



Aufgabe 1

- a) L r.e. $\implies \bar{L}$ r.e.

Diese Aussage stimmt nicht.

Gegenbeispiel: SAM ist r.e. und \overline{SAM} ist nicht r.e.. Beides wurde in der Vorlesung bewiesen und deshalb kann die Implikation nicht stimmen.

- b) L entscheidbar $\implies \bar{L}$ entscheidbar

Diese Aussage stimmt.

Bei einer Turing Maschine die L entscheidet müssen nur die Zustände für JA bzw. NEIN vertauscht werden um \bar{L} zu entscheiden

Alternative: L entscheidbar $\iff L$ akzeptierbar und \bar{L} akzeptierbar $\iff \bar{L}$ akzeptierbar und $\overline{\bar{L}} = L$ akzeptierbar $\iff \bar{L}$ akzeptierbar.

- c) L entscheidbar und \bar{L} nicht r.e. $\implies L$ regulär

Diese Aussage ist richtig.

L entscheidbar $\iff L$ akzeptierbar und \bar{L} akzeptierbar. Das bedeutet, daß die linke Seite der Implikation (L entscheidbar und \bar{L} nicht r.e.) stets falsch ist. Damit ist die Implikation richtig. („Aus etwas falschem folgt beliebiges.“)

- d) $L \leq L'$ und L r.e. $\implies L'$ r.e.

Diese Aussage stimmt nicht.

Intuition: L ist „leichter“ als L' und deshalb kann man nicht schliessen, dass L' r.e. ist, auch wenn L auf L' reduzierbar ist.

Wir widerlegen die Behauptung durch Angabe eines Gegenbeispiels. Wir können SAM auf die Sprache L aus Aufgabe 2 mittels $f(x) = x\$(M_2)$ reduzieren. (M_2 ist eine beliebige Sprache aus \overline{SAM} .) Dort zeigen wir, daß L nicht akzeptierbar ist.

- e) $L \leq L'$ und L' r.e. $\implies L$ r.e.

Diese Aussage ist richtig.

L' sei akzeptierbar, d.h. es gibt eine Turing-Maschine M_2 , die L' akzeptiert. $L \leq L'$ bedeutet, dass es eine berechenbare Funktion f gibt mit $x \in L \iff f(x) \in L'$. Diese werde realisiert durch die Turing-Maschine M_1 . Wir konstruieren daraus eine Turing-Maschine M , die L akzeptiert. Es sei eine Eingabe x gegeben. Zuerst simuliert M die Maschine M_1 und berechnet $f(x)$ (M_1 terminiert für alle $x \in L$), danach simuliert sie M_2 bei Eingabe $f(x)$. Sie akzeptiert $f(x)$ genau dann, wenn $f(x) \in L'$ ist, was gleichbedeutend mit $x \in L$ ist. Also akzeptiert M eine Eingabe x genau dann, wenn $x \in L$.

- f) $L \leq L'$ und L nicht r.e. $\implies L'$ nicht r.e.

Diese Aussage ist richtig.

Nehmen wir an, L' wäre akzeptierbar. Dann folgt aus Implikation e), dass L akzeptierbar wäre, was aber der Voraussetzung widerspricht.

Theoretische Informatik (WS04)

Prof. Dr. Raimund Seidel

Wei Ding, Dierk Johannes

Lösungsvorschlag zu Aufgabenblatt 9



g) L regulär $\implies L$ entscheidbar

Die Aussage ist richtig.

Für eine reguläre Sprache L existiert ein endlicher Automat, der L akzeptiert. Dieser kann von einer Turing-Maschine simuliert werden. Also ist L akzeptierbar. Bei den Absehlufseigenschaften für reguläre Sprachen hatten wir gezeigt, daß mit L auch \bar{L} regulär ist. Also ist auch \bar{L} akzeptierbar. Damit ist L entscheidbar.

h) L kontextfrei $\implies L$ r.e.

Diese Aussage ist wahr.

Beweis: Falls L kontextfrei ist, existiert ein Kellerautomat, der L akzeptiert. Somit gibt es auch eine Turing-Maschine, die dies tut und insbesondere folgt die Behauptung.

i) L kontextfrei $\implies L$ entscheidbar

Diese Aussage ist wahr.

Beweis: Im vorherigen Teil (h) wurde gezeigt: L kontextfrei $\implies L$ r.e., außerdem wurde bereits in der Vorlesung gezeigt: L entscheidbar $\Leftrightarrow L$ r.e. $\wedge \bar{L}$ r.e. Um beides kombinieren zu können bleibt z.Z.: L kontextfrei $\implies \bar{L}$ r.e.

