Einführung in die theoretische Informatik

Sommersemester 2017 – Abgabeblatt Lösungsskizze 3

Abgabeblatt

Wir unterscheiden zwischen Übungs- und Abgabeblättern. Mit diesem *Abgabeblatt* fragen wir eine *Auswahl* an Kernaspekten ab, die auf den Trainingsblättern 10 bis 12 zu finden sind. Auch die Kernaspekte, die sich nicht auf Abgabeblättern finden, sind für die Klausur relevant. Wenn Sie die Abgabeblätter regelmäßig abgeben, können Sie zusätzlich einen *Bonus* von 0,3 (nach Bestehen) für beide Klausuren erreichen. Die genauen Regularien, wann Sie einen Notenbonus erhalten und wann nicht, finden Sie auf der Vorlesungshomepage.

AUFGABE 3.1. (*Restless-TMs*) 3 Punkte

Wir bezeichnen eine TM $\mathsf{M} = (\mathsf{Q}, \Sigma, \Gamma, \delta, \mathsf{q}_0, \Box, \mathsf{F})$ als Restless-TM falls $\delta \colon \mathsf{Q} \times \Gamma \to \mathsf{Q} \times \Gamma \times \{L, R\}$. Somit bewegt eine Restless-TM in jedem Schritt ihren Kopf und kann nicht an der gleichen Bandposition verweilen. Geben Sie eine allgemeine Übersetzung von einer TM $\mathsf{M} = (\mathsf{Q}, \Sigma, \Gamma, \delta, \mathsf{q}_0, \Box, \mathsf{F})$ zu einer Restless-TM $\mathsf{M}' = (\mathsf{Q}, \Sigma, \Gamma, \delta', \mathsf{q}_0, \Box, \mathsf{F})$ an, so dass $\mathsf{L}_F(\mathsf{M}) = \mathsf{L}_F(\mathsf{M}')$ gilt.

- Definieren Sie nur die neue Transitionsfunktion δ'. Alle anderen Komponenten der TM M sollen unverändert übernommen werden.
- Stellen sie sicher, dass falls die TM M nur die Kopfbewegungen $\{N, R\}$ verwendet, M′ nur R verwendet.

Lösungsskizze

Alle Transitionen, die den Kopf nicht bewegen, sammeln wir in $\mathcal{N}$:

$$\mathcal{N}(\delta) = \{((q, a), (p, b)) \mid (q, a, p, b, N) \in \delta\}$$

…und wir bezeichnen mit $\mathcal{N}(\delta)^*$ die reflexiv-transitive Hülle von $\mathcal{N}(\delta)$ über der Grundmenge $\mathsf{Q} \times \Gamma$. Dann definieren wir δ':

$$\begin{aligned}\delta' =&\{(q, a, p, b, D) \mid ((q, a), (r, c)) \in \mathcal{N}(\delta)^* \wedge (r, c, p, b, D) \in \delta \wedge D \in \{L, R\}\} \\ &\cup \{(q, a, p, b, R) \mid ((q, a), (p, b)) \in \mathcal{N}(\delta)^* \wedge p \in \mathsf{F}\}\end{aligned}$$

Hierbei müssen wir darauf achten, dass falls ein Endzustand über $\mathcal{N}(\delta)^*$ erreicht werden kann, wir eine neue Transition, die in einen Endzustand mit einer rechts Bewegung geht, hinzufügen.

AUFGABE 3.2. (*Reguläre Sprachen und Turing Maschinen*) 6 Punkte

Wir untersuchen den Zusammenhang zwischen speziellen Turing Maschinen und regulären Sprachen.

(a) Sei M eine Restless-TM (siehe Aufgabe 3.1), die ihren Kopf nicht nach links bewegen kann. Konstruieren Sie einen ε-NFA N mit $\mathsf{L}_F(\mathsf{M}) = \mathsf{L}(\mathsf{N})$.

(b) Sei D ein DFA. Konstruieren Sie eine TM M mit $\mathsf{L}(\mathsf{D}) = \mathsf{L}_F(\mathsf{M})$.

Geben Sie für beide Konstruktionen die Definitionen der Zustände und der Transitionen an, sowie eine Begründung für die Korrektheit, indem Sie die Läufe auf den FAs mit den Berechnungen der TMs vergleichen.

