Übungsblatt 6

Aufgabe 1
Wahr oder falsch? Mit Begründung!

(a) Seien A und B Sprachen über dem gleichen Alphabet. Gelten $A \leq B$ und $B \leq A$, so ist $A = B$.

(b) Sei f eine Reduktion von A auf B. Dann ist f^{-1} eine gültige Reduktion von B auf A.

(c) Falls $A \leq B$ gilt, so auch $\overline{A} \leq \overline{B}$.

(d) Falls $A \leq B$ gilt und A semi-entscheidbar ist, dann ist auch B semi-entscheidbar.

(e) Wenn $A \leq B$ und B semi-entscheidbar ist, dann ist auch A semi-entscheidbar.

(f) Wenn $A \subseteq B$ gilt und B entscheidbar ist, dann ist auch A entscheidbar.

(g) Wenn $A \subseteq B$ gilt und A unentscheidbar ist, dann ist auch B unentscheidbar.

Lösung

(a) Falsch: Sei z.B. $A = \{a\}$ und $B = \{aa\}$. Offenbar ist $A \neq B$. Gültige Reduktionen bekommt man zum Beispiel mit $f, g : \{a\}^* \to \{a\}^*$ und $f(a^i) = a^{i+1}$ bzw. $g(a^{i+1}) = a^i, g(\varepsilon) = \varepsilon$ (für $i \in \mathbb{N}$). Heißt: Die Relation $\leq$ ist nicht antisymmetrisch.

(b) Falsch: Offenbar muss f nicht surjektiv sein. Somit ist das Urbild von manchen Wörtern aus B leer, was aber nicht für das Bild einer gültigen Reduktion gelten darf.

(c) Wahr, denn $w \in A \Leftrightarrow f(w) \in B$ ist gleich $w \notin A \Leftrightarrow f(w) \notin B$, was wiederum gleich $w \in \overline{A} \Leftrightarrow f(w) \in \overline{B}$ ist.

(d) Falsch: Zum Beispiel $A = \{a\}$ und $B = \{w \mid M_w \text{ akzeptiert } w \text{ nicht}\}$. B ist die Diagonalsprache (das Komplement des speziellen Haltproblems) und nicht semi-entscheidbar. Eine Reduktion ist $f(a) = w_0 \in B$ und $f(w) = w_1 \notin B$, $w \neq a$, wobei w_0 und w_1 beliebig gewählt sind. Die Existenz von w_0, w_1 genügt für die Berechenbarkeit von f.

(e) Wahr: Mit der Definition von $\leq$ sieht man leicht, dass $\chi'_A(w) = \chi'_B(f(w))$ für alle $w \in \Sigma^*$ ist. Außerdem sind χ'_B und f berechenbar, also ist χ'_A berechenbar.

(f) Falsch: Gegenbeispiel $B = \Sigma^*$ und A ein beliebiges unentscheidbares Problem.

(g) Falsch, gleiches Gegenbeispiel.

Aufgabe 2
Die Softwarefirma HALTING & CO. KG bietet folgende Produkte zur Programmverifikation an.

(a) Produkt A überprüft, ob ein gegebenes Programm auf allen Eingaben terminiert.

(b) Produkt B überprüft, ob ein gegebenes Programm auf einer gegebenen Eingabe höchstens 1,000 Rechenschritte durchführt.

(c) Produkt C überprüft, ob ein gegebenes Programm auf einer gegebenen Eingabe höchstens 1 GB Speicher benötigt.

(d) Produkt D überprüft, ob ein gegebenes Programm niemals die Ausgabe 123 produziert.

Welche Produkte kann es tatsächlich geben?

Lösung

(a) Falsch. Das spezielle Halteproblem K (Folie 127) kann auf das Problem (a) reduziert werden. Da das spezielle Halteproblem unentscheidbar ist, folgt dass auch dieses Problem unentscheidbar ist. Formal kann das Problem (a) als die Sprache

$$A = \{w \mid M_w \text{ hält für alle Eingaben } x \in \{0,1\}^*\}$$

angesehen werden.

