Theoretische Informatik Gruppe 8
Loris Reiff

Endterm Vorbereitung – Lösungen

1. Betrachte

$$\begin{aligned} L_{\not\subseteq} &= \{\text{Kod}(M_1)\#\text{Kod}(M_2) \mid M_1 \text{ und } M_2 \text{ sind TMs mit } L(M_1) \not\subseteq L(M_2)\} \\ L_{\text{U},\lambda} &= \{\text{Kod}(M) \mid M \text{ ist eine TM und } \lambda \in L(M)\} \\ L_{\text{intersect}} &= \{\text{Kod}(M_1)\#\text{Kod}(M_2)\#w \mid M_1 \text{ und } M_2 \text{ sind TMs mit } w \in L(M_1) \cap L(M_2)\} \end{aligned}$$

   (a) Zeige $L_{\text{intersect}} \notin \mathcal{L}_{\text{R}}$

Lösung: Wir haben in der Vorlesung gesehen, dass $L_{\text{U}} \notin \mathcal{L}_{\text{R}}$. Wir zeigen nun $L_{\text{U}} \leq_{\text{EE}} L_{\text{intersect}}$, was $L_{\text{intersect}} \notin \mathcal{L}_{\text{R}}$ impliziert.
Folgende TM A transformiert eine Eingabe x für L_{U} in eine Eingabe für $L_{\text{intersect}}$:

- Prüfe, ob die Eingabe die Form $\text{Kod}(M)\#w$ für eine TM M hat.
  - Falls ja: Gib $\text{Kod}(M)\#\text{Kod}(M)\#w$ zurück.
  - Falls nein: gib λ aus

Wir haben offensichtlich

$$\begin{aligned} \text{Kod}(M)\#w \in L_{\text{U}} &\iff w \in L(M) \\ &\iff w \in L(M) \cap L(M) \\ &\iff \text{Kod}(M)\#\text{Kod}(M)\#w \in L_{\text{intersect}} \end{aligned}$$

und für $x \neq \text{Kod}(M)$ haben wir $x \notin L_{\text{U}} \implies \lambda \notin L_{\text{intersect}}$.
es gilt also $x \in L_{\text{U}} \iff A(x) \in L_{\text{intersect}}$.

(b) Zeige $L_{\text{intersect}} \leq_{\text{EE}} L_{\text{U},\lambda}$

Lösung: Folgende TM B transformiert eine Eingabe x für $L_{\text{intersect}}$ in eine Eingabe für $L_{\text{U},\lambda}$:

- Prüfe, ob die Eingabe die Form $\text{Kod}(M_1)\#\text{Kod}(M_2)\#w$ für TM M_1 und M_2 hat.
  - Falls ja: Konstruiere die TM $\overline{M}$, welche die Eingabe ignoriert und w parallel auf M_1 und M_2 simuliert und nur akzeptiert, falls beide TM akzeptieren, d.h. wenn M_1 oder M_2 verwirft, verwirft auch $\overline{M}$.
  - Falls nein: gib λ aus.

Wir zeigen nun: $x \in L_{\text{intersect}} \iff B(x) \in L_{\text{U},\lambda}$:
"$\Rightarrow$":
Sei $x \in L_{\text{intersect}}$, d.h. $x = \text{Kod}(M_1)\#\text{Kod}(M_2)\#w$ für TM M_1 und M_2 mit $w \in L(M_1) \cap L(M_2) \implies w \in L(M_1) \wedge w \in L(M_2)$. Folglich hält $\overline{M}$ gemäss Konstruktion für jede Eingabe in q_{accept}, wir haben insbesondere $\lambda \in L(\overline{M})$, also $B(x) \in L_{\text{U},\lambda}$.
"$\Leftarrow$":
Sei $x \notin L_{\text{intersect}}$, wir betrachten zwei Fälle:

- x hat nicht die Form $\text{Kod}(M_1)\#\text{Kod}(M_2)\#w$ für TM M_1 und M_2, dann ist $B(x) = \lambda \notin L_{\text{U},\lambda}$.
- $x = \text{Kod}(M_1)\#\text{Kod}(M_2)\#w$ für TM M_1 und M_2, d.h. $w \notin L(M_1) \cap L(M_2)$ das impliziert $w \notin L(M_1) \vee w \notin L(M_2)$, somit akzeptiert $\overline{M}$ keine Eingabe, also $\lambda \notin L(\overline{M}) = \emptyset$ und somit haben wir $B(x) \notin L_{\text{U},\lambda}$