Lösungsskizze

(a) Sei $\mathsf{M} = (\mathsf{Q}, \Sigma, \Gamma, \delta, \mathsf{q}_0, \Box, \mathsf{F})$ die zu übersetzende Restless-TM mit $\delta \colon \mathsf{Q} \times \Gamma \to \mathsf{Q} \times \Gamma \times \{R\}$. Wir definieren dann $\mathsf{N} = (\{q, q' \mid q \in \mathsf{Q}\}, \Sigma, \delta', \mathsf{q}_0, \{q, q' \mid q \in \mathsf{F}\})$ und

$$\begin{aligned}\delta' &= \{(q, a, p) \mid (q, a, p, x, R) \in \delta, a \in \Sigma, q \notin \mathsf{F}\} \cup \{(q, a, q) \mid q \in \mathsf{F}, a \in \Sigma\} \\ &\cup \{(q, \boldsymbol{\varepsilon}, p') \mid (q, \Box, p, x, R) \in \delta\} \cup \{(q', \boldsymbol{\varepsilon}, p') \mid (q, \Box, p, x, R) \in \delta\}\end{aligned}$$

Korrektheit:
Da die TM M ihren Kopf nur nach rechts bewegen kann ist, ist es unerheblich was die Maschine auf das Band schreibt und was links in einer Konfiguration steht.
Sei $w \in \mathsf{L}_F(\mathsf{M})$. Dann existiert eine Berechnung $(\varepsilon, \mathsf{q}_0, w) \to^* (\alpha, q, \beta)$ mit $q \in \mathsf{F}$. Falls die TM kein $\Box$ gelesen hat, dann kann der ε-NFA die folgende Schritte vollziehen: $\delta'(\mathsf{q}_0, w) \supseteq \delta'(q, \beta) \supseteq \{q\} \subseteq \mathsf{F}$. Somit $w \in \mathsf{L}(\mathsf{N})$.
Falls die TM mindestens ein $\Box$ (von Zustand p aus) gelesen hat: $\delta'(\mathsf{q}_0, w) \supseteq \delta'(p, \varepsilon) \supseteq \delta'(p', \varepsilon) \ni q'$ und somit $w \in \mathsf{L}(\mathsf{N})$.
Die andere Richtung ist analog.

(b) Sei $\mathsf{D} = (\mathsf{Q}, \Sigma, \delta, \mathsf{q}_0, \mathsf{F})$ der zu übersetzende DFA.
Intuition: Die TM liest einfach Eingabe von links nach rechts und vollzieht die Zustandswechsel, die der DFA für das gelesene Zeichen machen würde. Bei Erreichen des rechten Rands der Eingabe, stoppt die TM und akzeptiert, falls der DFA sich in einem Endzustand befindet.
Formal: $\mathsf{M} = (\mathsf{Q} \cup \{q_f\}, \Sigma, \Sigma \cup \{\Box\}, \delta', \mathsf{q}_0, \Box, \{q_f\})$ mit $q_f \notin \mathsf{Q}$ und

$$\begin{aligned} \delta'(q, a) &= (p, a, R) \qquad &&\text{für alle } (q, a, p) \in \delta \\ \delta'(q, \Box) &= (q_f, \Box, R) \qquad &&\text{für alle } q \in \mathsf{F} \end{aligned}$$

Korrektheit:
Sei $w \in \mathsf{L}(\mathsf{D})$. Dann können wir aus dem akzeptierendem Lauf $\delta(\mathsf{q}_0, w) = q \in \mathsf{F}$ eine akzeptierende Berechnung für die TM M konstruieren: $(\varepsilon, \mathsf{q}_0, w) \to^* (w, q, \varepsilon) \to (w\Box, q_f, \varepsilon)$ mit $q_f \in \mathsf{F}$. Somit $w \in \mathsf{L}_F(\mathsf{M})$. Der Fall $w \notin \mathsf{L}(\mathsf{D}) \implies w \notin \mathsf{L}_F(\mathsf{M})$ folgt analog.

AUFGABE 3.3. (*Collatz-Vermutung*) 4 Punkte

Zu einem Startwert $a_0 \in \mathbb{N}$ definieren wir eine Folge $(a_n)_{n \in \mathbb{N}_0}$ wie folgt:

$$a_{n+1} \coloneqq \begin{cases} a_n/2 & a_n \text{ gerade} \\ 3a_n + 1 & a_n \text{ ungerade} \end{cases}$$

Die seit 1937 unbewiesene Collatz-Vermutung besagt:

Für alle positiven Startwerte $a_0 \in \mathbb{N}$ gibt es einen Index $i \in \mathbb{N}_0$, so dass $a_i = 1$.

