Prof. T. Nipkow — Technische Universität München
K. Kappelmann, J. Rädle, L. Stevens — Lehrstuhl für Logik und Verifikation

ABGABE 15.06.2020, 23:59 (Σ 4,5 P.)

Einführung in die Theoretische Informatik

Sommersemester 2020 – Übungsblatt 7

AUFGABE 7.1. (*Here we convert again*)

Konstruieren Sie eine Turingmaschine $T = (Q, \{|\}, \{|, 0, 1\}, \delta, q_0, \square, F)$, die als Eingabe eine mit | unär kodierte Zahl nimmt. Die gegebene Zahl soll in die entsprechende Binärzahl im Little-Endian-Format umgewandelt werden.. Beispielsweise soll T die Strichfolge |||| durch die Folge 001 ersetzen und dann in einem Endzustand halten. Die leere Eingabe kodiert 0. Ihre Lösung soll die Konvention aus der Vorlesung, dass Endzustände keine ausgehenden Kanten haben dürfen, respektieren. Es wird eine Webseite bereitgestellt, auf der Sie die Turingmaschine konstruieren, visualisieren und mit Eingaben testen können. Schauen Sie sich dort insbesondere das Beispiel *binary increment* an. Die Abgabe der Aufgabe erfolgt mit TUMJudge, laden Sie sich das Template auf Moodle herunter in dem erklärt wird wie die Abgabe aussehen soll. 1P

AUFGABE 7.2. (*. . . sehen deine Schuhe prima aus!*)

Wir definieren eine alternative Akzeptanzbedingung für Schleifomaten: Ein Schleifomat A mit Endzuständen ist ein 7-Tupel $(Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$. Die ersten 6 Einträge bleiben unverändert wie auf Hausaufgabenblatt 6 definiert. Der neue Eintrag $F \subseteq Q$ kennzeichnet die Endzustände des Schleifomats. Die mit Endzuständen akzeptierte Sprache eines Schleifomats ist definiert als 2P

$$L_F(M) \coloneqq \left\{ w \in \Sigma^* \mid \exists q_f \in F, \gamma \in \Gamma^*.\ (q_0, w, Z_0) \to_M^* (q_f, \varepsilon, \gamma) \right\},$$

wobei γ die Ausgabe der Berechnung des Schleifomats ist. Wie auch bei Turingmaschinen nehmen wir an, dass $\delta(q_f, a)$ nicht definiert ist für $q_f \in F$ und $a \in \Gamma$.

Zeigen Sie nun, dass jede Turingmaschine durch einen Schleifomat mit Endzuständen simuliert werden kann, indem Sie eine passende Konstruktion formal angeben (d.h. jede Komponente des 7-Tupels muss präzise definiert sein). Dabei soll sowohl die Akzeptanz eines Wortes als auch die Ausgabe der Turingmaschine simuliert werden. Letzteres bedeutet, dass falls eine TM mit Band $\square \cdots \square w_1 \cdots w_n \square \cdots \square$ mit $w_1, \ldots, w_n \in \Gamma$ hält, die Schleife schlussendlich bei $\square^k w_1 \cdots w_n \square^m$ für beliebiges $k, m \in \mathbb{N}$ stehen soll. Ein ungültiger, finaler Schleifeninhalt wäre hingegen $w_2 \cdots w_n w_1$. Die finale Kopfposition der TM muss nicht ersichtlich sein.

Beschreiben Sie dabei auch die Idee ihrer Konstruktion informell (ansonsten erheblicher Punkteabzug)! Die Korrektheit der Konstruktion müssen Sie nicht beweisen.

AUFGABE 7.3.

Entscheiden Sie ob die folgenden Aussagen wahr oder falsch sind. Begründen Sie ihre Behauptung. (Ein ausführlicher formaler Beweis ist nicht gefordert, aber eine umfassende Begründung) 1,5P

(a) Sei k eine Konstante. Die Klasse der Turing Maschinen die ihren Kopf maximal k mal bewegen können akzeptiert genau die regulären Sprachen.

(b) Die Klasse der Turing Maschinen die ihren Kopf nur nach rechts bewegen können akzeptiert genau die regulären Sprachen.

(c) Für eine beliebige, aber feste, Haskell Funktion P mit n Argumenten ist die Funktion

$$f_P(x) = \begin{cases} 1 \text{ falls } P \text{ für alle (typkorrekten) Inputs } x_1, \ldots, x_n \text{ terminiert} \\ 0 \text{ sonst} \end{cases}$$

berechenbar.

AUFGABE 7.4. (***Bonusaufgabe***)

Mit dem Verfahren auf Folie 158 der Vorlesung kann man für eine CFG G eine CFG G' in Chomsky-Normalform konstruieren, so dass $L(G') = L(G) \setminus \{\varepsilon\}$ gilt. Auf Folie 152 wird behauptet dass man die Produktion $S \to \varepsilon$ zu G' hinzufügen kann um G'' mit $L(G'') = L(G)$ zu erhalten, wenn $\varepsilon \in L(G)$. 0,5 + 1 Bonuspunkte

(a) Man füge die Produktion $S \to \varepsilon$ zu einer beliebigen Grammatik G mit Startsymbol S in CNF hinzu um die Grammatik G' zu erhalten. Zeigen Sie dass man daraus im Allgemeinen nicht $\mathsf{L}(G') = \mathsf{L}(G) \cup \{\varepsilon\}$ folgern kann.

(b) Sei G eine beliebige CFG mit Startsymbol S und $\varepsilon \in \mathsf{L}(G)$. Seien G' und G'' die Grammatiken die daraus nach dem Verfahren von oben hervorgehen. Zeigen oder widerlegen Sie $\mathsf{L}(G'') = \mathsf{L}(G)$.

> . . . so that one day, every application you run on your computer, can be run within PowerPoint.
>
> — Tom Wildenhain – On The Turing Completeness of PowerPoint