Dazu zunächst eine intuitive Betrachtung: Sowohl reguläre als auch DKA-Sprachen sind unter Komplementbildung abgeschlossen, d.h. L DKA-Sprache $\implies \bar{L}$ DKA-Sprache \bar{L} r.e. Für NKA-Sprachen und somit für kontextfreie Sprachen allgemein ist dies nicht der Fall. Der Grund ist intuitiv folgender: Ein nichtdeterministischer Kellerautomat akzeptiert ein Wort genau dann, wenn es eine akzeptierende Übergangsfolge für dieses Wort gibt. Dabei ist es irrelevant, ob es beliebig viele weitere, nicht akzeptierende Übergangsfolgen für dieses Wort gibt. Genau deshalb funktioniert der Trick, der bei DEAs und DKAs zur Komplementbildung benutzt wird aber nicht mehr: dort werden normale und Endzustände getauscht, was dazu führt, daß der betreffende Automat die komplementäre Sprache akzeptiert. Beim NKA bedeutet dies: Wenn es einen akzeptierenden und einen nicht akzeptierenden Übergang gibt, akzeptieren sowohl der normale Automat, als auch der Automat mit vertauschten Zuständen beide das Wort. Was man bräuchte wäre ein Automat, der ein Wort explizit ablehnt, wenn es auch nur einen nicht akzeptierenden Übergang gibt, statt einem Automaten, der es akzeptiert, wenn auch nur ein akzeptierender Übergang existiert. Bei Turing-Maschinen mit expliziten Ja- und Nein-Endzuständen ist aber genau dies möglich und damit wird auch diese Sprache akzeptierbar.

Formal lässt sich dies jedoch auch über Grammatiken zeigen: Für jede kontextfreie Grammatik existiert eine (kontextfreie) Grammatik in Chomsky Normalform (Beweis siehe z.B. Uwe Schöning - Theoretische Informatik kurzgefasst Seite 53). Bei einer Grammatik in Chomsky Normalform haben alle Regeln eine der folgenden beiden Formen:

$$A \rightarrow BC$$

$$A \rightarrow a$$

Durch Anwendung einer Regel der ersten Form erhöht sich die Anzahl der Variablen um 1, durch Anwendung einer Regel der zweiten Form ersetzt man eine Variable durch ein Terminalsymbol. Mit einer solchen Grammatik kann man ein Wort der Länge n also in $2n - 1$ Schritten ableiten – Ausgehend von einem Startsymbol kann man durch $n - 1$

Theoretische Informatik (WS04)

Prof. Dr. Raimund Seidel

Wei Ding, Dierk Johannes

Lösungsvorschlag zu Aufgabenblatt 9



Anwendungen von Regeln der ersten Form n Variablen erzeugen, die man dann durch n Anwendungen von Regeln der zweiten Form in Terminale umwandelt. Nun kann man alle Wörter der Länge n bestimmen, die von der Grammatik erzeugt werden können und mit dem Eingabewort vergleichen. Es gibt maximal $|\Sigma|^n$ viele Wörter, für die man jeweils genau $2n - 1$ Ableitungsregeln anwenden muß. Beachtet man nun noch, daß eine Grammatik nur endlich viele Produktionsregeln haben kann, ist das für ein gegebenes Wort von beliebiger aber konstanter Länge n eine Konstante (wenn auch eine große). Daraus folgt die Behauptung.

j) $L = \{ \langle M \rangle \mid L(M) \text{ ist kontextfrei} \} \Rightarrow L \text{ ist entscheidbar}$

Diese Aussage ist falsch. Dies folgt sofort aus dem Satz von Rice.

Aufgabe 2

Es sei $\Sigma = \{0, 1, \$\}$. Wir betrachten

$$L = \{ u\$v \mid u \in SAM \text{ und } v \in \overline{SAM} \}.$$

Weder L noch \overline{L} sind akzeptierbar.

a) Wir zeigen $\overline{SAM} \leq L$. Dazu konstruieren wir eine TM-berechenbare Funktion $f : \Sigma^* \rightarrow \Sigma^*$ mit $x \in \overline{SAM} \iff f(x) \in L$.

Es sei M_1 eine Turing-Maschine mit $M_1 \in SAM$. Wir setzen $f(x) = \langle M_1 \rangle \x . Dann gilt:

$$\begin{aligned} x \in \overline{SAM} &\implies f(x) = \langle M_1 \rangle \$x \in L \\ x \notin \overline{SAM} &\implies x \in SAM \implies f(x) = \langle M_1 \rangle \$x \notin L. \end{aligned}$$

Aus der Vorlesung wissen wir, daß \overline{SAM} nicht akzeptierbar ist. Mit Behauptung f aus Aufgabe 1 ($L_1 \leq L_2$ und L_1 nicht akzeptierbar $\implies L_2$ nicht akzeptierbar) sehen wir, daß L nicht akzeptierbar ist.

b) Wir zeigen $\overline{SAM} \leq \overline{L}$. Dazu sei M_2 eine Turing-Maschine mit $M_2 \notin SAM$. Wir setzen $f(x) = x\$ \langle M_2 \rangle$. Dann gilt:

$$\begin{aligned} x \in \overline{SAM} &\implies f(x) = x\$ \langle M_2 \rangle \notin L \implies f(x) \in \overline{L} \\ x \notin \overline{SAM} &\implies x \in SAM \implies f(x) = x\$ \langle M_2 \rangle \in L \implies f(x) \notin \overline{L}. \end{aligned}$$

Wie oben folgt, daß \overline{L} nicht akzeptierbar ist.