Nehmen Sie an, es gibt ein Programm N, welches als Eingabe ein WHILE-Programm P mit genau einer Eingabevariable nimmt und zu jedem solchen P angibt, ob P die Nullfunktion berechnet. *Zeigen Sie, dass Sie dann die Collatz-Vermutung beweisen oder widerlegen können.*
Hinweise:
- Geben Sie auch das WHILE-Programm P, das sie für Ihren Beweis verwendet haben, an.
- Sie dürfen jede Syntax, die in den Folien für WHILE-Programme eingeführt worden ist, verwenden.

Lösungsskizze
Das folgende Programm P terminiert, falls es für einen Startwert a_0 (in Variable x) in der Collatz-Folge das Folgeglied $a_i = 1$ findet. Beachte, dass für die Eingabe $x = 0$ die Collatz-Folge nicht definiert ist und das Programm sofort mit 0 terminiert. Zur besseren Lesbarkeit verwenden wir $y = x_0$, $x = x_1$, $z = x_2$, $c = x_3$ als Abkürzungen. Berechnet nun das Programm die Nullfunktion, so ist die Collatz-Vermutung korrekt, da für alle Startwerte das Programm hält und 0 zurückgibt. Berechnet das Programm nicht die Nullfunktion, so gibt es mindestens einen Startwert, für dass das Programm nicht terminiert.
Programm P:

```
x := x - 1;
WHILE x ≠ 0 DO
  x := x + 1;
  c := 2;
  z := x MOD c;
  IF z = 0 DO
    x := x DIV c
  ELSE
    z := x + x;
    x := z + x;
    x := x + 1
  END;
  x := x - 1
END;
y := 0
```

AUFGABE 3.4. (*Reduktionen*) 6 Punkte

Betrachten Sie die folgende Menge:

$$A \coloneqq \{w \in \{0,1\}^* \mid \exists x \in \{0,1\}^*.\ \varphi_w(x) = |x|\}$$

Zeigen Sie die nun folgenden Behauptungen:
(a) A ist nicht entscheidbar.
(b) A ist semi-entscheidbar.
(c) $\overline{A}$ ist nicht semi-entscheidbar.
Hinweise:
- (a): Geben Sie eine passende Reduktion an und verwenden Sie nicht den Satz von Rice.

- (b): Beschreiben Sie eine passende Turing-Maschine.

Lösungsskizze

(a) $\mathsf{H}_0 \leq A$:
Reduktion: f bildet jedes $w \in \{0,1\}^*$ auf die Kodierung $f(w)$ einer TM $\mathsf{M}_{f(w)}$ ab, die erst die Eingabe löscht und dann $\mathsf{M}_w[\varepsilon]$ simuliert. Sobald die simulierte Turing-Maschine hält, gibt $\mathsf{M}_{f(w)}$ 0 aus.
Berechenbarkeit: Diese Reduktion ist berechenbar, da das Löschen der Eingabe, sowie die Simulation berechenbar sind.
Korrektheit: Sei $w \in \mathsf{H}_0$, dann $\mathsf{M}_{f(w)}[x] \downarrow$ für alle $x \in \{0,1\}^*$ mit Ausgabe 0, also insbesondere für $x = \varepsilon$. Somit $f(w) \in A$. Sei $w \notin \mathsf{H}_0$, dann hält $\mathsf{M}_{f(w)}[x]$ nie für alle $x \in \{0,1\}^*$ und somit $f(w) \notin A$.
Da H_0 unentscheidbar ist, ist A auch unentscheidbar.

(b) **Algorithmus:** Die TM enumeriert für $i = 0, 1, 2, \ldots$ alle Eingaben $x \in \bigcup_{k=0}^{i} \{0,1\}^k$ für die TM M_w, simuliert $\mathsf{M}_w[x]$ für i Schritte und prüft, ob $\varphi_w(x) = |x|$. Falls dieser Test wahr ist, terminiert die Maschine und akzeptiert w.
Korrektheit: Sei $w \in A$. Dann terminiert der Algorithmus, da es eine Eingabe x mit $\varphi_w(x) = |x|$ gibt und die $\mathsf{M}_w[x]$ nach einer fixen Anzahl an Schritten terminiert. Sei $w \notin A$. Dann terminiert der Algorithmus nie, da die Abbruchbedingung nie wahr ist.

(c) Anwendung von Satz 5.42, liefert das gewünschte Ergebnis. Alternativ: Aus (a) folgt $\overline{\mathsf{H}_0} \leq \overline{A}$. Da $\overline{\mathsf{H}_0}$ nicht semi-entscheidbar ist, ist $\overline{A}$ nicht semi-entscheidbar.