(c) Zeige $L_{\mathrm{U},\lambda} \leq_{\mathrm{EE}} L_{\not\subseteq}$

Lösung: Folgende TM C transformiert eine Eingabe x für $L_{\mathrm{U},\lambda}$ in eine Eingabe für $L_{\not\subseteq}$:

- Prüfe, ob die Eingabe die Form $\mathrm{Kod}(M)$ für eine TM M hat.
  - Falls ja: Konstruiere die TM $\overline{M}$, welche die Eingabe ignoriert und λ auf M simuliert. Konstruiere des Weiter die TM $M_\emptyset$, welche alle Eingaben verwirft. Gib $\mathrm{Kod}(\overline{M})\#\mathrm{Kod}(M_\emptyset)$ zurück.
  - Falls nein: gib $\mathrm{Kod}(M_\emptyset)\#\mathrm{Kod}(M_\emptyset)$ aus

Wir zeigen nun: $x \in L_{\mathrm{U},\lambda} \iff C(x) \in L_{\not\subseteq}$:
Sei $x \in L_{\mathrm{U},\lambda}$, d.h. $x = \mathrm{Kod}(M)$ für eine TM M mit $\lambda \in L(M)$. Es gilt also $L(\overline{M}) = \Sigma^*$, und da $\Sigma^* \not\subseteq \emptyset = L(M_\emptyset)$ ist $C(x) \in L_{\not\subseteq}$
Sei $x \notin L_{\mathrm{U},\lambda}$, wir betrachten zwei Fälle:

- x hat nicht die Form $\mathrm{Kod}(M)$ für eine TM M, dann ist $C(x) = \mathrm{Kod}(M_\emptyset)\#\mathrm{Kod}(M_\emptyset)$ und da $\emptyset \subseteq \emptyset$ ist $C(x) \notin L_{\not\subseteq}$
- $x = \mathrm{Kod}(M)$ für eine TM M, dann gilt $\lambda \notin L(M)$, somit akzeptiert $\overline{M}$ keine Eingabe, also $L(\overline{M}) = \emptyset$ und somit haben wir $C(x) = \mathrm{Kod}(\overline{M})\#\mathrm{Kod}(M_\emptyset) \notin L_{\not\subseteq}$, da $\emptyset \subseteq \emptyset$.

2. (a) Entwerfe eine reguläre Grammatik für folgende Sprache:

$$L_a = \{w \in \{a,b\} \mid 1 \leq |w|_a \leq 2 \text{ oder } w \text{ enthält das Teilwort } baaab\}$$

Begründe deinen Entwurf kurz.

Lösung: Die Sprache ist äquivalent zu $L_1 \cup L_2$ mit

$$L_1 = \{w \in \{a,b\} \mid 1 \leq |w|_a \leq 2\}$$
$$L_2 = \{w \in \{a,b\} \mid w \text{ enthält das Teilwort } baaab\}$$

Wir konstruieren zuerst die Grammatiken für L_1 und L_2.
Sei $G_1 = (\{S_1, A_1, A_2\}, \{a,b\}, P_1, S_1)$ mit

$$\begin{aligned} P_1 = \{&S_1 \to bS_1 \mid aA_1, \\ &A_1 \to bA_1 \mid aA_2 \mid \lambda, \\ &A_2 \to bA_2 \mid \lambda\} \end{aligned}$$

Die Idee ist hierbei die Anzahl der bereits generierten a's über die Nichtterminale zu merken. Es gilt offensichtlich $L(G_2) = L_1$.
Und sei $G_2 = (\{S_2, E\}, \{a,b\}, P_2, S_2)$ mit

$$\begin{aligned} P_2 = \{&S_2 \to aS_2 \mid bS_2 \mid baaabE, \\ &E \to aE \mid bE \mid \lambda\} \end{aligned}$$

Hier ist die Idee mit den Regeln $S_2 \to aS_2 \mid bS_2$ einen beliebigen Präfix über $\{a,b\}$ zu generieren. Mit $S_2 \to baaabE$ stellen wir sicher, dass das generierte Wort das Teilwort $baaab$ enthält. Und mit $E \to aE \mid bE \mid \lambda$ kann ein beliebiger Suffix generiert werden.
Wir konstruieren nun die Vereinigung der zwei Grammatiken:
$G_a = (\{S, S_1, S_2, A_1, A_2, E\}, \{a,b\}, P, S)$ mit

$$\begin{aligned} P_1 = \{&S \to S_1 \mid S_2, \\ &S_1 \to bS_1 \mid aA_1, \\ &A_1 \to bA_1 \mid aA_2 \mid \lambda, \\ &A_2 \to bA_2 \mid \lambda, \\ &S_2 \to aS_2 \mid bS_2 \mid baaabE, \\ &E \to aE \mid bE \mid \lambda\} \end{aligned}$$

