FGI-1 – Formale Grundlagen der Informatik I

Logik, Automaten und Formale Sprachen

Musterlösung 12 : Entscheidbarkeit

Präsenzaufgabe 12.1:

1. Erläutern Sie den Unterschied zwischen Entscheidbarkeit und Aufzählbarkeit!

   Lösung: Eine Sprache L heißt aufzählbar, wenn es eine NTM A gibt, die L akzeptiert: $L = L(A)$. (Def. 18.10)

   Eine Sprache L heißt entscheidbar, wenn es eine NTM A gibt, die L akzeptiert und außerdem für jede Eingabe terminiert. (Def. 18.9)

2. Erläutern Sie den Unterschied zwischen Auf- und Abzählbarkeit!

   Lösung: Eine nicht-leere Sprache $L \subseteq \Sigma^*$ heißt abzählbar, wenn es eine totale Abbildung $f : \mathbb{N} \to \Sigma^*$ gibt, so dass $f(\mathbb{N}) = L$ gilt. (Def. 18.11 bzw. 18.1)

   L heißt aufzählbar, wenn f sogar von einer TM berechnet werden kann. (Def. 18.11)

3. Nennen Sie jeweils ein Beispiel für eine entscheidbare, nicht entscheidbare, aufzählbare und nicht aufzählbare Sprache. Ordnen Sie Ihre Sprachen auch unter dem jeweils anderen Begriff ein.

   Lösung:

   - entscheidbar, aufzählbar: Σ^*
   - nicht entscheidbar, aufzählbar: H (Halteproblem)
   - nicht entscheidbar, nicht aufzählbar: L_d (Diagonalsprache)
   - entscheidbar, nicht aufzählbar: nicht möglich

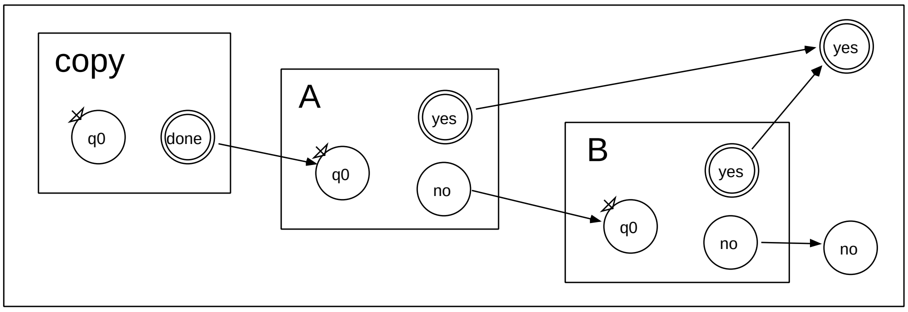
Präsenzaufgabe 12.2:

1. Die Familie der aufzählbaren Sprachen: $\mathcal{R}e$ ist in Bezug auf Vereinigung abgeschlossen.

   Alice hat folgende Konstruktionsskizze angegeben, um dies nachzuweisen. Bob meint: „Ich ahne, was Du meinst. Das klappt aber so nicht“.

   (a) Erläutern Sie die Konstruktionsskizze.

   (b) Was könnte Bob meinen?



Lösung: Sei $L_1, L_2 \in \mathcal{R}e$, dann existiert eine NTM A mit $L(A) = L_1$ und eine NTM B mit $L(B) = L_2$. Es ist $L_1 \cup L_2 \in \mathcal{R}e$ zu zeigen.

(a) Alice wollte eine 2-band NTM M konstruieren (oder eine NTM mit 2 Spuren, das ginge auch). Zuerst kopiert M die Eingabe w auf ihr 2. Band. Danach fahren beide Köpfe an den Anfang zurück. Dann führen wir zunächst die Rechnung der TM A auf dem 1. Band aus aus. Der 2. LSK wird dabei nicht bewegt. Wenn A den *yes*-Endzustand erreicht (d.h. die Maschine A akzeptiert die Eingabe w, d.h. $w \in L(A)$), dann macht M direkt einen Übergang in seinen *yes*-Endzustand.

