Prof. J. Křetínský
M. Weininger, T. Meggendorfer Technische Universität München
KEINE ABGABE

Einführung in die theoretische Informatik

Sommersemester 2018 – Übungsblatt 10

Selbstständige Vorbereitung

Bereiten Sie sich auf die Tutorgruppen selbstständig vor, indem Sie die Aufgaben 10.1 bis 10.5 ansehen.
Lösen Sie insbesondere Aufgabe 10.3, da Aufgabe 10.6 darauf aufbaut.
Lösen Sie außerdem unbedingt Aufgabe 10.5a, da 10.7 bis 10.9 darauf aufbauen.

AUFGABE 10.1. (*Wichtige Begriffe*) Stufe A

Überprüfen Sie, dass Sie die Folgenden Begriffe korrekt definieren können.

- WHILE-Programme
- GOTO-Programme
- enscheidbar
- charakteristische Funktion
- spezielles Halteproblem
- allgemeines Halteproblem
- reduzierbar
- semi-entscheidbar
- rekursiv-aufzählbar

AUFGABE 10.2. (*Permutations-TM*) Stufe C

Geben Sie eine nichtdeterministische 1-Band-TM mit $\Sigma = \{\mathsf{a}, \mathsf{b}\}$ an, die nichtdeterministisch die Eingabe permutiert. Ist w die Eingabe, dann terminiert die TM auf *jedem* Rechenpfad, und zu jeder Permutation w' von w gibt es mindestens eine Rechnung der TM, an deren Ende der Bandinhalt gerade w' ist.
Beispiel: Auf Eingabe $w = \mathsf{aabb}$ soll die TM für jede der möglichen Permutation $w' \in \{\mathsf{abab}, \mathsf{abba}, \mathsf{baba}, \mathsf{bbaa}, \mathsf{baab}, \mathsf{aabb}\}$ jeweils mindestens einen Rechenpfad besitzen, an deren Ende genau w' auf dem Band steht.

AUFGABE 10.3. (*Entscheidbarkeit vs. Berechenbarkeit*) Stufe B

Ordnen Sie die folgenden Satzanfänge den Satzenden so zu, dass richtige Aussagen entstehen. Sei dazu $\mathsf{A}, \mathsf{B} \subseteq \{0,1\}^*$:

(a) Die Funktion χ_{A} ist berechenbar,
(b) Die Funktion χ'_{A} ist berechenbar,
(c) A ist entscheidbar,
(d) B ist nicht entscheidbar,

(i) wenn $\mathsf{A} \leq \mathsf{B}$ gilt und B entscheidbar ist.
(ii) wenn A rekursiv aufzählbar ist.
(iii) wenn A entscheidbar ist.
(iv) wenn $\mathsf{A} \leq \mathsf{B}$ gilt und A nicht entscheidbar ist.

AUFGABE 10.4. (*Länge von Wörtern*) Stufe B

Diskutieren Sie, wie viele Schritte eine Turing-Maschine mindestens machen muss, um zu entscheiden, ob für eine Eingabe w gilt: $|w| \geq 314$.

AUFGABE 10.5. (*GOTO und WHILE*) Stufe C

Geben Sie für jede der folgenden Funktionen sowohl ein WHILE-Programm als auch ein GOTO-Programm an, das die jeweilige Funktion implementiert:

(a) $f(x_1, x_2) = x_1 \bmod x_2$

(b) $g(x_1) = \begin{cases} \max\{k \in \mathbb{N}_0 \mid \frac{x_1}{2^k} \in \mathbb{N}_0\} & \text{falls } x_1 > 0 \\ \perp & \text{sonst} \end{cases}$

(c) $h(x_1, x_2) = (2x_1 + 1) \cdot 2^{x_2}$

Hinweise:

- Verwenden Sie für Teilaufgabe (a) nur das Grundschema für WHILE- und GOTO-Programme und folgende Abkürzungen: $x_i := x_j + x_k$, $x_i := x_j - x_k$, $x_i := n$, $x_i := x_j$.
- Für Teilaufgaben (b) und (c) dürfen Sie allen syntaktischen Zucker aus der Vorlesung, sowie den Befehl MOD verwenden.
- Halten Sie sich an die Konventionen aus der Vorlesung: Soll ein WHILE-Programm eine Funktion $F\colon \mathbb{N}_0^k \to \mathbb{N}_0$ berechnen, so werden die Variablen $x_1, \ldots, x_k$ entsprechend mit den Eingabewerten $n_1, \ldots, n_k \in \mathbb{N}_0$ initialisiert, während alle anderen Variablen zu Beginn auf 0 initialisiert werden. Nach Terminierung speichert die Variable x_0 den Funktionswert $F(n_1, \ldots, n_k)$.