(b) Entwerfe eine Grammatik für folgende Sprache:

$$L_b = \{a^i b^j c^k \mid i, j, k \in \mathbb{N}, i = j + k\}$$

Begründe deinen Entwurf kurz.

Lösung: Sei $G_b = (\{S, X, Y, Z\}, \{a, b, c\}, P_b, S)$ mit

$$\begin{aligned} P_b = \{&S \to aSX \mid Y \mid Z, \\ &YX \to bY \mid bZ, \\ &ZX \to cZ, \\ &Z \to \lambda\} \end{aligned}$$

Die Idee ist wie folgt: $S \to aSX$ generiert eine beliebige Anzahl a's und für jedes a ein X, wobei die a's von den X durch ein S getrennt bleiben. S kann danach mit $S \to Y$ bzw. $S \to Z$ zu einem Cursor für b's bzw. c's umgewandelt werden. Die Regel $YX \to bY$ erlaubt das Umwandeln von X zu b's. Mit der Regel $YX \to bZ$ wird zustätzlich der Cursor in den Cursor für c umgewandelt. Mit $ZX \to cZ$ können dann die restlichen X zu c's umgewandelt werden.

(c) Betrachte die Grammatik $G_c = (\{S, X, Y\}, \{0, 1\}, P_c, S)$ mit

$$\begin{aligned} P_c = \{&S \to XY, \\ &X \to 0X1 \mid \lambda, \\ &Y \to 1Y1 \mid X\} \end{aligned}$$

Gib die erzeugte Sprache der Grammatik G_c an und begründe kurz.

Lösung:
Wir bemerken, die Regeln für X erzeugen $\{0^i 1^i \mid i \in \mathbb{N}\}$. Des Weitern sehen wir, dass die Regeln für Y folgende Sprache $\{1^i 0^j 1^{i+j} \mid i, j \in \mathbb{N}\}$ generieren. Mit $S \to XY$ haben wir also $\{0^i 1^i \mid i \in \mathbb{N}\} \cdot \{1^i 0^j 1^{i+j} \mid i, j \in \mathbb{N}\}$ und somit:

$$\begin{aligned} \{0^i 1^i \mid i \in \mathbb{N}\} \cdot \{1^j 0^k 1^{j+k} \mid j, k \in \mathbb{N}\} &= \{0^i 1^i 1^j 0^k 1^{j+k} \mid i, j, k \in \mathbb{N}\} \\ &= \{0^i 1^{i+j} 0^k 1^{j+k} \mid i, j, k \in \mathbb{N}\} \\ &= \{0^i 1^m 0^k 1^l \mid i + l = m + k, i \leq m\} \end{aligned}$$

Die erzeugte Sprache ist also: $L_c = \{0^i 1^j 0^k 1^l \mid i + l = j + k, i \leq j\}$