Wenn A dagegen den *no*-Endzustand erreicht, dann macht M einen Übergang zum Startzustand von B und beginnt die Rechnung der TM B auf dem 2. Band. Wenn auch B den *yes*-Endzustand erreicht, dann auch M, sonst erreicht M seinen *no*-Endzustand.

(b) Problem: Alice hat übersehen, dass die Termination nicht garantiert ist. Wenn die TM A nicht terminiert, dann wird die TM B niemals gestartet werden, so dass man nie feststellen würde, ob/dass B die Eingabe akzeptiert. Also würde der Fall $w \in L(B)$ nie berücksichtigt.

2. Ist Alices Konstruktion, die zeigen sollte, dass $\mathcal{R}e$ bzgl. Vereinigung abgeschlossen ist, korrekt für die Familie der entscheidbaren Sprachen $\mathcal{R}ec$?

Lösung: Für entscheidbare Mengen gibt es immer mindestens eine akzeptierende TM, die garantiert terminiert (d.h. auch auf nicht akzeptierten Wörtern hält). Da die konstruierte TM genau dann akzeptiert, wenn mindestens eine der beiden Ausgangsmaschinen akzeptiert, ist die Konstruktion korrekt.

Präsenzaufgabe 12.3: Wir definieren die Sprache

$$L := \{\langle M, w, n\rangle \mid \text{ es gibt für die NTM } M \text{ mindestens } n \text{ Erfolgsrechnungen auf dem Wort } w\ \}.$$

Zeigen Sie, dass die Sprache L nicht entscheidbar ist, indem sie ein unentscheidbares Problem auf L reduzieren.

Lösung: Zur Auswahl stehen zwei in der Vorlesung präsentierte unentscheidbare Probleme: H und $\overline{L_d}$.

Variante H: Um H auf L zu reduzieren, ist zu zeigen, dass jede Eingabe v zum Problem H so umkodiert werden kann, dass die umkodierte Eingabe $r(v)$ in L zum selben Ergebnis führt wie v in H:

$$v \in H \text{ gdw. } r(v) \in L$$

Folgende Funktion r leistet dies (M_H ist dabei die Kodierung einer beliebigen, festen Turingmaschine, welche H akzeptiert – eine solche TM muss existieren, da H aufzählbar ist):

$$\begin{aligned} r(\langle M, w\rangle) &:= \langle M_H, \langle M, w\rangle, 1\rangle && \text{falls } \langle M, w\rangle \text{ eine wohlgeformte Eingabe ist} \\ r(v) &:= \varepsilon && \text{sonst} \end{aligned}$$

Es gilt $\langle M, w\rangle \in H \implies \langle M_H, \langle M, w\rangle, 1\rangle \in L$, weil zu der Eingabe $\langle M, w\rangle \in H$ mindestens eine Erfolgsrechnung in M_H möglich sein muss (Definition von M_H).

Es gilt $\langle M, w\rangle \notin H \implies \langle M_H, \langle M, w\rangle, 1\rangle \notin L$, weil zu der Eingabe $\langle M, w\rangle \notin H$ keine Erfolgsrechnung in M_H möglich sein kann (sonst wäre doch $\langle M, w\rangle \in H$).

