Prof. Dr. Jasmin Blanchette
Jannis Limperg
Lydia Kondylidou

Ludwig-Maximilians-Universität München
Institut für Informatik
Abgabe bis 14. Juni 2023, 10:00 Uhr

Lösungsvorschlag zur Übung 8 zur Vorlesung
Theoretische Informatik für Medieninformatiker

TIMI8-1 ***Sprachen einordnen*** (2 Punkte)

Für die i-fache Wiederholung des Worts w schreiben wir $(w)^i$, um Anfang und Ende von w zu markieren. Die Klammern sind daher *nicht* Teil des Alphabets der jeweiligen Sprachen. Die formalen Sprachen L_i, $i = 0, \dots, 2$, seien definiert als

$$\begin{array}{lcll} L_0 & := & \{(a|a(a|b)^i a) \mid i \in \mathbb{N}\} & \subseteq \{a, b\}^* \\ L_1 & := & \{(a(b)^k a(b)^l)^i \mid i, k, l \in \mathbb{N}\} & \subseteq \{a, b\}^* \\ L_2 & := & \{a^i b^{i+1} \mid i \in \mathbb{N}\} & \subseteq \{a, b\}^* \end{array}$$

Bearbeiten Sie die folgenden Arbeitsaufträge für jede der Sprachen L_i.

a) Beweisen oder widerlegen Sie, dass L_i regulär ist.

b) Falls L_i deterministisch kontextfrei ist, beweisen Sie dies. Falls nicht, begründen Sie, warum das so ist.

c) Falls L_i kontextfrei ist, beweisen Sie dies. Falls nicht, begründen Sie, warum das so ist.

Hinweis: Nutzen Sie, dass manche Aussagen direkt aus anderen Aussagen folgen. Um zu beweisen, dass L_i regulär/deterministisch kontextfrei/kontextfrei ist, genügt es, ein geeignetes Konstrukt K_i (Grammatik, Automat oder regulärer Ausdruck) anzugeben und kurz zu begründen, warum $L(K_i) = L_i$ gilt.

LÖSUNGSVORSCHLAG:

- $L_0 = \{(a|a(a|b)^i a) \mid i \in \mathbb{N}\} = L((a|a(a|b)^* a)$ ist regulär (und damit auch kontextfrei und deterministisch kontextfrei), da die Sprache von einem regulären Ausdruck erzeugt wird. Jedes Wort der Sprache ist entweder ein einzelnes a oder es fängt mit a an, enthält dann beliebig viele a's oder b's und hört mit einem weiteren a auf. Der gegebene reguläre Ausdruck erzeugt genau solche Wörter.

- L_1 ist nicht kontextfrei und damit auch nicht regulär oder deterministisch kontextfrei. Beweis mit dem Pumping-Lemma für kontextfreie Sprachen. Gegeben sei eine Zahl $n \in \mathbb{N}$ und wir wählen das Wort $z = (ab^n ab^n)^2 = ab^n ab^n ab^n ab^n \in L_1$ mit $|z| \geq n$.

Betrachte nun Zerlegungen $z = uvwxy$ mit $|vwx| \leq n$ und $|vx| > 0$. Wir zeigen, dass es stets ein $i \in \mathbb{N}$ gibt sodass $uv^i wx^i y \notin L$ ist. Vollständige Fallunterscheidung:

- Das Teilwort vx enthält a. Dann enhält vx genau ein a, da $|vwx| \leq n$ und zwischen den a's jeweils b^n-Blöcke liegen. Das aufgepumpte Wort uv^2wx^2y enthält also 5 a's, aber Wörter in L_1 enthalten immer eine gerade Anzahl a's.

- Das Teilwort vx enthält kein a. Dann liegt vx in maximal zwei benachbarten b-Blöcken, da $|vwx| \leq n$. Liegt vx vollständig in einem b-Block, so ist uv^2wx^2y offensichtlich nicht in L_1. Liegt vx in zwei der b-Blöcke, so gibt es folgende Möglichkeiten:

  * vx liegt im ersten und zweiten Block.

