Übungsblatt04 - Semientscheidbarkeit - Musterlösung

TH Mittelhessen, FB MNI, Berechenbarkeit und Komplexität, Prof. Dr. B. Just

Aufgabe 1

Bitte beweisen Sie die folgenden Aussagen, ohne währenddessen noch einmal in Ihre Mitschrift zu schauen :):

a.) $\text{HALT}_{\text{TM}} = \{(M, \omega) : \text{Turingmaschine } M \text{ hält bei Input } \omega\}$ ist aufzählbar (d.h., semi-entscheibar).

b.) $\text{A}_{\text{TM}} = \{(M, \omega) : \text{Turingmaschine } M \text{ akzeptiert Input } \omega\}$ ist aufzählbar (d.h., semi-entscheibar).

c.) $\text{NotEmpty}_{\text{TM}} = \{M : \text{Turingmaschine } M \text{ akzeptiert mindestens einen Input }\}$ ist aufzählbar (d.h., semi-entscheibar).

d.) $\overline{\text{HALT}}_{\text{TM}} = \{(M, \omega) : \text{Turingmaschine } M \text{ hält nicht bei Input } \omega\}$ ist nicht aufzählbar.

e.) $\overline{\text{A}}_{\text{TM}} = \{(M, \omega) : \text{Turingmaschine } M \text{ akzeptiert nicht den Input } \omega\}$ ist nicht aufzählbar.

f.) $\text{Empty}_{\text{TM}} = \{M : \text{Turingmaschine } M \text{ akzeptiert keinen Input }\}$ ist nicht aufzählbar.

Lösung Aufgabe 1:

a.) Zu zeigen ist: Es gibt eine Turingmaschine H mit $L(H) = \text{HALT}_{\text{TM}}$.
D.h., H muss für jeden Input (M, ω) einer TM M und eines Inputs ω für diese TM erfüllen:

$$H(M, \omega) = \text{accept}, \quad \text{falls } M \text{ bei Input } \omega \text{ anhält}, \quad \text{und}$$
$$H(M, \omega) \in \{\text{reject, loop}\}, \quad \text{falls } M \text{ bei Input } \omega \text{ nicht anhält.}$$

Die Turingmaschine H, die bei Input (M, ω) die Berechnung von M auf ω simuliert, und akzeptiert, sobald M anhält, erfüllt das.

b.) Zu zeigen ist: Es gibt eine Turingmaschine A mit $L(A) = \text{A}_{\text{TM}}$.
D.h., A muss für jeden Input (M, ω) einer TM M und eines Inputs ω für diese TM erfüllen:

$$A(M, \omega) = \text{accept}, \quad \text{falls } M \text{ bei Input } \omega \text{ akzeptiert}, \quad \text{und}$$
$$A(M, \omega) \in \{\text{reject, loop}\}, \quad \text{falls } M \text{ bei Input } \omega \text{ nicht akzeptiert.}$$

Die Turingmaschine A, die bei Input (M, ω) die Berechnung von M auf ω simuliert, und akzeptiert, sobald M akzeptiert, erfüllt

$$L(A) = \text{A}_{\text{TM}}.$$

c.) Zu zeigen ist: Es gibt eine Turingmaschine X, die bei Input einer Turingmaschine M erfüllt:

$$X(M) = \text{accept}, \quad \text{falls } M \text{ irgendeinen Input akzeptiert,} \quad \text{und}$$
$$X(M) \in \{\text{reject, loop}\}, \quad \text{falls } M \text{ keinen Input akzeptiert.}$$

X sei die Turingmaschine, die wie folgt arbeitet:

```
for i=1, 2, 3, .... do
    generiere alle Inputs der Länge i
    Für jeden Input ω aus diesen Inputs do
      führe die ersten i Berechnungsschritte von M(ω) aus
      falls M dabei den akzeptierenden Zustand erreicht, akzeptiere
    endfor
 endfor.
```

Jede endliche Berechnung auf jedem Input von M wird von X bei Eingabe einer TM M simuliert. D.h., wenn es einen Input ω gibt, bei dem die Berechnung $M(\omega)$ akzeptiert, so akzeptiert auch $X(M)$. Andernfalls akzeptiert $X(M)$ nicht, da $X(M)$ für keinen Input ω und keine endliche Berechnungslänge in den akzeptierenden Zustand geht. Somit ist $L(X) = \text{NotEmpty}_{\text{TM}}$, qed.

