Theoretische Informatik 2

Sommersemester 2017

Georg Schnitger
Philipp Tertel
Hung The Tran

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Institut für Informatik
AG Theoretische Informatik

Übungsblatt 7

Ausgabe: 06.06.17
Abgabe: 13.06.17

Aufgabe 7.1 ***DL*** ((2+2)+4 Punkte)

Zeigen Sie, dass die folgenden Probleme mit deterministischen Algorithmen auf logarithmischem Speicher[1] gelöst werden können. Eine informelle Beschreibung Ihrer Vorgehensweise ist ausreichend, solange Sie die Idee Ihrer Konstruktion überzeugend beschreiben. Begründen Sie aber sorgfältig, warum Ihr Algorithmus nur logarithmischen Speicherplatz benötigt.

a) Sei $\Sigma = \{0, 1\}$. Für ein Wort $u \in \{0, 1\}^*$ sei $\text{Zahl}(u)$ die natürliche Zahl mit Binärdarstellung u: das niedrigstwertige Bit steht ganz rechts.

   i) Zeigen Sie, dass die Addition zweier Binärzahlen auf logarithmischem Platz ausgeführt werden kann: Für Eingabe $x\#y$ (mit $x, y \in \{0, 1\}^*$) ist die Binärdarstellung z der Summe $\text{Zahl}(x) + \text{Zahl}(y)$ auf dem Ausgabeband auszugeben. Für $n = |x| + |y|$ dürfen nur $\mathcal{O}(\log_2 n)$ Zellen des Arbeitsbands benutzt werden.

   ii) Zeigen Sie, dass die Multiplikation zweier Binärzahlen auf logarithmischem Platz ausgeführt werden kann: Für Eingabe $x\#y$ (mit $x, y \in \{0, 1\}^*$) ist die Binärdarstellung z des Produkts $\text{Zahl}(x) \cdot \text{Zahl}(y)$ auf dem Ausgabeband auszugeben. Für $n = |x| + |y|$ dürfen nur $\mathcal{O}(\log_2 n)$ Zellen des Arbeitsbands benutzt werden.

b) Sei L die Sprache aller wohlgeformten Klammerausdrücke mit den Klammertypen $($, $)$ und $[$, $]$. D. h. L wird erzeugt von der Grammatik $G = (\Sigma, \{S\}, S, P)$ mit dem Alphabet $\Sigma = \{\, (,), [,]\, \}$ und den Produktionen

$$S \to (S) \mid [S] \mid SS \mid \varepsilon.$$

   Zeigen Sie, dass L zur Klasse **DL** gehört.

   Hinweis: Der Fall von zwei Klammertypen ist komplizierter als der eines Klammertyps. Das Beispiel ([ )] zeigt, dass sich die beiden Klammertypen gegenseitig „stören“ können.

Bitte wenden!

[1] Sie können annehmen, dass Sie mit einer konstanten Anzahl von Leseköpfen auf dem Eingabeband und mit konstant vielen Arbeitsbändern mit jeweils eigenem Lese-/Schreibkopf arbeiten können.

Aufgabe 7.2 *NL* (2+4 Punkte)

a) Zeigen Sie, dass **NL** unter den folgenden Operationen abgeschlossen ist: Vereinigung, Durchschnitt, Konkatenation und Stern. Eine kurze informelle Begründung ist völlig ausreichend; Sie sollten aber herausstellen, wann und wie Sie Nichtdeterminismus einsetzen.

   Kommentar: Wir zeigen später, dass **NL** auch unter Komplementbildung abgeschlossen ist!

b) Eine kontextfreie Grammatik $G = (\Sigma, V, S, P)$ heißt genau dann **linear**, wenn für alle Produktionen $A \to \alpha$ in P höchstens eine Variable in der rechten Seite α vorkommen.

   1. Zeigen Sie: Jede reguläre Sprache L besitzt eine lineare Grammatik G mit $L = L(G)$, aber es gibt eine lineare Grammatik G, so dass $L(G)$ nicht regulär ist.
   2. Es gelte $L = L(G)$ für eine lineare Grammatik G. Zeigen Sie, dass L in **NL** liegt.

Aufgabe 7.3 *Das Wortproblem für DFAs und NFAs* (2+4+4 Punkte)

Sei $A = (\Sigma, Q, \delta, q_0, F)$ ein DFA oder NFA. Wie schwierig ist die Simulation von DFAs oder NFAs? Um diese Frage zu beantworten, entwerfen wir zuerst eine „maschinenverständliche“ Kodierung $\langle A \rangle$ des jeweiligen Automaten. Dazu nehmen wir an, dass $\Sigma = \{0, 1\}$, $Q = \{1, 2, \ldots, k\}$ und $q_0 = 1$ gilt; es müssen also nur noch k, F und δ spezifiziert werden. Für $F = \{r_1, \ldots, r_f\}$ setze

$$\langle A \rangle := 1^k \# 1^{r_1} \# 1^{r_2} \# \cdots \# 1^{r_f} \#\# \cdot \prod_{(p,x,q) \in \delta} \left(1^p \# 1^x \# 1^q \# \right)$$

a) Zeigen Sie, dass DFAs mit logarithmischem Speicherplatz simuliert werden können, d.h. zeigen Sie, dass das Wortproblem für DFAs, also die Sprache

$$L_{\text{DFA}} = \{ \langle A \rangle w \ : w \in \{0,1\}^*, \text{ der DFA } A \text{ akzeptiert } w \}$$

   in **DL** liegt.

b) Das Wortproblem für NFAs ist schwieriger: Zeigen Sie, dass die Sprache

$$L_{\text{NFA}} = \{ \langle A \rangle w \ : w \in \{0,1\}^*, \text{ der NFA } A \text{ akzeptiert } w \}$$

   NL-vollständig ist.

   Hinweis: Zeigen Sie zuerst, dass L_{NFA} in **NL** liegt. Dann genügt der Nachweis der LOGSPACE-Reduktion

$$\text{D-Reachability} \leq_{\text{LOG}} L_{\text{NFA}}.$$

c) Zeigen Sie, dass die Sprache

$$E_{\text{DFA}} = \{ \langle A \rangle \ : \ A \text{ ist ein DFA mit } L(A) \neq \emptyset \}$$

   NL-vollständig ist.

   Hinweis: Zeigen Sie zuerst, dass E_{DFA} in **NL** liegt. Dann genügt der Nachweis der LOGSPACE-Reduktion

$$\text{D-Reachability} \leq_{\text{LOG}} E_{\text{DFA}}.$$

   Hier müssen Sie einem gerichteten Graphen G einen DFA A_G über eine LOGSPACE-Transformation zuweisen. Betrachten Sie zuerst den Fall, dass jeder Knoten in G höchstens zwei direkte Nachfolger hat.