BuK Abgabe 3 | Gruppe 17

Malte Meng (354529) , Charel Ernster (318949), Sebastian Witt (354738)

November 9, 2016

1 Aufgabe 3.1

Die Turingmaschine funktioniert nach folgendem Muster:
Maschine besitzt vier Zustände um die Anzahlen von 0 und 1 zu zählen: $(q0odd, q1odd, q0even, q1even)$, es gibt noch eine Schrittgröße die bei 1 anfängt und pro Iteration verdoppelt wird, außerdem wird noch die Schrittrichtung gespeichert und wechselt nach jeder Iteration. Jede Iteration läuft folgendermaßen ab:

1. Gehe in Schrittrichtung mit der Schrittgröße, wenn dass Zeichen währenddessen unter dem Kopf wechselt merke dir ob Zustand odd oder even ist. Nach jedem erfolgreichen Schritt wird der Zustand gewechselt.
2. Gehe in Schrittrichtung mit der Schrittgröße weiter bis ein Blank erreicht wird. Ist dann der Zustand nicht gleich dem gemerkten Zustand, wird das Wort abgelehnt. Nach jedem erfolgreichen Schritt wird der Zustand gewechselt.
3. Ändere die Schrittrichtung, verdopple die Schrittlänge und setzte die Zustände auf Odd. Nächste Iteration.

Der Algorithmus beendet erfolgreich, wenn die Schrittlänge länger ist als die Anzahl von "0" bzw "1", also während des ersten Schrittes das Vorzeichen gewechselt wird. (Ausnahme bei $L = 0^*|L = 1^*$ vor Wechsel wird bereits blank erreicht)
Das Grundkonzept funktioniert indem verglichen wird ob:
$(n_0 mod 2^i > 0) == (n_1 mod 2^i > 0)$
nach jeder Iteration i

Der Algorithmus hat den Zeitbedarf $O(m \log m)$, da pro Iteration m Schritte gemacht werden und der Algorithmus im worst-case nach $\log m$ Schritten terminiert, wenn Schrittlänge $> n_0$.

2 Aufgabe 3.2

```
"RAM Zweierlogarithmus"
1  CLOAD 2
2  STORE 2
3  CLOAD n
4  STORE 1
5  IF c(0) = 0 THEN GOTO 13
6  LOAD 1
7  DIV 2
8  STORE 1
9  LOAD 3
10 CADD 1
11 STORE 3
12 GOTO 5
13 LOAD 3
14 STORE 1
15 END
```

Die RAM Läd zunächst die Eingabe an erste Stelle. Dann wird in jedem Schritt die Eingabe halbiert, bis sie Leer ist. Die Schritte werden in Register 3 gespeichert. Somit wird gezählt an welcher Stelle das höchste bit ist, welches die Zweierpotenz bestimmt. Ist die Stelle bestimmt, wird der Zähler ins erste Register geladen und ausgegeben.

3 Aufgabe 3.3

Idee: Äquivalenz von $\mathbb{N} = \mathbb{N}^*$ zeigen.

$a.b = a * 10^n + b$ mit $a, b \in \mathbb{N} \wedge n =$ Anzahl Stellen von b.
$\implies \mathbb{N}.\mathbb{N} \in \mathbb{N}$ "." steht für die Konkatenation bzw 1.2 = 12.
$\implies \mathbb{N}.\mathbb{N}.\mathbb{N}... \in \mathbb{N} \implies \mathbb{N} = \mathbb{N}^*$
$\mathbb{N}$ ist abzählbar somit auch $\mathbb{N}^*$

4 Aufgabe 3.4

1. Konstruiere eine TM die $\langle M \rangle w$ simuliert. Simuliere 42 Schritte und akzeptiere, wenn bis dahin $\langle M \rangle w$ hält.$\square$

2. Annahme: Es gibt eine Maschine die $H_{\leq 42}$ entscheidet.
   Somit lässt sich eine Maschine M_H mit Unterprogramm $H_{\geq 42}$ konstruieren, so dass M_H das Halteproblem entscheidet.
   M_H ist wie folgt aufgebaut:
   Simuliere die ersten 42 Schritte mit $H_{\leq 42}$ aus a). Wenn in die Maschine in den ersten 42 Schritten stoppt akzeptiere. Wenn nicht wird das Wort an

$H_{\geq 42}$ weitergeleitet. Akzeptiert/Ablehnt diese Akzeptiert/Ablehnt auch M_H.
Diese Maschine entspricht dem Halteproblem:

$$M_H = \left\{ \begin{matrix} Schritte \leq 42 \mid \text{Akzeptiere wenn hält, sonst nicht} \\ Schritte > 42 \mid \text{Akzeptiere wenn hält, sonst nicht} \end{matrix} \right\} = H$$

Da das Halteproblem, wie in der Vorlesung gezeigt, nicht entscheidbar ist, kommt es zum Widerspruch. Somit kann es keine Maschine $H_{\geq 42}$ geben. $\square$