Vorbemerkung zu d.), e.) und f.): Alle drei Beweise sind in ihrer Struktur gleich. Die Sprachen sind die Komplemente semi-entscheidbarer Mengen. Wären sie ebenfalls semi-entscheidbar, so wären die Sprachen entscheidbar (nach dem in der Vorlesung bewiesenen Satz, dass eine Sprache genau dann entscheidbar ist, wenn sie selbst und ihr Komplement semi-entscheidbar sind).

d.)

- Das Komplement der gegebenen Menge, also die Menge

$$\text{HALT}_{\text{TM}} = \{(M, \omega) : \text{Turingmaschine } M \text{ hält bei Input } \omega\},$$

ist aufzählbar nach Aufgabe 1.a.

- Weiter hatten wir in der Vorlesung bewiesen, dass eine Sprache entscheidbar ist, genau dann, wenn sie selbst und ihr Komplement aufzählbar sind.

- Wäre also $\overline{\text{HALT}_{\text{TM}}}$ aufzählbar, so wäre A_{TM} entscheidbar. Dies ist jedoch nicht der Fall, also kann $\overline{\text{HALT}_{\text{TM}}}$ nicht aufzählbar sein.

e.)

- Das Komplement der gegebenen Menge, also die Menge

$$\text{A}_{\text{TM}} = \{(M, \omega) : \text{Turingmaschine } M \text{ akzeptiert Input } \omega\},$$

ist aufzählbar nach Aufgabe 1.b.

- Weiter hatten wir in der Vorlesung bewiesen, dass eine Sprache entscheidbar ist, genau dann, wenn sie selbst und ihr Komplement aufzählbar sind.

- Wäre also $\overline{\text{A}_{\text{TM}}}$ aufzählbar, so wäre A_{TM} entscheidbar. Dies ist jedoch nicht der Fall, also kann $\overline{\text{A}_{\text{TM}}}$ nicht aufzählbar sein.

f.)

- Das Komplement der gegebenen Menge, also die Menge

$$\text{NotEmpty}_{\text{TM}} = \{M : \text{Turingmaschine } M \text{ akzeptiert mindestens einen Input }\}$$

ist aufzählbar nach Aufgabe 1.c.

- Weiter hatten wir in der Vorlesung bewiesen, dass eine Sprache entscheidbar ist, genau dann, wenn sie selbst und ihr Komplement aufzählbar sind.

- Wäre also Empty_{TM} aufzählbar, so wäre Empty_{TM} entscheidbar. In einem früheren Übungsblatt wurde aber gezeigt, dass dies jedoch nicht der Fall ist, also kann Empty_{TM} nicht aufzählbar sein.

Aufgabe 2

Bitte beweisen Sie, dass die folgende Menge nicht aufzählbar und auch nicht co-aufzählbar ist:

$$\text{EQ}_{\text{TM}} = \{(M_1, M_2) : M_1, M_2 \text{ sind Turingmaschinen mit } L(M_1) = L(M_2)\}$$

Hinweis: Anspruchsvoll, aber spannend :). Beides wird mit Widerspruchsbeweis gemacht.
Man zeigt mit den Bezeichnungen aus Aufgabe 1: Wäre EQ_{TM} aufzählbar (also, semi-entscheidbar), so wäre auch Empty_{TM} aufzählbar, im Widerspruch zu Aufgabe 1. Und man zeigt: Wäre EQ_{TM} co-aufzählbar, so wäre $\overline{\text{A}}_{\text{TM}}$ aufzählbar, ebenfalls im Widerspruch zu Aufgabe 1.