3. Seien $L \in \text{NTIME}(f)$ und $L' \in \text{TIME}(f)$. Zeige, dass dann $L - L' \in \text{NTIME}(f)$ gilt.

Lösung: Seien $L \in \text{NTIME}(f)$ und $L' \in \text{TIME}(f)$. Dann existieren eine nichtdeterministische k_1-Band-Turingmaschine M_1 für L und eine deterministische k_2-Band-Turingmaschine M_2 für L' mit $\text{Time}_{M_1}(n), \text{Time}_{M_2}(n) \in \mathcal{O}(f(n))$. Wir konstruieren hieraus eine nichtdeterministische $(k_1 + k_2)$-Band-TM M für $L - L'$ mit $\mathcal{O}(f(n))$ Zeitbedarf wie folgt. Zunächst simuliert M die Arbeit von M_2 auf der Eingabe w der Länge n auf den Arbeitsbändern $k_1 + 1$ bis $k_1 + k_2$. Falls M_2 den akzeptierenden Zustand erreicht hat, dann gilt $w \in L'$, also $w \notin L - L'$, also verwirft M die Eingabe. Falls M_2 den verwerfenden Zustand erreicht, dann gilt $w \notin L'$. In diesem Fall setzt M den Lesekopf auf dem Eingabeband zurück an den Anfang und startet eine Simulation von M_1 auf w auf den ersten k_1 Arbeitsbändern. Falls M_1 das Wort w akzeptiert, dann akzeptiert auch M. Die Zeitkomplexität von M lässt sich wie folgt abschätzen. Die Simulation von M_2 benötigt offenbar $\mathcal{O}(f(n))$ Schritte, das Zurücksetzen des Lesekopfes dann noch einmal höchstens $\mathcal{O}(f(n))$ Schritte. Wenn das Wort w von M_1 akzeptiert wird, gibt es nach Definition der nichtdeterministischen Zeitkomplexität auch eine Berechnung, in der die Simulation von M_1 in $\mathcal{O}(f(n))$ Zeit durchgeführt wird. Also gilt $\text{Time}_M(n) \in \mathcal{O}(f(n))$.

4. (a) Sei VIERFACH-SAT die Menge aller KNF-Formeln, welche vier erfüllende Belegungen hat.
Zeige, dass VIERFACH-SAT NP-vollständig ist.

Lösung: Es gilt VIERFACH-SAT $\in$ NP, denn eine NTM kann die vier Belegungen der Formel nichtdeterministisch erraten und prüfen, ob sie erfüllt werden. Dies ist offensichtlich in polynomieller Zeit möglich. (Es kann natürlich auch mit einem polynomiellen Verifizierer argumentiert werden, da VC = NP.)

Wir zeigen nun SAT $\leq_p$ VIERFACH-SAT, was die Behauptung impliziert:

Sei $F = F_1 \wedge F_2 \wedge \cdots \wedge F_m$ eine Formel in KNF über die Variablen $X = \{x_1, \ldots, x_n\}$. Wir konstruieren aus F eine Eingabe C für das VIERFACH-SAT Problem, so dass

$$F \in \text{SAT} \iff C \in \text{VIERFACH-SAT}$$

Dies tun wir wie folgt: Seien $y_1, y_2, y_3 \notin X$ drei neue Variablen, dann definieren wir $C = F_1 \wedge F_2 \wedge \cdots \wedge F_m \wedge (y_1 \vee y_2 \vee y_3)$.
Diese Konstruktion von C ist offensichtlich in polynomieller Zeit möglich.

Wir zeigen nun $F \in \text{SAT} \iff C \in \text{VIERFACH-SAT}$:

"$\Rightarrow$":
Sei $F \in$ SAT, es gibt also eine Belegung α, welche die Klauseln $F_1, F_2, \ldots, F_m$ erfüllt. Wir können α zu vier Belegungen $\widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha}_3, \widehat{\alpha}_4$ auf $X \cup \{y_1, y_2, y_3\}$ erweitern. Dies tun wir wie folgt: Wir setzen $\widehat{\alpha}_i(x) = \alpha(x)$ für alle $x \in X$ und $i \in \{1, 2, 3, 4\}$ und

- $\widehat{\alpha}_1(y_1) = \widehat{\alpha}_1(y_2) = \widehat{\alpha}_1(y_3) = 1$
- $\widehat{\alpha}_2(y_1) = 0$ und $\widehat{\alpha}_2(y_2) = \widehat{\alpha}_2(y_3) = 1$
- $\widehat{\alpha}_3(y_2) = 0$ und $\widehat{\alpha}_3(y_1) = \widehat{\alpha}_3(y_3) = 1$
- $\widehat{\alpha}_4(y_3) = 0$ und $\widehat{\alpha}_4(y_1) = \widehat{\alpha}_4(y_2) = 1$

Da $\widehat{\alpha}_i(x) = \alpha(x)$ für alle $x \in X$, erfüllen alle $\widehat{\alpha}_i$ die Klauseln $F_1, \ldots, F_m$. Und da immer eine Variable aus $(y_1 \vee y_2 \vee y_3)$ auf 1 gesetzt wird, ist auch diese Klausel erfüllt. Wir haben also vier erfüllende Belegungen für C und somit $C \in$ VIERFACH-SAT.

"$\Leftarrow$":
Sei $F \notin$ SAT, d.h. es gibt keine Belegung, die alle Klauseln $F_1, F_2, \ldots, F_m$ erfüllt. Es gibt also insbesondere keine Belegung für C, da y_1, y_2, y_3 nicht in F vorkommen. Es gilt also $C \notin$ VIERFACH-SAT.

