Prof. Javier Esparza
Philipp Czerner, Kevin Kappelmann, Valentin Krasotin
Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2024 – Hausaufgabenblatt 7

Abgabe: 10.06.2024, 12:00 CEST

- Die Aufgaben werden in folgender Reihenfolge korrigiert: H7.2, H7.3.
- Die Knobelaufgabe bitte separat auf Moodle abgeben. Sie wird korrigiert.

AT-Aufgabe H7.1. (*VerCYKendes KlapPDAch*) unkorrigiert (2+2+1+1+3+3 Punkte)

Bearbeiten Sie folgende Aufgabe mit Automata Tutor.

Bearbeiten Sie die Hausaufgaben H7.1 (a–d). Bei den *PDA Construction*-Aufgaben darf ihr konstruierter PDA nicht zu viele Zustände oder zu viele Stacksymbole haben (siehe Aufgabenstellung). Wenn Sie einen ε-Übergang angeben wollen, geben Sie statt ε bitte `E` ein (siehe Hinweisbox).

Aufgabe H7.2. (*Zuschnitt*) 7 Punkte

Sei $\Sigma := \{a, b\}$, $G = (V, \Sigma, P, S)$ eine kontextfreie Grammatik in CNF und $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA. Wir wollen nun eine kontextfreie Grammatik $G' = (V', \Sigma, P', S')$ für $L(G) \cap L(M)$ erzeugen, und damit beweisen, dass die kontextfreien Sprachen abgeschlossen unter Schnitt mit regulären Sprachen sind.

Dazu verwenden wir Variablen $V' := \{S'\} \cup \{X_{q,r} : X \in V, q, r \in Q\}$. Die Idee ist, dass $X_{q,r}$ genau die Wörter erzeugt, die sowohl von X erzeugt werden können, als auch im DFA von Zustand q nach r gehen. Formal soll also $L_{G'}(X_{q,r}) = \{w \in L_G(X) : \hat{\delta}(q, w) = r\}$ gelten. Zusätzlich ist S' ein besonderes Startsymbol.

Konstruieren Sie G'. Geben Sie also insbesondere die Produktionen P' an.

Hinweis: G' muss nicht in CNF sein.

Aufgabe H7.3. (*Ge-Stern*) 1+1+1+2+3 Punkte

Sei $G = (V, \Sigma, P, S)$ eine CFG. Wir betrachten Wörter, die sowohl Terminale als auch Nichtterminale enthalten können. Für eine Sprache $L \subseteq (V \cup \Sigma)^*$ definieren wir

$$pre^*(L) = \{\alpha \in (V \cup \Sigma)^* \mid \exists \beta \in L : \alpha \to^* \beta\}$$

als die Menge aller Wörter über $V \cup \Sigma$, aus denen man ein Wort in L ableiten kann. Ein polynomieller Algorithmus A ist bekannt[1], der für einen NFA N über dem Alphabet $V \cup \Sigma$ einen NFA für $pre^*(L(N))$ zurückgibt. Insbesondere ist $pre^*(L)$ regulär, falls L regulär ist.

Geben Sie unter Verwendung von A ein Verfahren an, das

(a) für ein gegebenes Wort $w \in \Sigma^*$ entscheidet, ob $w \in L(G)$.

(b) entscheidet, ob $L(G) = \emptyset$.

(c) für ein gegebenes Nichtterminal $X \in V$ entscheidet, ob X erreichbar ist.

(d) für einen gegebenen DFA M entscheidet, ob $L(G) \subseteq L(M)$.

(e) entscheidet, ob $|L(G)| = \infty$.

[1] Kiefer, S., Kretínský, J., Kucera, A., *Taming the Infinities of Concurrency*, pp. 255-280

Quizaufgabe H7.4. (*Würze* $\in$ *Kürze*) unkorrigiert (5 Punkte)

Der kleine Theodor muss morgen in der Schule einen Grammatiktest schreiben. Eigentlich sollte er lernen, aber er würde viel lieber den Blümchen beim Wachsen zuschauen. Die Grammatik hat er sich auch schon aufgeschrieben, aber leider passt sie nicht auf seinen Spickzettel. Können Sie Ihre moralischen Bedenken überwinden und ihn dabei unterstützen?

Die Grammatik G sei über die folgenden Produktionen gegeben:

$$
\begin{aligned}
S &\to SS \mid AD \mid DB \mid T & E &\to aABb \mid bBAa \mid EabU \\
A &\to aT \mid aaD & T &\to S \mid ETb \mid aAU \mid \varepsilon \\
B &\to Ub \mid BB & U &\to WDW \mid aEb \mid aU \\
C &\to aV \mid \varepsilon & W &\to aB \mid bAUb \mid bWa \\
D &\to Sb \mid b & V &\to aSb \mid ab
\end{aligned}
$$

(a) Eliminieren Sie alle unnützen Symbole aus G mit den aus der Vorlesung bekannten Verfahren. Geben Sie ihren Rechenweg an.

(b) Leider ist G noch nicht klein genug. Geben Sie eine Grammatik G' mit $L(G') = L(G)$ an, die höchstens zwei Produktionen enthält. Beschreiben Sie ihr Vorgehen.

Knobelaufgabe H7.5.

Seien $n \in \mathbb{N}_{>0}$, $\Sigma := \{a, b\}$, $A, B \subseteq \Sigma^n$. Wir schreiben $|x|_A := |\{u : x = uvw, v \in A\}|$, bezeichnen also damit die Anzahl an Vorkommnissen von einem Wort in A in x. Es gilt etwa $|abaaba|_{\{ab,ba\}} = 4$.

Wir wollen herausfinden, ob die Sprache $L' := \{w \in \Sigma^* : |w|_A = |w|_B\}$ regulär ist. Dafür definieren wir $\lambda(wc) := |w|_A - |w|_B$ für $w \in \Sigma^*, c \in \Sigma$. Beachten Sie, dass λ das letzte Zeichen seines Argumentes ignoriert; insbesondere gilt $\lambda(wtz) = \lambda(wt) + \lambda(tz)$ für $w, z \in \Sigma^*$ und $t \in \Sigma^n$. Offensichtlich ist L' genau dann regulär, wenn $L := \lambda^{-1}(0)$ regulär ist.

Zeigen Sie nun, dass L genau dann regulär ist, wenn es keine $u \in A, v \in B$ und $x, y \in \Sigma^*$ gibt, sodass $\lambda(uxu) > 0 > \lambda(vyv)$ gilt.