Lösung Aufgabe 2:

Es werden folgende beiden Aussagen gezeigt:

a.) $\text{EQ}_{\text{TM}} = \{(M_1, M_2) : M_1, M_2 \text{ sind Turingmaschinen mit } L(M_1) = L(M_2)\}$
ist nicht semi-entscheidbar, und

b.) $\overline{\text{EQ}_{\text{TM}}} = \{(M_1, M_2) : M_1, M_2 \text{ sind Turingmaschinen mit } L(M_1) = L(M_2)\}$
ist nicht semi-entscheidbar.

Die erste Aussage besagt, dass EQ_{TM} nicht aufzählbar ist. Die zweite besagt, dass EQ_{TM} auch nicht co-aufzählbar ist.

<u>Beweis von a.):</u>

Angenommen, es gäbe eine Turingmaschine X mit $L(X) = \text{EQ}_{\text{TM}}$.
D.h., für jedes beliebig Paar (M_1, M_2) von Turingmaschinen ist

$$X(M_1, M_2) = \text{accept}, \quad \text{falls } L(M_1) = L(M_2), \quad \text{und}$$
$$X(M_1, M_2) \in \{\text{reject, loop}\}, \quad \text{falls } L(M_1) \neq L(M_2).$$

Sei M_E („E" für „empty") eine Turingmaschine, die jeden Input verwirft. Eine solche kann man leicht konstruieren. Und sei M eine beliebige Turingmaschine. Dann ist

$$X(M_E, M) = \text{accept}, \quad \text{falls } L(M) = \{\}, \quad \text{und}$$
$$X(M_E, M) \in \{\text{reject, loop}\}, \quad \text{falls } L(M) \neq \{\}.$$

Sei Y die Turingmaschine, die bei Input einer beliebigen Turingmaschine M die Turingmaschine X anwendet auf M_E und M, also

$$Y(M) = X(M_E, M).$$

Dann ist $L(Y) = \text{Empty}_{\text{TM}}$.

Somit ist Empty_{TM} semi-entscheidbar, ein Widerspruch zu Aufgabe 1f. Also gibt es die TM X nicht, und EQ_{TM} ist nicht semi-entscheidbar.

<u>Beweis von b.):</u>

Angenommen, es gäbe eine Turingmaschine U mit $L(U) = \overline{\text{EQ}_{\text{TM}}}$.

Wir konstruieren aus U eine Turingmaschine V, die $\overline{\text{A}_{\text{TM}}}$ erkennt (d.h., akzeptiert, wenn der Input in $\overline{\text{A}_{\text{TM}}}$ ist, und ansonsten verwirft oder loopt). Das ist dann ein Widerspruch zu Aufgabe 1 b.

Bei Input (M, ω) arbeitet V wie folgt:

i. Kontruiere M_ω, die Turingmaschine, die bei allen Inputs außer ω verwirft, und bei Input ω die Turingmaschine M auf ω anwendet. Es ist

$$L(M_\omega) = L(M) \cap \{\omega\}.$$

Man sieht:

$$L(M_\omega) = \begin{cases} \{\omega\}, & \text{falls } M(\omega) = \text{accept} \\ \{\}, & \text{falls } M(\omega) \in \{\text{reject, loop}\}. \end{cases}$$

ii. Konstruiere NUR_ω, die Turingmaschine, die den Input ω akzeptiert und alle anderen Inputs verwirft. Es ist

$$L(\text{NUR}_\omega) = \{\omega\}.$$

iii. Wende U auf M_ω und NUR_ω an. Es ist also

$$V(M, \omega) = U(M_\omega, \text{NUR}_\omega).$$

Man überzeugt sich:

Falls M bei Input ω akzeptiert, ist $\{\omega\} = L(M_\omega) = L(\text{NUR}_\omega)$, also ist $U(M_\omega, \text{NUR}_\omega) \neq \text{acc}$, also auch $V(M, \omega) \neq \text{acc}$.

Falls M bei Input ω nicht akzeptiert (also verwirft oder loopt), ist $\{\omega\} = L(\text{NUR}_\omega) \neq L(M_\omega) = \{\}$, also ist $U(M_\omega, \text{NUR}_\omega) = \text{acc}$, also auch $V(M, \omega) = \text{acc}$.