Somit ist VIERFACH-SAT NP-schwer und mit VIERFACH-SAT $\in$ NP schliessen wir, dass VIERFACH-SAT NP-Vollständig ist.

(b) **[Aufgabe 4 – Endterm 2017]**
Wir nennen eine Klausel einer KNF-Formel *monoton*, wenn sie entweder keine negierten Variablen oder nur negierte Variablen enthält. Wir betrachten die Menge non-3-monotone-3SAT aller erfüllbaren KNF-Formeln, die aus Klauseln der Länge höchstens 3 bestehen und keine monotonen Klauseln der Länge genau 3 enthalten (Monotone Klauseln der Längen 2 und 1 sind somit erlaubt). Zeige, dass non-3-monotone-3SAT NP-vollständig ist.

Lösung: Es ist offensichtlich möglich eine Belegung in polynomieller Zeit zu prüfen. Es gilt also non-3-monotone-3SAT $\in$ NP.

Wir zeigen nun 3SAT $\leq_p$ non-3-monotone-3SAT, was impliziert, dass non-3-monotone-3SAT auch NP-Schwer ist.

Sei $F = F_1 \wedge F_2 \wedge \cdots \wedge F_m$ eine Formel in 3KNF über die Variablen $X = \{x_1, \ldots, x_n\}$.

Wir konstruieren aus F eine Eingabe C für das non-3-monotone-3SAT Problem, so dass

$$F \in \text{3SAT} \iff C \in \text{non-3-monotone-3SAT}$$

Dies tun wir wie folgt: Alle Klauseln welche eine Länge kürzer als 3 haben, bleiben unverändert. Eine Klausel $F_i = (l_{1,i} \vee l_{2,i} \vee l_{3,i})$ ersetzen wir durch $(l_{1,i} \vee y_{1,i}) \wedge (\overline{y_{1,i}} \vee l_{2,i} \vee y_{2,i}) \wedge (l_{3,i} \vee \overline{y_{2,i}})$, wobei $y_{1,i}, y_{2,i} \notin X$ jeweils neue Variablen sind. Diese Konstruktion von C ist offensichtlich in polynomieller Zeit möglich.

Wir zeigen nun $F \in \text{3SAT} \iff C \in \text{non-3-monotone-3SAT}$:

"$\Rightarrow$":

Sei $F \in$ 3SAT, es gibt also eine erfüllende Belegung α, die alle Klauseln F_i erfüllt, d.h. mindestens ein Literal pro Klausel ist 1. Folgende Belegung β erfüllt also C: $\beta(x) = \alpha(x)$ für alle $x \in X$ und

$$\beta(y_{1,i}) = \begin{cases} 1 & \text{falls } l_{1,i} = 0 \\ 0 & \text{sonst} \end{cases} \qquad \beta(y_{2,i}) = \begin{cases} 0 & \text{falls } l_{3,i} = 0 \\ 1 & \text{sonst} \end{cases}$$

Alle Klauseln der Länge kleiner 3, werden trivialerweise erfüllt. Und mit $\beta(y_{1,i})$ wird garantiert, dass $(l_{1,i} \vee y_{1,i})$ erfüllt ist, und mit $\beta(y_{2,i})$ wird garantiert, dass $(l_{3,i} \vee \overline{y_{2,i}})$ immer erfüllt ist. Man bemerke, dass auch $(\overline{y_{1,i}} \vee l_{2,i} \vee y_{2,i})$ immer erfüllt ist. Somit $C \in$ non-3-monotone-3SAT.

"$\Leftarrow$":

Sei $F \notin$ 3SAT, es gibt also keine erfüllende Belegung α, d.h. es gibt eine Klausel F_i welche nicht erfüllt ist. Hat die Klausel eine Länge kleiner 3, so ist der Fall trivial. Hat die Klausel die Länge drei, so sehen wir, dass es auch keine erfüllende Belegung für die Konstruierte Formel gibt. Da $\alpha(l_{1,i}) = \alpha(l_{2,i}) = \alpha(l_{3,i}) = 0$ muss gem. Konstruktion $\beta(y_{1,i}) = 1$ und $\beta(y_{2,i}) = 0$ gelten, somit ist also $(\overline{y_{1,i}} \vee l_{2,i} \vee y_{2,i})$ nicht erfüllt. Wir haben also $C \notin$ non-3-monotone-3SAT.