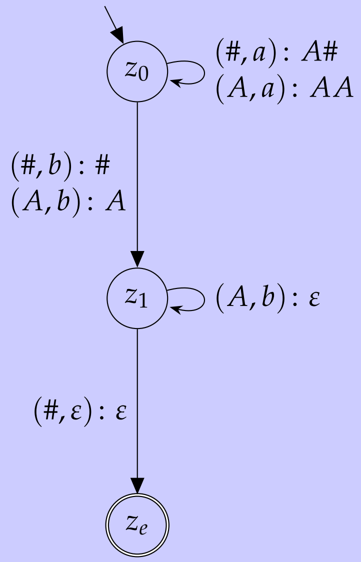
  * vx liegt im zweiten und dritten Block.

  * vx liegt im dritten und vierten Block.

  In jedem dieser Fälle sind in uv^2wx^2y die zwei Hälften des Wortes nicht mehr gleich, sodass das aufgepumpte Wort nicht in L_1 liegen kann.

- L_2 ist nicht regulär. Beweis mit dem Pumping-Lemma für reguläre Sprachen. Sei $n \geq 1$ beliebig. Wir wählen $z = a^n b^{n+1}$ mit $z \in L_2$ und $|z| \geq n$. Sei $z = uvw$ eine beliebige Zerlegung von z mit $|uv| \leq n$ und $|v| \geq 1$. Aus diesen zwei Eigenschaften von uv folgt, dass uv nur aus a's besteht und dass v mindestens ein a enthält. Somit enthält uv^2w mehr als n a's, aber immer noch genau $n+1$ b's, also ist $uv^2w \notin L_2$. L_2 verletzt somit die Pumpingeigenschaft und ist nicht regulär.

  L_2 ist deterministisch kontextfrei (und damit auch kontextfrei), da der folgende DPDA M mit Startsymbol # die Sprache L_2 erkennt:



M schreibt zunächst für jedes gelesene a ein A in den Keller. Der Automat wechselt beim Lesen des ersten b in den Zustand z_1. Von dort aus liest er b-Eingaben, wobei für jede solche Eingabe genau ein A aus dem Keller entfernt wird. Wenn wir das Ende des Wortes erreichen und der Keller nur noch das Startsymbol # enthält, so haben wir n a's und $n+1$ b's gelesen und akzeptieren durch Übergang in z_e.

TIMI8-2 ***Kontextsensitive Sprachen*** (0 Punkte)

Geben Sie eine Typ-1-Grammatik an, welche genau die Sprache

$$L = \{a^n \mid n \in \mathbb{N} \text{ und } n \text{ ist keine Primzahl}\}$$

erzeugt. Erläutern Sie die Funktionsweise Ihrer Grammatik: Wozu dienen die einzelnen Nichtterminale und Produktionen?

Demonstrieren Sie die Funktionsweise Ihrer Grammatik, indem Sie Ableitungen für die Wörter a^0, a^1, a^4 und a^6 angeben.

Hinweis: Wörter $a^n \in L$ mit Mindestlänge 4 können geschrieben werden als $a^{n \cdot m}$ mit $n, m \geq 2$. Eine Herangehensweise, diese Wörter zu erzeugen, ist zunächst a^n zu erzeugen und anschließend dieses Wort $m-1$-Mal zu kopieren. Diese Idee können Sie in eine Typ-1-Grammatik umsetzen.

LÖSUNGSVORSCHLAG:

Wir verwenden den Hinweis und erzeugen a^0 und a^1 direkt und ansonsten $a^{n \cdot m}$ mit $n, m \geq 2$, wobei die Idee ist, zunächst a^n zu erzeugen und anschließend $m-1$ Kopien davon zu machen.