AUFGABE 3.5. (*Polynomielle Reduktionen*) 6 Punkte

Sei $\mathcal{F}$ die Menge aller aussagenlogischen Formeln über eine unendliche Menge $\mathcal{V}$ von Variablen. Betrachten Sie folgende Definitionen:

- Sei $\sigma\colon \mathcal{V} \to \{0,1\}$ eine Belegung. Wir bezeichnen mit $\overline{\sigma}$ die Negation von σ und definieren $\overline{\sigma}(x) \coloneqq 1 - \sigma(x)$.
- Eine Belegung $\sigma\colon \mathcal{V} \to \{0,1\}$ wird konsistent mit der Semantik der Aussagenlogik auf $\mathcal{F} \to \{0,1\}$ erweitert.
- Mit UNSAT bezeichnen wir das Problem, gegeben eine aussagenlogische Formel F zu entscheiden, ob F *nicht* erfüllbar ist. Formal:
$$\mathsf{UNSAT} \coloneqq \{F \in \mathcal{F} \mid \forall \sigma\colon \mathcal{V} \to \{0,1\}.\ \sigma(F) = 0\}$$
- Mit SYM bezeichnen wir das Problem, gegeben eine aussagenlogische Formel F zu entscheiden, ob F symmetrisch ist, d.h. für alle Belegungen σ gilt: $\sigma(F) = \overline{\sigma}(F)$. Formal:
$$\mathsf{SYM} \coloneqq \{F \in \mathcal{F} \mid \forall \sigma\colon \mathcal{V} \to \{0,1\}.\ \sigma(F) = \overline{\sigma}(F)\}$$

Zeigen Sie die folgenden Aussagen, in dem Sie geeignete Reduktionen angeben. Zeigen Sie auch die Korrektheit der Reduktionen und begründen Sie, warum die Reduktionen in polynomieller Zeit liegen.

(a) $\mathsf{UNSAT} \leq_p \mathsf{SYM}$

(b) $\mathsf{SYM} \leq_p \mathsf{UNSAT}$

Lösungsskizze

(a) **Reduktion:**
$f(F) = F \wedge x$, wobei $x \in \mathcal{V}$ eine Variable ist, die nicht in F vorkommt.

Korrektheit:
Sei $F \in \mathsf{UNSAT}$. Dann gilt für alle Belegungen σ: $\sigma(F) = 0$ und somit $\sigma(f(F)) = 0$ und $\overline{\sigma}(f(F)) = 0$. Somit $f(F) \in \mathsf{SYM}$.
Sei $F \in \overline{\mathsf{UNSAT}}$. Dann gibt es eine Belegung σ mit $\sigma(F) = 1$ und $\sigma(x) = 1$, da x nicht in F vorkommt. Somit $\sigma(f(F)) = 1$ und $\overline{\sigma}(f(F)) = 0$.

Polynomielle Berechenbarkeit:
Die Reduktion muss die Eingabe nur einmal linear durchsuchen um eine nicht verwendete Variable x zu finden. Diese wird dann an F angehängt.

(b) **Reduktion:**
$f(F) = F \wedge \neg F^*$ mit $F^* \coloneqq F[x \mapsto \neg x \mid x \in \mathcal{V}]$, d.h. wir ersetzen jedes $x \in \mathcal{V}$ mit $\neg x$. Es gilt somit $\sigma(F) = \overline{\sigma}(F^*)$ und $\overline{\sigma}(F) = \sigma(F^*)$ für alle Belegungen σ.

Korrektheit:
Sei $F \in \mathsf{SYM}$. Dann gilt für alle Belegungen σ: $\sigma(F) = \overline{\sigma}(F)$ und somit auch $\sigma(F) = \sigma(F^*)$. Dann gilt $\sigma(f(F)) = \min(\sigma(F), 1 - \sigma(F)) = 0$ und somit $f(F) \in \mathsf{UNSAT}$.
Sei $F \in \overline{\mathsf{SYM}}$. Dann gibt es eine Belegung σ mit $\sigma(F) \neq \overline{\sigma}(F)$. Sei o.B.d.A. $\sigma(F) = 1$. Dann $\sigma(F^*) = 0$ und somit $\sigma(f(F)) = \min(\sigma(F), 1 - \sigma(F^*)) = 1$. Damit $f(F) \notin \mathsf{UNSAT}$.

Polynomielle Berechenbarkeit:
Die Reduktion muss F einmal kopieren und x mit $\neg x$ ersetzen, was in polynomieller Laufzeit gemacht werden kann.