Somit ist tatsächlich $L(V) = \overline{\text{A}_{\text{TM}}}$, was nicht sein kann, denn $\overline{\text{A}_{\text{TM}}}$ ist nicht semi-entscheidbar. Somit kann es die TM U nicht gegeben haben, also ist EQ_{TM} nicht co-aufzählbar. Q.e.d.

Aufgabe 3

Bitte entscheiden Sie, ob die folgenden Sprachen entscheidbar, aufzählbar, co-aufzählbar oder weder aufzählbar noch co-aufzählbar sind. Es geht dabei vor allem darum, ein Gefühl für die Sprachen zu bekommen, formale Beweise sind nicht erforderlich.

a.) $\{p \in \mathbb{N} : p \text{ ist prim}\}$.

b.) $\{G = (V, E) : G \text{ ist zusammenhängender Graph}\}$.

c.) $\{G = (V, E) : G \text{ ist nicht zusammenhängender Graph}\}$

d.) Die Menge aller Turingmaschinen, die das Wort „Katze“ akzeptieren.

d.) Die Menge aller Turingmaschinen, die nur das Wort „Katze“ akzeptieren.

e.) Die Menge aller Turingmaschinen, die gar nichts akzeptieren.

f.) Die Menge aller Turingmaschinen, die jedes Wort akzeptieren.

g.) Die Menge aller C++ Programme, die stets anhalten.

h.) Die Menge aller C++ Programme, die syntaktisch korrekt sind.

Lösung Aufgabe 3:

a.) $\{p \in \mathbb{N} : p \text{ ist prim}\}$.

Ist entscheidbar. Der Algorithmus (die TM), der für i=1 bis n die Teilbarkeit prüft, und akzeptiert, wenn kein Rest bleibt, und ansonsten verwirft, entscheidet die Sprache.

b.) $\{G = (V, E) : G \text{ ist zusammenhängender Graph}\}$.

Ist entscheidbar. Mithilfe eines breath-first-search-Algorithmus kann festgestellt werden, ob ein Graph eine oder mehrere Zusammenhangskomponenten hat.

c.) $\{G = (V, E) : G \text{ ist nicht zusammenhängender Graph}\}$.

Ist entscheidbar mit dem Algorithmus aus c. (Man sieht: Komplemente entscheidbarer Sprachen sind ebenfalls entscheidbar).

d.) Die Menge aller Turingmaschinen, die das Wort „Katze“ akzeptieren.

Ist aufzählbar. Die Turingmaschine K, die die Sprache semi-entscheidet, arbeitet wie folgt:
Bei Input einer Turingmaschine M simuliert K die Berechnung von M auf dem Input „Katze“. K akzeptiert, falls die Berechnung akzeptierend endet, verwirft, falls die Berechnung verwerfend endet, und loopt ansonsten wie die Berechnung selbst.
$L(K)$ ist das Wort „Katze“, wenn M das Wort akzeptiert, und andernfalls die leere Menge.

d.) Die Menge aller Turingmaschinen, die nur das Wort „Katze“ akzeptieren.

Ist weder aufzählbar noch co-aufzählbar.

(Beweis läuft ähnlich aufwändig wie Aufgabe 2, entfällt hier) e.) Die Menge aller Turingmaschinen, die gar nichts akzeptieren.

co-aufzählbar, siehe Aufgabe 1f.

f.) Die Menge aller Turingmaschinen, die jedes Wort akzeptieren.

Ist weder aufzählbar noch co-aufzählbar (Beweis wieder aufwändig, entfällt hier).

g.) Die Menge aller C++ Programme, die stets anhalten.

Ist weder aufzählbar noch co-aufzählbar (Beweis schon wieder aufwändig, entfällt hier auch).

h.) Die Menge aller C++ Programme, die syntaktisch korrekt sind.

Ist entscheidbar (durch einen Parser).

Aufgabe 4

Bitte überlegen Sie sich selbst Sprachen, die entscheidbar, aufzählbar, co-aufzählbar oder weder aufzählbar noch co-aufzählbar sind. Wieder geht es dabei darum, ein Gefühl für die Sprachen zu bekommen, formale Beweise sind nicht erforderlich.

Lösung Aufgabe 4:

Hier gibt es keine Musterlösung ;).