Stufe B/C

AUFGABE 10.6. (*Entscheidbarkeit*)
Entscheiden Sie, ob die folgenden Behauptungen korrekt oder inkorrekt sind. Begründen Sie dann Ihre Antworten wie folgt: Wenn L entscheidbar bzw. semi-entscheidbar ist, beschreiben Sie einen Algorithmus, der die charakteristische Funktion χ_L bzw. die Funktion χ'_L berechnet. Wenn L unentscheidbar ist, leiten Sie einen Widerspruch zu einem Ergebnis der Vorlesung ab.

(a) Wenn A und B entscheidbare Sprachen sind, dann ist $A \cap B$ entscheidbar.
(b) Wenn A und $A \cup B$ entscheidbar sind, dann ist B entscheidbar.
(c) Das Problem, ob $L_H(M) \neq \emptyset$ für eine gegebene Turingmaschine M gilt, ist semi-entscheidbar.
(d) Das Problem, ob $L_H(M) = \emptyset$ für eine gegebene Turingmaschine M gilt, ist semi-entscheidbar.

AUFGABE 10.7. (*WHILE → TM*) Stufe C
Überlegen Sie, was das folgende WHILE-Programm berechnet.
Skizzieren Sie dann eine TM, die das While-Programm umsetzt.

```
x0 = 0
WHILE x1 <= x2 DO
  x1 := x1 + x2;
  x0 := x0 + 1;
END;
```

AUFGABE 10.8. (*Collatz-Vermutung*) Stufe D
Zu einem Startwert $a_0 \in \mathbb{N}$ definieren wir eine Folge $(a_n)_{n \in \mathbb{N}_0}$ wie folgt:

$$a_{n+1} := \begin{cases} a_n/2 & a_n \text{ gerade} \\ 3a_n + 1 & a_n \text{ ungerade} \end{cases}$$

Die seit 1937 unbewiesene Collatz-Vermutung besagt:

Für alle positiven Startwerte $a_0 \in \mathbb{N}$ gibt es einen Index $i \in \mathbb{N}_0$, so dass $a_i = 1$.

Nehmen Sie an, es gibt ein Programm N, welches als Eingabe ein WHILE-Programm P mit genau einer Eingabevariable nimmt und zu jedem solchen P angibt, ob P die Nullfunktion berechnet. *Zeigen Sie, dass Sie dann die Collatz-Vermutung beweisen oder widerlegen können.*
Hinweise:

- Geben Sie auch das WHILE-Programm P, das sie für Ihren Beweis verwendet haben, an.
- Sie dürfen jede Syntax, die in den Folien für WHILE-Programme eingeführt worden ist, verwenden.

AUFGABE 10.9. (*GOTO mit Stack*) Stufe D
Wir erweitern GOTO-Programme um die Möglichkeit, Variablenwerte auf einem globalen Stack zwischenzuspeichern. Mittels der Anweisung **PUSH** x_i wird der aktuelle Wert der Variable x_i auf den Stack gelegt, mittels $x_i :=$ **POP** wird der aktuell oberste Wert vom Stack genommen und in der Variable x_i gespeichert – sollte der Stack aktuell keine Werte enthalten, wird x_i auf 0 gesetzt. Ob der Stack aktuell einen Wert enthält, kann mittels $x_i :=$ **EMPTY** erfragt werden, wobei x_i auf 1 gesetzt wird, falls der Stack leer ist, ansonsten wird x_i auf 0 gesetzt.
Beispiel:

```
M1: IF x1 = 0 GOTO M2;
    IF x1 = 1 GOTO M3;
    x1 := x1 - 1;
    PUSH x1;
    x1 := x1 - 1;
    GOTO M1;
M2:  x0 := x0 + 0;
    GOTO M4;
M3:  x0 := x0 + 1;
    GOTO M4;
M4:  x1 := EMPTY;
    IF x1 = 1 GOTO M5;
    x1 := POP;
    GOTO M1;
M5: HALT
```

(a) Bestimmen Sie die Funktion $\mathbb{N}_0 \to \mathbb{N}_0$, die von Programm aus dem Beispiel berechnet wird.
(b) Skizzieren Sie, wie sich jedes GOTO-Programm mit Stack in ein GOTO-Programm ohne Stack übersetzen lässt.