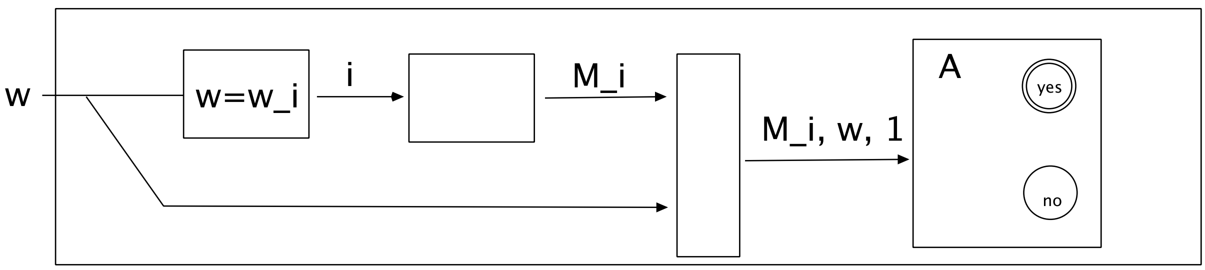
Für nicht wohlgeformte Eingaben v gilt offensichtlich $v \notin H$. Passend dazu gilt auch $\varepsilon \notin L$, da ε keine wohlgeformte Eingabe für L ist.

Variante $\overline{L_d}$: Die Eingabeworte w sind aufzählbar: Es ist möglich, zu jedem Wort w den Index i zu berechnen, für den $w = w_i$ gilt. Außerdem sind die Kodierungen von TM aufzählbar: Wir können zu jedem Index i die TM M_i berechenen.

Es gilt:

$$\begin{aligned} & w_i \in \overline{L_d} \\ \iff\ & w_i \in L(M_i) \\ \iff\ & \text{es gibt in } M_i \text{ mindestens eine Erfolgsrechnung für } w_i \\ \iff\ & \langle M_i, w_i, 1\rangle \in L \end{aligned}$$

Die in der Abbildung skizzierte Reduktion setzt genau dies um.



von
6

Übungsaufgabe 12.4: Seien $L_1, \dots, L_k$ Sprachen über dem Alphabet Σ mit den folgenden Eigenschaften:

1. Die Sprachen sind disjunkt: Für alle $i \neq j$ gilt $L_i \cap L_j = \emptyset$.
2. $L_1 \cup \dots \cup L_k = \Sigma^*$, d.h. jedes Wort ist in einer der Sprachen.
3. Jede Sprache $L_i, i = 1..k$ ist aufzählbar.

Zeige, dass jede Sprache $L_i, i = 1..k$ entscheidbar ist.

Hinweis: Es muss nicht das Zustandsdiagramm einer L_i entscheidenden TM angegeben werden. Es reicht, die Arbeitsweise der entscheidenden TM zu beschreiben.

Lösung: Da jede Sprache L_i aufzählbar ist, existieren TM A_i, die diese Sprachen akzeptieren.

Wir können zu jedem $i = 1..k$ eine TM B_i konstruieren, die folgenden Algorithmus ausführt:

```
n = 0;
b = true;
while b do
    for j = 1 to k do
        if A_j accepts w in exactly n steps
        then index = j; b = false endif
    n = n + 1
if index = i then accept else reject
```

Da jedes Wort w in einer der Sprachen liegt, muss es eine Anzahl von Schritten n und einen Index j geben, so dass A_j die Eingabe w nach genau n Schritten akzeptiert, denn ansonsten würde w ja von keiner der TM A_i akzeptiert.

Da alle Sprachen disjunkt sind, kann es nur genau einen Index j geben. Genau auf diesen Wert j wird dann die Variable $index$ gesetzt und die Schleife wird verlassen. Termination ist also gewährleistet.

Das Verfahren ist auch korrekt: Ist $index$ gleich dem Index i der zu konstruierenden TM B_i (gilt also $w \in L(A_i)$), dann und nur dann wird akzeptiert. Das Wort w wird also genau dann von B_i akzeptiert, wenn $w \in L(A_i)$ gilt.

Alternativbeweis: Sei i ein beliebiger Index. Da jedes Wort in genau einer der Sprachen ist, gilt für das Komplement $\bar{L}_i$:

$$\bar{L}_i = L_1 \cup \dots \cup L_{i-1} \cup L_{i+1} \dots \cup L_k$$

Also ist das Komplement $\bar{L}_i$ eine endliche Vereinigung aufzählbarer Mengen. Da die aufzählbaren Mengen unter Vereinigung abgeschlossen sind (warum?), ist daher auch das Komplement $\bar{L}_i$ eine aufzählbare Menge.

