Prof. T. Nipkow
K. Kappelmann, J. Rädle, L. Stevens

KEINE ABGABE

Technische Universität München
Lehrstuhl für Logik und Verifikation

Einführung in die Theoretische Informatik

Sommersemester 2020 – Übungsblatt Lösungsskizze 8

AUFGABE 8.1. (*Wichtige Begriffe*) Stufe A

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

- WHILE-Programme
- WHILE und GOTO-Programme
- Entscheidbarkeit
- charakteristische Funktion
- Gödelisierung
- spezielles Halteproblem
- allgemeines Halteproblem
- Halteproblem auf leerem Band
- Reduktion

AUFGABE 8.2. (*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) A ist entscheidbar,
(c) B ist nicht entscheidbar,

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

Lösungsskizze

(a) $\rightarrow$ (i)/(ii), (b) $\rightarrow$ (i)/(ii), (c) $\rightarrow$ (iii).

AUFGABE 8.3. (*Entscheidbarkeit*) Stufe B

Entscheiden Sie, ob die folgenden Behauptungen korrekt oder inkorrekt sind. Begründen Sie dann Ihre Antworten wie folgt: Wenn L entscheidbar ist, beschreiben Sie einen Algorithmus, der die charakteristische 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 $\mathsf{A} \cap \mathsf{B}$ entscheidbar.
(b) Wenn A und $\mathsf{A} \cup \mathsf{B}$ entscheidbar sind, dann ist B entscheidbar.

Lösungsskizze

(a) Korrekt. Wenn A und B entscheidbar sind, dann sind die charakteristischen Funktionen χ_A und χ_B berechenbar. Wir definieren

$$\chi_{A\cap B} = \begin{cases} 1, & \text{wenn } \chi_A = 1 \wedge \chi_B = 1, \\ 0 & \text{sonst.} \end{cases}$$

Diese Funktion ist intuitiv berechenbar und eine charakteristische Funktion von $A \cap B$, da

$$\chi_{A\cap B}(w) = 1 \Leftrightarrow \chi_A(w) = 1 \wedge \chi_B(w) = 1 \Leftrightarrow w \in A \wedge w \in B \Leftrightarrow w \in A \cap B.$$

Folglich ist die Menge $A \cap B$ entscheidbar.

Alternativ mit Turingmaschinen:
Sei T_A DTM, die A entscheidet, T_B DTM, die B entscheidet. DTM zu $A \cap B$: Gegeben x, berechne $T_A[x]$. Falls $T_A[x]$ mit 0 auf dem Band hält, lehne x ab, ansonsten berechne $T_B[x]$. Hält $T_B[x]$ mit 0 auf dem Band, lehne x ab, sonst akzeptiere x. Da T_A, T_B die jeweiligen Mengen entscheiden, terminieren beide DTM, womit auch die DTM zu $A \cap B$ stets mit dem korrekten Ergebnis terminiert. Damit ist $A \cap B$ entscheidbar.

(b) Inkorrekt. Sei $B = H_0 = \{w \in \{0,1\}^* \mid M_w[\varepsilon]\downarrow\} \subsetneq \{0,1\}^*$ das Halteproblem auf leerem Band und $A = \{0,1\}^*$. Dann sind A und $A \cup B = \{0,1\}^*$ entscheidbar, aber B nicht.

AUFGABE 8.4. (*Reduktionen*) Stufe B

Entscheiden Sie, ob folgende Aussagen korrekt oder inkorrekt sind, und begründen Sie Ihre Antwort, indem Sie einen Beweis bzw. ein passendes Gegenbeispiel angeben. Sei $\Sigma = \{0,1\}$.

(a) $\forall A \subseteq \Sigma^*.\ A \leq \Sigma^*$
(b) $\forall A, B \subseteq \Sigma^*.\ A \leq B \iff \overline{A} \leq \overline{B}$
(c) $\forall A, B, C \subseteq \Sigma^*.\ A \leq B \wedge B \leq C \implies A \leq C$

Lösungsskizze

(a) Falsch. Sei $A = \emptyset$. Damit $\overline{A} = \Sigma^*$. Dann muss für eine Reduktionsfunktion f gelten: $\forall x \in \overline{A}.\ f(x) \notin \Sigma^*$. Eine solche Funktion f existiert aber nicht.

(b) Wahr. Gelte $A \leq B$. Dann existiert ein totales und berechenbares f mit: $\forall x \in \Sigma^*. x \in A \Leftrightarrow f(x) \in B$. Somit gilt auch: $\forall x \in \Sigma^*. x \in \overline{A} \Leftrightarrow f(x) \in \overline{B}$. Daraus folgt dann $\overline{A} \leq \overline{B}$. Die Rückrichtung geht analog.

