Theoretische Informatik - Übungsblatt 2
Übungsgruppe 7 - Mi, 16-18 Uhr
Elena Kvassova

Jan Hendrik Dithmar (Matrikelnummer 2031259)
Dirk Heine (Matrikelnummer 2030941)

Aufgabe 1

Sind L und $\overline{L}$ semi-entscheidbar, so hält die TM auf allen Eingaben $w \in L$ an. Somit ist die einzigste für Entscheidbarkeit fehlende Bedingung erfüllt. $\square$

Aufgabe 2

Wir verwenden eine 1-Band-Turingmaschine (1 Lese-/Schreibkopf), die auf diesem Band zwei Spuren verwaltet. Dies läßt sich sehr einfach durch ein neues Bandalphabet realisieren, wobei jeder Bandposition ein Tupel (a, b) zugeordnet wird. Ferner führen wir ein neues Markiersymbol Ψ in das Bandalphabet ein.

$$\Gamma' = \Gamma^2 \vee \{\Psi\}$$

Notation: Im folgenden verwenden wir die Bezeichnung UTM für die zu emulierende unbeschränkte Turingmaschine und BTM für die zu bauende einseitig beschränkte Turingmaschine.

Initialisierung: Alle Werte auf der Startposition und rechts von der Startposition auf dem Band unserer UTM werden ab der Startposition unserer BTM in die erste Komponente des Tupels eingetragen. In die zweite Komponente des Tupels an der Position 1 (Startposition) unserer BTM tragen wir das neue Symbol Ψ ein. Alle Werte links von der Startposition auf dem Band unserer UTM werden ab Position 2 in die zweite Komponente des Tupels auf dem Band der BTM eingetragen (Reihenfolge an der Startposition gespiegelt).

Weitere neue Definitionen: Wir führen eine neue Zustandsmenge Q' ein. Wir betrachten die neuen Zustände als Tupel eines Zustandes der UTM und einem Wert aus $\{0, 1\}$. Dieser neue Wert gibt uns an, ob wir uns mit unserer BTM auf einer Lese-/Schreibposition links von bzw. auf der Startposition der UTM befinden (Wert 0, erste Tupelkomponente auf dem Band unserer BTM ist zu verwenden) oder rechts davon (Wert 1, zweite Tupelkomponente auf dem Band unserer BTM ist zu verwenden) befinden.

$$Q' = Q \times \{0, 1\}$$

Somit brauchen wir ebenfalls einen neuen Startzustand q_0'. Dieser ergibt sich trivial, da wir mit unserer BTM im selben Zustand beginnen wollen, in dem auch die UTM beginnen würde.

$$q_0' = (q_0, 0)$$

Ferner muss die Menge der akzeptierende Endzustände F angepaßt werden. Dies ist ebenfalls trivial, da es für den akzeptierenden Endzustand egal ist, ob wir uns links oder rechts von bzw. auf der Startposition der UTM befinden.

$$F' = \bigcup_{q \in F} \{(q, 0), (q, 1)\}$$

Anschließend muss die Übergangsfunktion δ angepaßt werden.

$$\delta'(\tilde{q}, \tilde{a}) = \delta'((q, r), (a, b)) = ((q', r'), (a', b'), d) = (\tilde{q}', \tilde{a}', d)$$

$$\begin{array}{rcl} \tilde{q}, \tilde{q}' \in Q' & \iff & \tilde{q}, \tilde{q}' \in Q \times \{0, 1\} \\ \tilde{a}, \tilde{a}' \in \Gamma' & \iff & \tilde{a}, \tilde{a}' \in \Gamma \times \Gamma \\ q, q' \in Q & & r, r' \in \{0, 1\} \\ a, a', b, b' \in \Gamma & & d \in \{L, N, R\} \end{array}$$

Im folgenden beschreiben wir mit Pseudocode die Realisierung der neuen Übergangsfunktion δ', welche ihre Eingabe in den Symbolen q, r, a, b erhält und das Ergebnis in den Symbolen q', r', a', b', d zurückgibt.

```
if (r=0) {

    (q', a', d) := δ(q, a)
    b' := b
    if ((d = L) ∧ (b = Ψ)) {
        d  := R
        r' := 1
    } else {
        r' := 0
    }

} else {

    if (b = Ψ) {
        q' := q
        a' := a
        b' := b
        r' := 0
        d  := N
    } else {
        (q', b', d) := δ(q, b)
        a' := a
        r' := 1
        if (d = L) {
            d := R
        } else {
            if (d = R) {
                d := L
            }
        }
    }

}
```

Die gesuchte Turingmaschine ist mit den oben eingeführten Symbolen

$$M' = (Q', \Sigma, \Gamma', q_0', \delta', F')$$

Aufgabe 3

Vorbetrachtung: Die Turingmaschine M gerät in eine Schleife, wenn M zum zweiten mal eine bestimmte Konfiguration erreicht. Dieses Kriterium gilt nur, wenn die Turingmaschine auf einem beidseitig beschränktem Band arbeitet, wie es in der Aufgabe gegeben ist.
$|\Gamma|^{s(n)}$ ist die Anzahl der möglichen verschiedenen Bandinhalte unter der Voraussetzung, dass nur die Bandzellen $1 \dots s(n)$ verwendet werden. $s(n)$ ist die Anzahl der möglichen Positionen, die der Lesekopf einnehmen kann. Somit ist $|Q| \cdot |\Gamma|^{s(n)} \cdot s(n)$ die Anzahl der möglichen verschiedenen Konfigurationen, die die Turingmaschine M auf einer Eingabe der Länge n annehmen kann. Nach obiger Vorbetrachtung darf die Turingmaschine, damit sie nicht in eine Schleife eintreten kann und somit überhaupt stoppen kann, die selbe Konfiguration nicht zweimal erreichen. Nach

