Theoretische Infomatik: Übungsaufgaben - Blatt 5

Abgabe bis 31. Oktober 2014

Prof. Hromkovič

Kevin Klein, David Bimmler und Vincent von Rotz

Aufgabe 14

Wir beweisen $L_1 \notin \mathcal{L}_{RE}$ indirekt mit einem Widerspruchsbeweis. Dann ist $L_1 = L(M)$ für eine TM M. Weil M eine der Turingmaschinen in der Nummerierung aller Turingmaschinen sein muss, existiert ein $i \in \mathbb{N} - \{0\}$, so dass $M = M_i$. Aber L_1 kann nicht gleich $L(M_i)$ sein, weil folgende Äquivalenz gilt:

$$w_{i^3+5} \in L_1 \iff d_{i,i^3+5} = 0 \iff w_{i^3+5} \notin L(M_i),$$

d.h., w_{i^3+5} ist in genau einer der Sprachen L_1 oder $L(M_i)$.

Dieselbe Methode funktioniert nicht für L_2, da wir verwendet haben, dass L_1 von irgendeiner TM erzeugt werden muss, falls $L_1 \in \mathcal{L}_{RE}$. D.h., wir haben alle TM M betrachtet, was die Definition von L_2 nicht ermöglicht.

Aufgabe 15

a) Wir wissen (aus einem Lemma im Skript), dass $L_U \in \mathcal{L}_{RE}$. Aus Teilaufgauben b) und c) lernen wir, dass $L_U \leq_{EE} L_{union}$ und $L_{union} \leq_{EE} L_U$. Es liegt somit auf der Hand, dass auch $L_{union} \in \mathcal{L}_{RE}$.

b) Wir beweisen $L_U \leq_{EE} L_{union}$ im Formalismus der Turingmaschinen. Wir beschreiben eine TM M, die L_U auf L_{union} reduziert. Für die Eingabe x arbeitet M wie folgt:

   M überprüft, ob die Eingabe die Form $x = \text{Kod}(M_1)\#w$ für eine TM M_1 und ein $w \in (\Sigma_{bool})^*$ hat.

   i) Falls x nicht besagte Form hat, so wird die Kodierung $\text{Kod}(M_\emptyset)$ einer TM $M_\emptyset$ generiert, die keine Eingabe akzeptiert, sondern immer direkt ablehnt. Dann wird noch das Symbol # angefügt, die $\text{Kod}(M_\emptyset)$ wiederholt und nochmal ein # angehängt. Dann hält M mit dem Bandinhalt $M(x) = \text{Kod}(M_\emptyset)\#\text{Kod}(M_\emptyset)\#$.

   ii) Falls $x = \text{Kod}(M_1)\#w$, dann modifiziert M die Eingabe folgendermassen: Es wird vor die Kodierung der Touringmaschine die Kodierung $\text{Kod}(M_\emptyset)$ einer zweiten Turingmaschine eingefügt. Diese Turingmaschine akzeptiert wiederum keine Eingabe und lehnt sofort ab. M beendet seine Arbeit mit dem Bandinhalt $M(w) = \text{Kod}(M_\emptyset)\#\text{Kod}(M_1)\#w$.

   Nun beweisen wir für alle $x \in \{0, 1, \#\}^*$, dass

$$x \in L_U \iff M(x) \in L_{union}$$

   Sei $x \in L_U$. Daher ist $x = \text{Kod}(M_1)\#w$ für eine TM M_1 und ein Wort $w \in \{0,1\}^*$. Dann ist $M(x) = \text{Kod}(M_\emptyset)\#\text{Kod}(M_1)\#w$.Wenn L_U nun x akzeptiert, bedeutet das, dass $w \in L(M_1)$. Somit gilt auch $w \in L(M_1) \cup L(M_\emptyset)$ und also $x \in L_{union}$.