Also ist sowohl L_i als auch das Komplement $\bar{L}_i$ aufzählbar, woraus nach Theorem 20.1 folgt, dass beide sogar entscheidbar sind.

von
6

Übungsaufgabe 12.5: Wir definieren die Sprache

$$L_\epsilon := \{\langle M\rangle \mid \text{ die TM } M \text{ hält auf dem leeren Wort } \epsilon\}$$

1. Zeigen Sie, dass die Sprache L_ϵ nicht entscheidbar ist, indem sie ein unentscheidbares Problem auf L_ϵ reduzieren.
2. Erläutern Sie Ihre Konstruktion.

Lösung: Wir reduzieren das Halteproblem H auf die Sprache L_ϵ.

Annahme: L_ϵ ist entscheidbar. Also existiert eine stets terminierende TM A_ϵ, die L_ϵ akzeptiert.

Zu jeder TM A und Eingabe w können wir eine TM A_w konstruieren, die sich wie folgt verhält:

- Zuerst schreibt A_w Zeichen-für-Zeichen das feste Wort w auf das Eingabeband.
- Dann fährt A_w wieder zurück bis zum 1. Zeichen von w.
- Danach wird in den Startzustand von A gesprungen.

Hilfssatz: Es gilt:

$$A \text{ hält auf } w \iff \epsilon \in L(A_w)$$

Beweis. Wenn $w \in L(A)$, dann wird A_w nach Konstruktion zunächst auf dem leeren Band w schreiben und dieses w dann akzeptieren, d.h. $\epsilon \in L(A_w)$. Wenn $\epsilon \in L(A_w)$, dann hat A_w zunächst auf das leeren Band w geschrieben und dann mit A dieses w akzeptiert, d.h. $w \in L(A)$. (Beweisende.)

Die Konstruktion von A_w kann durch eine TM $B_{A,w}$ geleistet werden, die A und w als Eingabe bekommt.

Wir konstruieren eine TM C, die A und w als Eingabe bekommt. Die TM C enthält $B_{A,w}$ als Unterprozedur und konstruiert zunächst aus A und w die TM A_w.

Die TM C enthält auch die hypothetische TM A_ϵ als Unterprozedur. Die TM A_w übergeben wir A_ϵ als Eingabe. Die TM C wird so definiert, dass sie genau dann akzeptiert, wenn auch A_ϵ akzeptiert. Da die TM A_ϵ nach Annahme stets terminiert, wird auch C dies tun. Es gilt:

$$C \text{ akzeptiert } \langle A, w\rangle \iff \epsilon \in L(A_w)$$

Zusammen mit dem Hilfssatz gilt dann:

$$C \text{ akzeptiert } \langle A, w\rangle \iff A \text{ hält auf } w$$

Also entscheidet C das Halteproblem. Da aber H unentscheidbar ist, kann keine stets terminierende, L_ϵ akzeptierende TM A_ϵ existieren. Also ist L_ϵ unentscheidbar.

von
6

Bonusaufgabe 12.6: Sei $f : \Sigma^* \to \Lambda^*$ eine totale(!) Funktion, von der wir zunächst nicht annehmen, dass sie Turing-berechenbar ist.

Sei \$ ein Zeichen, dass weder in Σ noch in Λ vorkommt. Definiere $L_f := \{w\$u \mid w \in \Sigma^*, u = f(w)\}$.

1. Zeigen Sie: Wenn $f : \Sigma^* \to \Lambda^*$ eine Turing-berechenbare Funktion ist, dann kann man eine DTM A konstruieren, die L_f akzeptiert und die auf allen Eingaben terminiert.

   Es reicht aus, wenn Sie die Arbeitsweise von A beschreiben.