Zu zeigen ist also $K \leq A$. Die Reduktion funktioniert wie folgt: Wir zeigen, dass wenn man prüfen könnte, ob ein gegebenes Programm (eine Turing-Maschine) auf allen Eingaben terminiert, so könnte man auch das spezielle Halteproblem entscheiden (welches prüft ob eine Turing-Maschine M_w auf der eigenen Kodierung w terminiert). Dazu modifizieren wir eine gegebene Turing-Maschine $M_{w'}$ so, dass zu Beginn die eigentliche Eingabe (egal welche) gegen die Kodierung w von M_w ausgetauscht wird, dass heißt die Turing-Maschine überschreibt den initialen Bandinhalt immer durch die entsprechende Kodierung der originalen Turing-Maschine. Anschließend arbeitet die Turing-Maschine $M_{w'}$ genau wie originale Turing-Maschine M_w, d.h. M_w wird auf $M_{w'}$ simuliert. Diese modifizierte Turing-Maschine $M_{w'}$ terminiert nun auf allen Eingaben genau dann, wenn M_w auf der eigenen Kodierung terminiert. Dies liefert uns eine Reduktionsabbildung $f(w) = w'$. Somit könnte man das spezielle Halteproblem mit Hilfe von A entscheiden, was zur Folge hat, dass A unentscheidbar ist.

(b) Wahr. Das Programm wird unter der gegebenen Eingabe simuliert und die Rechenschritte mitgezählt. Falls das simulierte Programm bis zum 1000. Rechenschritt terminiert, meldet der Simulator ≪terminiert≫. Andernfalls wird die Simulation abgebrochen und der Simulator meldet ≪terminiert nicht bei bis zu 1000 Rechenschritten≫ .

(c) Wahr. Das Programm wird unter der gegebenen Eingabe simuliert und alle bereits erreichten Konfigurationen gespeichert. Wichtig ist, dass es mit 1 GB Speicher nur endlich viele Konfigurationen gibt, die in diesem begrenzten Speicher arbeiten. Sollte eine Konfiguration auftauchen, die mehr als 1 GB Speicher benötigt, bricht der Simulator mit «braucht mehr als 1GB Speicher» ab. Sollte das simulierte Programm terminieren ohne jemals eine zu große Konfiguration zu erreichen, meldet der Simulator «das Programm terminiert und benötigt höchstens 1GB Speicher». Der letzte mögliche Fall ist der, dass eine bereits erreichte Konfiguration mit weniger als 1GB erneut erreicht wird (da es nur endlich viele Konfigurationen dieser Art gibt) ohne dass zuvor terminiert wurde oder eine zu große Konfiguration besucht wurde. In diesem Fall würde sich das Programm also in eine Endlosschleife begeben und der Simulator bricht mit «das Programm terminiert nicht» ab.

(d) Falsch. Dies können wir in zwei Schritten zeigen: Zunächst zeigen wir, dass das folgende Problem unentscheidbar ist:

$$P = \{w \mid M_w \text{ hält auf keiner Eingabe } x \in \{0,1\}^*\}$$

Man beachte, es gilt *nicht* $P = \overline{A}$ (sonst bräuchten wir nichts mehr zu zeigen wegen (a) und wegen Abschluss der (Un-)Entscheidbarkeit unter Komplementbildung). Hierzu reduzieren wir das Komplement des speziellen Halteproblems (die Diagonalsprache D) auf P: Wir definieren $f(w) = w'$, wobei die Turingmaschine $M_{w'}$ folgendes tun soll: Der eigentliche Bandinhalt / Input wird überschrieben durch w und anschließend wird M_w simuliert. Wenn M_w auf w nicht terminiert, so terminiert $M_{w'}$ also auf keiner Eingabe. Falls M_w auf w terminiert, so terminiert auch $M_{w'}$. Es gilt also $D \leq P$.