   Sei $x \notin L_U$. Das bedeutet entweder, dass es nicht von der Form $x = \text{Kod}(M_1)\#w$ ist, oder dass $M_1 x$ nicht akzeptiert. Im ersten Fall gilt $M(x) = \text{Kod}(M_\emptyset)\#\text{Kod}(M_\emptyset)\#$. Da $\lambda \notin L(M_\emptyset) \cup L(M_\emptyset)$ ist $x \notin L_{union}$. Ansonsten ist $M(x) = \text{Kod}(M_\emptyset)\#\text{Kod}(M_1)\#w$. Es gilt:

   $x \notin L_U \Rightarrow w \notin L(M_1) \Rightarrow w \notin L(M_\emptyset) \cup L(M_1) \Rightarrow x \notin L_{union}$

c) Wir beschreiben wiederum eine TM B. B arbeitet für die Eingabe x wie folgt:

Ist x nicht von der Form $Kod(M)\#Kod(M')\#w$, wobei M, M' Turingmaschinen sind und $w \in (\Sigma_{bool})^*$, so gibt B λ zurück.

Ansonsten kreieren wir eine NTM (die ja gleichwertig ist zu einer TM) M_U, die zu Beginn nichtdeterministisch entscheidet, ob sie ihre Eingabe mit M oder mit M' überprüfen soll. Wenn einer der beiden Berechnungspfade in einem akzeptierten Zustand landet, so soll M_U akzeptieren. Die jeweiligen Verwerfzustände von M und M' werden aufgehoben. Somit terminiert M_U akzeptierend, wenn die Eingabe $\in L(M) \cup L(M')$ ist, und terminiert sonst nicht. B gibt die Kodierung dieser Turingmaschine zurück: $B(x) = Kod(M_U)$

Nun beweisen wir für alle $x \in \{0, 1, \#\}^*$, dass

$$x \in L_{union} \iff B(x) \in L_U$$

Sei $x \in L_{union}$. Dies bedeutet, dass $x = Kod(M)\#Kod(M')\#w$ und $w \in L(M)$ oder $w \in L(M')$. Folglich gilt auch $w \in L(M) \cup L(M')$ und also auch $B(x) \in L_U$.

Sei $x \notin L_{union}$. Dann gilt entweder $x \neq Kod(M)\#Kod(M')\#w$ oder $w \notin L(M) \wedge w \notin L(M')$. Im ersten Fall ist $B(x) = \lambda \notin L_U$. Ansonsten gilt $B(x) = M_U\#w \notin L_U$, da M_U mit der Eingabe w niemals terminiert.

Aufgabe 16

a) Wir zeigen, dass $L_H \leq_{EE} L_U$, woraus $L_H \leq_R L_U$ folgt. Wir konstruieren also eine Turingmaschine A, die die Eingabe.

Falls x nicht von der Form $Kod(M_3)\#w$ ist, so gibt A λ aus.

Falls x aber diese Form hat, so kreieren wir eine neue Turingmaschine M_3', die zu M_3 identisch ist bis auf eine Änderung: wir passen δ an, indem wir jegliche Transition der Form $\delta(q, a) = q_{reject}$ zu $\delta'(q, a) = q_{accept}$ ändern. A gibt dann $Kod(M_3')\#w$ aus.

Nun beweisen wir für alle $x \in \{0, 1, \#\}^*$, dass

$$x \in L_H \iff A(x) \in L_U$$

Sei $x \in L_H$. Somit hat x die Form $x = Kod(M_3)\#w$. Nun ist $A(x) = Kod(M_3')\#w$. Wir wissen, dass M_3 mit w als Eingabe hält. Zudem wissen wir, dass in M_3' q_{reject} durch q_{accept} ersetzt wurde. Dies bedeutet, dass die M_3' akzeptiert, solange sie hält. Somit ist klar, dass M_3' w akzeptiert, und somit gilt $A(x) \in L_U$.

Sei $x \notin L_H$. Falls x nicht von der angegebenen Form ist, so ist $A(x) = \lambda$. $\lambda \notin L_U$ ist offensichtlich. Ansonsten ist $A(x) = Kod(M_3')\#w$. Falls $x \notin L_H$ bedeutet das, dass die L_H mit der Eingabe x nicht hält. Somit hält auch $A(X)$ mit derselben Eingabe nicht und $A(x) \notin L_U$.

b)