   Hinweis: Erläutern Sie zunächst, dass es eine DTM B_f geben muss, die f berechnet. Konstruieren Sie mit Hilfe von B_f die DTM A. Konstruieren Sie A als DTM mit zwei Spuren.

   Lösung: Sei B_f die DTM, die f berechnet.

   Die zu konstruierende DTM A bekommt ein Wort $w\$u$ als Eingabe mit $w \in \Sigma^*$ und $u \in \Lambda^*$.

   Wir arbeiten mit A auf zwei Spuren. Auf der ersten Spur speichern wir die Eingabe; Auf der zweiten Spur simulieren wir die DTM B_f.

   - Zuerst kopieren wir das Teilwort w auf die 2. Spur und fahren den Kopf an den Anfang zurück.
   - Wir führen die Berechnung, die B_f ausführt aus: $q_0 w \vdash q_e f(w)$. Da f total ist, terminiert die Berechnung stets. Wir können annehmen, dass die DTM B_f terminiert, sobald sie einen Endzustand einnimmt (klar warum?).
   - Jetzt vergleichen wir das Teilwort u auf der 1. Spur mit $f(w)$ auf der 2. Spur, indem wir die Zeichen jeweils markieren und immer wieder zurückfahren.
   - Sind das Teilwort u und $f(w)$ gleich, dann akzeptieren wir im Endzustand, ansonsten halten wir in einem Nicht-Endzustand.

   Die DTM A_f hat die gesuchte Eigenschaft.

2. Sei A eine DTM, die L_f akzeptiert und auf allen Eingaben stets terminiert. Zeigen Sie, dass f Turing-berechenbar ist, indem Sie eine DTM B_f konstruieren, die $f : \Sigma^* \to \Lambda^*$ berechnet.

   Es reicht aus, wenn Sie die Arbeitsweise von B_f beschreiben.

   Hinweis: Verwenden Sie drei Spuren und enumerieren Sie alle Teilworte $u \in \Lambda^*$ lexikalisch auf.

   Lösung: Die zu konstruierende DTM B_f arbeitet auf drei Spuren. Auf der ersten Spur speichern wir die Eingabe; auf der zweiten Spur enumerieren wir alle Teilworte $u \in \Lambda^*$ in lexikalischer Ordnung; auf der dritten Spur simulieren wir die DTM B_f.

   Die zu konstruierende DTM B_f bekommt $w \in \Sigma^*$ als Eingabe.

   - Wir schreiben die Eingabe so auf drei Spuren um, so dass w auf der 1. Spur steht.
   - Wir intialisieren die 2. Spur mit $u = \epsilon$.
   - Wir beginnen folgende Schleife:
   - Wir kopieren das Teilwort w von der 1. auf die 3. Spur.
   - Dahinter schreiben wir auf die 3. Spur das Symbol \$.
   - Dahinter schreiben wir auf die 3. Spur das Wort u von der 2. Spur.
   - Wir führen A auf der 3. Spur aus, auf der nun das Wort $w\$u$ steht. Wir können annehmen, dass die DTM A terminiert, sobald sie einen Endzustand einnimmt (klar warum?). Da A stets terminiert, überprüfen wir, ob wir in einem Endzustand gelangt sind.

(a) Sind wir in einem Endzustand, dann wurde $w\$u$ akzeptiert, d.h. $u = f(w)$. Wir verlassen die Schleife.

(b) Andernfalls zählen wir auf der 2. Spur das nächste u in lexikalischer Ordnung auf und starten die Schleife erneut.

- Wir schreiben die Ausgabe $f(w)$ wieder auf eine Spur zurück, fahren an den Anfang von $f(w)$ und terminieren in einem Endzustand.

Die DTM A_f hat die gesuchte Eigenschaft, denn sie terminiert stets, da B auf jeder Eingabe terminiert und es zu jedem w auch einen Funktionswert gibt. Irgendwann wird u dieser Funktionswert sein und die Schleife wird verlassen. Dass A_f dann den Funktionswert berechnet ist auch klar.