?

Aufgabe 4: (1+1 Punkte)

Turingmaschinen können nicht nur Funktionen berechnen sondern auch Sprachen entscheiden. Betrachte die Sprache $L = \{a^n b^n | n \geq 0\}$ über $\Sigma = \{a, b\}$.

(a) Entwerfen Sie eine Turingmaschine, deren Haltebereich (oder Definitionsbereich) genau die Sprache L ist. Bestimmen Sie die Anzahl der Schritte, die zum Akzeptieren von Wörtern aus L benötigt werden.

(b) Eine 2-Band-Turingmaschine besitzt 2 Bänder mit je einem Kopf, die in jedem Schritt unabhängig voneinander bewegt werden können. Entwerfen Sie eine 2-Band-Turingmaschine für L. Bestimmen Sie die Anzahl der Schritte, die zum Akzeptieren von Wörtern aus L benötigt werden.

Zusatzaufgabe: (4 Punkte)

In jeder natürlichen Sprache kann man innere Widersprüche (Antimonien) formulieren. Als einfachster Satz gilt die Formulierung „Dieser Satz ist falsch“. Denn dieser Satz macht eine Wahrheitsaussage über sich selbst, die nicht zutreffen kann: Wäre der Satz nämlich richtig, dann wäre er nach seiner eigenen Aussage falsch, und wäre er falsch, dann besagt er inhaltlich, daß er nicht falsch sein kann, also richtig ist. Diese etwas gekünstelte Antinomie läßt sich aber auf konkretere Fälle übertragen. Bekannt ist die Barbier-Antinomie. Der Barbier in einem Dorf sei folgendermaßen definiert:
Definition: Der Barbier ist der Mann im Dorf, der alle die Männer im Dorf rasiert, die sich nicht selbst rasieren.
Diese Definition ist nicht korrekt, d. h. sie enthält einen Widerspruch. Dieser entsteht durch Selbstanwendung: Rasiert der Barbier sich selbst? Falls ja, dann darf er nicht vom Barbier, also von sich selbst rasiert werden, und falls nein, dann gehört er zu den Personen, die vom Babier, also von ihm selbst, rasiert werden. Nach obiger Definition rasiert der Barbier sich also genau dann selbst, wenn er sich nicht selbst rasiert. Dieser Widerspruch läßt sich auflösen, wenn man die Selbstanwendung der definierten Objekte verhindert:

Definition: Der Barbier ist der Mann, der alle die Männer im Dorf rasiert, die sich nicht selbst rasieren.
Bei dieser Definition wird der Barbier aus dem Dorf herausgenommen. Wenn er im Dorf lebt, bleibt der Widerspruch; wenn er aber nicht im Dorf lebt, sondern von außen zum Zwecke des Rasierens anreist, ist die Definition „wohldefiniert“.

Nun zur Aufgabe: Wir betrachten alle mathematischen Funktionen über den natürlichen Zahlen, die genau einen Parameter besitzen und einen Wert liefern, also die Abbildungen $f : \mathbb{N}_0 \to \mathbb{N}_0$. Diese Menge ist für unsere Zwecke aber noch viel zu groß. Wir betrachten daher nur die Funktionen, deren Berechnungen man in einer gegebenen natürlichen (oder formalen) Sprache beschreiben kann. Es ist klar, daß man diese Teilmenge der Funktionen herausgreifen kann, denn einstellige zahlentheoretische Funktionen werden stets durch eine Berechnungsvorschrift definiert (z. B. ganzzahlige Wurzel, n-te Primzahl, Fakultät usw.).
So erhalten wir eine sogenannte „aufzählbare“ Menge von Funktionen: Wir schreiben nacheinander alle möglichen Textfolgen der deutschen (oder einer formalen) Sprache auf und sondern genau die Texte heraus, die die Berechnung einer einstelligen zahlentheoretischen Funktion darstellen. Man mache sich klar, daß dies zwar ein außerordentlich mühseliges und Jahrmillionen dauerndes Unterfangen ist, aber aus prinzipieller Sicht tatsächlich ausführbar ist. Wir erhalten dann eine unendliche Folge von Texten $T_1, T_2, T_3, \ldots$, wobei T_1 der kürzeste Text in deutscher Sprache ist, der eine einstellige zahlentheoretische Funktion berechnet, T_2 der nächst kürzeste Text usw. Zu T_1 gehört eine Funktion $f_1 : \mathbb{N}_0 \to \mathbb{N}_0$, zu T_2 eine Funktion $f_2 : \mathbb{N}_0 \to \mathbb{N}_0$ usw. Dann definieren wir folgende Funktion $g : \mathbb{N}_0 \to \mathbb{N}_0$

$$g(n) = f_n(n) + 1, \text{ für alle natürlichen Zahlen } n.$$

Zweifellos kann man einen Text T angeben, der die Berechnung der Funktion g beschreibt: Man formuliert zunächst aus, wie die Texte $T_1, T_2, T_3, \ldots$ erzeugt wurden, dann ermittelt man zu vorgegebenem n den Text T_n, und anschließend wertet man den Text T_n für die Zahl n aus und addiert die Zahl 1 zum Ergebnis. Offenbar gibt es also einen Text T zur Berechnung von g. Dieser Text muß in der Folge der Texte $T_1, T_2, T_3, \ldots$ auftreten, sagen wir $T = T_k$ für eine geeignete Zahl k. Dann gilt aber:

$$f_k(k) = g(k) = f_k(k) + 1.$$

Wir erhalten einen Widerspruch, obwohl die obige Argumentation logisch korrekt zu sein scheint. Wo liegt der Fehler?