Im zweiten Schritt geben wir eine Reduktion von P auf

$$B = \{w \mid M_w \text{ produziert niemals die Ausgabe } 123\}$$

an. Mit $D \leq P \leq B$ gilt insbesondere $D \leq B$, also ist B unentscheidbar. Die Reduktion funktioniert wie folgt: Wir zeigen, dass wenn man prüfen könnte, ob ein gegebenes Programm niemals die Ausgabe 123 produziert, so könnte man auch prüfen ob ein Programm auf keiner Eingabe terminiert. Dazu modifizieren wir das gegebene Programm so, dass wann immer das Programm terminiert (also im Falle einer Turing-Maschine in einen Endzustand übergeht), so wird vorher noch die Ausgabe 123 produziert. Dieses modifizierte Programm produziert nun niemals die Ausgabe 123 genau dann, wenn das ursprüngliche Programm auf keiner Eingabe terminiert (da im Umkehrschluss die Ausgabe 123 immer produziert würde, wenn das Programm terminieren würde). Somit könnte man mit Hilfe von B ein unentscheidbares Problem entscheiden, was bedeutet, dass auch B unentscheidbar sein muss.

Formal definieren wir die Reduktion f durch $f(w) = w'$, wenn $M_{w'}$ jedes mal, wenn M_w terminiert, noch die Ausgabe 123 produziert. Mit obiger Begründung ist $w \in P$ genau dann, wenn $f(w) = w' \in B$ ist.

Aufgabe 3
Zeigen Sie $A \leq \overline{B}$ für

- $A = \{w \mid M_w$ hält nicht auf w oder M_w gibt auf w 0 aus$\}$ und
- $B = \{w \mid M_w$ hält auf $w\}$.

Lösung
Wir konstruieren eine Turingmaschine $M_{w'}$ wie folgt: $M_{w'}$ ignoriert ihren Input, simuliert M_w auf w und speichert den Output von M_w in v. Falls $v = 0$, geht $M_{w'}$ in eine Endlosschleife. Andernfalls gibt $M_{w'}$ 1 aus. Wir definieren nun $f(w) = w'$. Offensichtlich ist f total und berechenbar. Es bleibt zu zeigen, dass f tatsächlich die gewünschte Reduktionsfunktion darstellt.
Sei $w \in A$. Dann gerät $M_{w'}$ in eine Endlosschleife (entweder weil die Simulation nicht hält oder weil M_w auf w 0 ausgegeben hat und $M_{w'}$ in eine Endlosschleife geht). Somit hält $M_{w'}$ insbesondere für Input w nicht. Also ist $f(w) = w' \in \overline{B}$.
Sei $w \notin A$. Dann hält M_w auf w und gibt etwas Anderes als 0 aus. Somit gilt niemals $v = 0$ und daher gibt $M_{w'}$ immer 1 aus. Insbesondere hält $M_{w'}$ also immer und somit auch auf dem Input w. Daher gilt $f(w) = w' \notin \overline{B}$.

Aufgabe 4
Entscheidbar oder nicht? Mit Beweis!

(a) $P_1 = \{w \mid M_w$ hält nicht bei leerer Eingabe$\}$

(b) $P_2 = \{w\#x \mid$ In der Rechnung von M_w bei Eingabe x gibt es außer der Startkonfiguration eine weitere, bei der der Leseschreibkopf auf $\Box$ steht.$\}$

Lösung

(a) Unentscheidbar: Das Komplement $\overline{P_1}$ ist genau das Halteproblem auf leerem Band H_0. Wir wissen, dass H_0 unentscheidbar ist (Satz 23, Folie 140) und dass entscheidbare Sprachen unter Komplementbildung abgeschlossen sind (Blatt 5, Aufgabe 1a), also ist P_1 ebenfalls nicht entscheidbar.

(b) Entscheidbar: Die Zusatzforderung „außer der Startkonfiguration“ bei unserer Definition von Turingmaschinen ist nur für das leere Wort relevant. Ansonsten fragen wir uns nur, ob wir ein Blank sehen oder nicht. Die Überlegung ist ähnlich wie bei der Aufgabe 2c! Sei $z = |Z|$ die Anzahl der Zustände von M_w, $s = |\Gamma| - 1$ und $n = |x|$. Falls die Turingmaschine nie auf ein Blank trifft, gibt es höchstens $m = nzs^n$ viele Konfigurationen, weil sich der Leseschreibkopf innerhalb des Wortes x bewegen muss (n Positionen für z Zustände und s^n viele Wörter der Länge n ohne Blank). Also haben wir nach $m+1$ Schritten entweder eine Schleife festgestellt, ein Blank erreicht oder wir terminieren in weniger als $m + 1$ Schritten. Dies lässt sich mit einer Turingmaschine entscheiden, die M_w simuliert und mitzählt.