(c) Wahr. Seien f und g Reduktionen von $A \leq B$ und $B \leq C$. Sei nun $h(x) = g(f(x))$. h ist total und berechenbar, da f und g total und berechenbar sind. Sei $x \in A$ beliebig. Dann gilt $f(x) \in B$ und $h(x) = g(f(x)) \in C$. Sei nun $x \notin A$ beliebig. Dann gilt $f(x) \notin B$ und damit $h(x) = g(f(x)) \notin C$. Somit ist h eine geeignete Reduktion.

AUFGABE 8.5. (*Reduktionen und Unentscheidbarkeit*) Stufe C

Sei Σ ein beliebiges Alphabet. Zeigen Sie: Es ist unentscheidbar zu prüfen, ob für zwei als Eingabe gegebene Turing-Maschinen die eine auf allen Eingaben genau dann hält, wenn die andere nicht hält. Formaler: Zeigen Sie, die Menge

$$\mathsf{H}_{\mathsf{NEQ}} := \left\{ w_1 \# w_2 \mid w_1, w_2 \in \{0,1\}^* \text{ und } \forall x \in \Sigma^*.\ M_{w_1}[x]\!\downarrow \iff \neg M_{w_2}[x]\!\downarrow \right\}$$

ist unentscheidbar.

Lösungsskizze
Wir reduzieren das Halteproblem auf leerem Band H_0 auf $\mathsf{H}_{\mathsf{NEQ}}$. Dabei gilt zu beachten, dass die Reduktion total, berechenbar und korrekt sein muss.

Reduktion von H_0*:* Sei $w_\perp$ die Kodierung einer Turing-Maschine, die auf keiner Eingabe hält. Sei $w \in \{0,1\}^*$ beliebig. Wir berechnen zunächst die Kodierung w' einer Turing-Maschine, die bei jeder Eingabe das Band löscht und dann $M_w[\varepsilon]$ ausführt. Anschließend geben wir $w' \# w_\perp$ zurück.

Die Reduktion ist total: Für jede Eingabe w wird die Ausgabe $w' \# w_\perp$ erzeugt.

Die Reduktion ist berechenbar: Die En- und Dekodierungsfunktionen für Turing-Maschinen sind berechenbar. Außerdem kann eine TM alle Zeichen auf dem Band, die nicht dem Leerzeichen entsprechen, durch geeignete Übergänge anfangs überschreiben.

Die Reduktion ist korrekt:

$$\begin{aligned}
w \in \mathsf{H}_0 &\iff M_w[\varepsilon]\!\downarrow && \text{(Def. } \mathsf{H}_0\text{)} \\
&\iff \forall x \in \Sigma^*.\ M_{w'}[x]\!\downarrow && (M_{w'} \text{ führt stets } M_w \text{ auf leerem Band aus)} \\
&\iff \forall x \in \Sigma^*. \big(M_{w'}[x]\!\downarrow \text{ und } \neg M_{w_\perp}[x]\!\downarrow\big) && (M_{w_\perp} \text{ hält nie)} \\
&\iff w' \# w_\perp \in \mathsf{H}_{\mathsf{NEQ}} && \text{(Def. } \mathsf{H}_{\mathsf{NEQ}}\text{)}
\end{aligned}$$

□

AUFGABE 8.6. (*Collatz-Vermutung*) Stufe C

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 $i \in \mathbb{N}_0$, sodass $a_i = 1$.

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.

Lösungsskizze
Das folgende Programm P terminiert, falls es für einen Startwert a_0 (in Variable x) in der Collatz-Folge das Folgeglied $a_i = 1$ findet. Beachte, dass für die Eingabe $x = 0$ die Collatz-Folge nicht definiert ist und das Programm sofort mit 0 terminiert (Achtung: modifizierte Differenz). Zur besseren Lesbarkeit verwenden wir $y = x_0$, $x = x_1$, $z = x_2$, $c = x_3$ als Abkürzungen. Berechnet nun das Programm die Nullfunktion, so ist die Collatz-Vermutung korrekt, da für alle Startwerte das Programm hält und 0 zurückgibt. Berechnet das Programm nicht die Nullfunktion, so gibt es mindestens einen Startwert, für den das Programm nicht terminiert.
Programm P:

```
x := x - 1;
WHILE x ≠ 0 DO
  x := x + 1;
  c := 2;
  z := x MOD c;
  IF z = 0 DO
    x := x DIV c
  ELSE
    z := x + x;
    x := x + z;
    x := x + 1
  END;
  x := x - 1
END;
y := 0
```