Betrachte die Grammatik $G = (V, \Sigma, P, S)$ mit $V = \{S, N, A, \hat{A}, M, C, \hat{C}\}$, $\Sigma = \{a\}$ und den folgenden Produktionen P:

$$\begin{aligned}
&S \to a \mid \varepsilon \mid NA\hat{A} \\
&N \to NA \mid M \\
&M \to \hat{C} \mid MC \\
&CA \to aAC \\
&C\hat{A} \to a\hat{A} \\
&Ca \to aC \\
&\hat{C}A \to aa\hat{C} \\
&\hat{C}a \to a\hat{C} \\
&\hat{C}\hat{A} \to aa
\end{aligned}$$

Die ersten beiden Produktionen $S \to a$ und $S \to \varepsilon$ erzeugen die Wörter a^0 und a^1. Die Regel $S \to NA\hat{A}$ und die Produktion für Nichtterminal N erzeugen eine Satzform der Form $MA^i\hat{A}$ mit $i \geq 1$. Hierbei stellt $A^i\hat{A}$ quasi das spätere Wort a^n dar, welches noch $m-1$-Mal kopiert wird. Das letzte A ist dabei als $\hat{A}$ geschrieben, um es zu markieren. Offensichtlich gilt $n \geq 2$. Die Produktionen für M führen dann zu einer Satzform der Form $\hat{C}C^jA^i\hat{A}$ mit $j \geq 0$. Die Anzahl an C und $\hat{C}$ gibt an, wie viele zusätzliche Kopien von a^n erstellt werden (d.h. im Fall $j = 0$ wird a^n trotzdem einmal kopiert und damit ist $m \geq 2$). Die Verwendung von $\hat{C}$ ganz links dient dazu das linkeste C zu markieren.

Die Produktionen $CA \to aAC$ und $Ca \to aC$ schieben ein C von links nach rechts, wobei für jedes A ein a erzeugt wird, aber kleine a selbst gleich bleiben. Kommt ein C am Ende rechts an, so ist $C\hat{A}$ das Suffix. Mit $C\hat{A} \to a\hat{A}$ wird ein a für das letzte A erzeugt und C gelöscht. Das linkeste C ist durch $\hat{C}$ repräsentiert und wird genauso von links nach rechts geschoben, und erstellt für jedes A ein a, zudem werden alle vorhandenen A nun auch in a konvertiert (Regel $\hat{C}A \to aa\hat{C}$ und für das rechteste die Regel $\hat{C}\hat{A} \to aa$). Danach ist die Satzform von der Form $a^{n \cdot m}$. G ist wirklich Typ-1, da es keine verkürzenden Regeln gibt.

Beispielableitungen:

- $\underline{S} \Rightarrow \varepsilon = a^0$
- $\underline{S} \Rightarrow a = a^1$
- $\underline{S} \Rightarrow \underline{N}A\hat{A} \Rightarrow \underline{M}A\hat{A} \Rightarrow \underline{\hat{C}A}\hat{A} \Rightarrow aa\underline{\hat{C}\hat{A}} \Rightarrow aaaa$

- Variante 1 für a^6 $(6 = 3 \cdot 2)$: $\underline{S} \Rightarrow \underline{N}A\hat{A} \Rightarrow \underline{N}AA\hat{A} \Rightarrow \underline{M}AA\hat{A} \Rightarrow \underline{\hat{C}A}A\hat{A} \Rightarrow aa\underline{\hat{C}A}\hat{A} \Rightarrow aaaa\underline{\hat{C}\hat{A}} \Rightarrow aaaaaa$
- Variante 2 für a^6 $(6 = 2 \cdot 3)$: $\underline{S} \Rightarrow \underline{N}A\hat{A} \Rightarrow \underline{M}A\hat{A} \Rightarrow \underline{M}CA\hat{A} \Rightarrow \hat{C}\underline{CA}\hat{A} \Rightarrow \hat{C}aA\underline{C\hat{A}} \Rightarrow \underline{\hat{C}a}Aa\hat{A} \Rightarrow a\underline{\hat{C}A}a\hat{A} \Rightarrow aaa\underline{\hat{C}a}\hat{A} \Rightarrow aaaa\underline{\hat{C}\hat{A}} \Rightarrow aaaaaa$