Voraussetzung soll M auf einer Eingabe der Länge n stoppen. Somit stoppt sie spätestens nach $|Q| \cdot |\Gamma|^{s(n)} \cdot s(n)$ Schritten, da es keine weitere noch nicht vorher erreichte Konfiguration gibt. □

Aufgabe 4

Um zu zeigen, dass $f^{-1}(1)$ rekursiv aufzählbar ist, müssen wir zeigen, dass man eine Turingmaschine konstruieren kann, welche diese Sprache akzeptiert.
Idee für Turingmaschine:

- Turingmaschine produziert Prefix p der Dezimaldarstellung von π
- sucht Eingabe in p
- wenn erfolgreich, Eingabe akzeptieren
- anderenfalls, wiederhole mit größerem Prefix

Die Turingmaschine die den Prefix erzeugt, ist nach Aufgabenstellung gegeben (mit endlicher Laufzeit). Eine vergleichende Suche ist trivialerweise mit einer Turingmaschine realisierbar. Da der jeweils durchsuchte Prefix endlich ist, läßt sich die Suche innerhalb des Prefixes auch in endlicher Zeit realisieren. □

Aufgabe 5

(a) Sei eine Funktion gegeben, die wie folgt von $\mathbb{N} \to \mathbb{N}^2$ bzw. von $\mathbb{N}^2 \to \mathbb{N}$ abbildet. Die fortlaufenden Werte aus $\mathbb{N}$ werden diagonal von rechts oben nach links unten immer linienweise eingetragen. Die Funktion ist bijektiv, da die Zuordnung zwischen $\mathbb{N}$ und $\mathbb{N}^2$ offensichtlich in beide Richtungen eindeutig ist und auch jedes Element beider Mengen erreicht wird.

	0	1	2	3	4
0	0	1	5	6	10
1	2	4	7	11	
2	3	8	12		
3	9	13			
4	14				

Wir zeigen die Realisierbarkeit der Berechnung der Funktion und der Umkehrfunktion mittels RAM durch Implementierung eines Programms, das diese Abbildungen berechnet.

Berechnung von $f_2(n) = (i, j)$ Sei `c(1)=n` gegeben. Das Ergebnis steht nach der Ausführung in `c(2)=i` und `c(3)=j`.

```
// wenn c(0)=0, dann fertig
1: LOAD 1
2: IF c(0) = 0 GOTO 23
// wenn c(2)=i=0, dann neue Diagonale beginnen
3: LOAD 2
4: if c(0) = 0 GOTO 15
// sonst, diagonal nach links unten wandern
5: CSUB 1
6: STORE 2
7: LOAD 3
8: CADD 1
```

```
9: STORE 3
// c(1)=n dekrementieren und neu vergleichen
10: LOAD 1
11: CSUB 1
12: STORE 1
13: GOTO 2
// neue Diagonale beginnen
14: LOAD 3
15: CADD 1
16: STORE 2
17: CLOAD 0
18: STORE 3
// c(1)=n dekrementieren und neu vergleichen
19: LOAD 1
20: CSUB 1
21: STORE 1
22: GOTO 2
// Berechnung beendet
23: END
```

Berechnung von $f_2^{-1}(i,j) = n$ Seien `c(1)=i` und `c(2)=j` gegeben. Das Ergebnis steht nach der Ausführung in `c(3)=n`.

```
// wenn c(1)=c(4) und c(2)=c(5) dann fertig
1: LOAD 1
2: SUB 4
3: STORE 6
4: LOAD 4
5: SUB 1
6: ADD 6
7: IF c(0) > 0 GOTO 14
8: LOAD 2
9: SUB 5
10: STORE 6
11: LOAD 5
12: SUB 2
13: ADD 6
14: IF c(0) = 0 GOTO 32
// noch keine Übereinstimmung
// inkrementiere c(3)=n
15: LOAD 3
16: CADD 1
17: STORE 3
// wenn c(4)=i'=0, neue Diagonale beginnen
18: LOAD 4
19: if c(0) = 0 GOTO 26
// diagonal nach links unten wandern
20: CSUB 1
21: STORE 4
22: LOAD 5
23: CADD 1
24: STORE 5
25: GOTO 1
// neue Diagonale beginnen
26: LOAD 5
```

```
27: CADD 1
28: STORE 4
29: CLOAD 0
30: STORE 5
31: GOTO 1
// Berechnung beendet
32: END
```

Beweis der Berechenbarkeit mittels Implementation erbracht. □

(b) Das Abbildungsprinzip aus Aufgabenteil (a) läßt sich trivialerweise auf $\mathbb{N} \rightarrow \mathbb{N}^k$ erweitern. □