Prof. Dr. Jasmin Blanchette
Jannis Limperg
Luca Maio

Ludwig-Maximilians-Universität München
Institut für Informatik
Abgabe bis 26. Juni 2024, 10:00 Uhr

Übung 9 zur Vorlesung

Formale Sprachen und Komplexität

Wenn Sie Automaten angeben, tun Sie dies immer in Form eines Zustandsgraphen. Andere Formen der Darstellung (z.B. als Liste von Übergängen) werden nicht gewertet, da sie sehr viel aufwändiger zu korrigieren sind. Vergessen Sie nicht, im Zustandsgraph Start- und Endzustände zu markieren.

Erlaubte Konstrukte für

LOOP-Programme	WHILE-Programme	GOTO-Programme
$x_i := x_j + c$	$x_i := x_j + c$	$M_k : x_i := x_j + c$
$x_i := x_j - c$	$x_i := x_j - c$	$M_k : x_i := x_j - c$
$x_i := c$	$x_i := c$	$M_k : x_i := c$
$P_1; P_2$	$P_1; P_2$	$P_1; P_2$
LOOP x_i **DO** P **END**	**LOOP** x_i **DO** P **END**	$M_k :$ **GOTO** M_j
	WHILE $x_i \neq 0$ **DO** P **END**	$M_k :$ **IF** $x_i = c$ **THEN GOTO** M_j
		$M_k :$ **HALT**

Dabei darf jede Marke M_k nur einmal vorkommen

FSK9-1 ***WHILE-, LOOP- und GOTO-Programme*** (2 Punkte)

In den folgenden Teilaufgaben sollen Sie jeweils ein Programm angeben. Verwenden Sie dabei nur die erlaubten Anweisungen aus der obigen Tabelle. Kommentieren Sie außerdem Ihre Programme, sodass klar wird, wie sie funktionieren sollen. Nicht kommentierte Programme werden eventuell nicht gewertet.

a) Schreiben Sie ein GOTO-Programm, das der Variablen x_0 den Wert von $3 \cdot x_1 \cdot x_2$ zuweist.

b) Schreiben Sie ein WHILE-Programm, das die folgende Funktion berechnet:

$$f : \mathbb{N}^2 \to \mathbb{N}, \quad f(x, y) = 5 * (x + y)$$

c) Schreiben Sie ein LOOP-Programm, das den folgenden Pseudocode implementiert:

$$\textbf{FOR } x_i := 0 \textbf{ TO } x_1 \textbf{ DO } x_0 := x_0 + x_i \textbf{ END}$$

FSK9-2 ***LOOPY-Programme*** (0 Punkte)

Wir betrachten LOOPY-Programme, die aus folgenden Anweisungen bestehen:

- $x_i := x_j + c$, $x_i := x_j - c$ und $P_1;\ P_2$ wie bei LOOP-Programmen.
- $(P_1 \mid P_2)$, wobei P_1 und P_2 LOOPY-Programme sind. Diese Anweisung führt nichtdeterministisch entweder P_1 oder P_2 aus.
- **LOOPY** P **END**. Diese Anweisung führt das LOOPY-Programm P nichtdeterministisch 0 oder mehr Male aus.

Beispielsweise führt das LOOPY-Programm

$$\textbf{LOOPY}\ (x_0 := x_0 + 2 \mid x_0 := x_0 - 1)\ \textbf{END}$$

beliebig oft entweder die Anweisung $x_0 := x_0 + 2$ oder die Anweisung $x_0 := x_0 - 1$ aus, wobei es in jeder Iteration eine andere Entscheidung treffen kann. Der Wert von x_0 am Ende des Programms kann also jede natürliche Zahl sein.

a) Die Semantik eines LOOPY-Programms können wir als eine Relation $\xrightarrow[\text{LOOPY}]{}$ definieren. Diese Relation ist ähnlich der für WHILE-Programme, aber LOOPY-Programme können nichtdeterministisch verschiedene Ergebnisse haben. Dementsprechend kann ein LOOPY-Programm P mit einer Variablenbelegung ρ mehrere Nachfolge-Variablenbelegungen ρ' und mehrere Nachfolge-Programme P' haben, je nachdem, welche nichtdeterministischen Entscheidungen das Programm trifft. Für alle diese ρ' und P' soll gelten:

$$(\rho, P) \xrightarrow[\text{LOOPY}]{} (\rho', P')$$

Definieren Sie die Semantik von LOOPY-Programmen.

b) Zeigen Sie, dass mit Ihrer Semantik gilt:

$$(\{x_0 \mapsto 0\}, \textbf{LOOPY}\ (x_0 := x_0 + 2 \mid x_0 := x_0 - 1)\ \textbf{END}) \xrightarrow[\text{LOOPY}]{}{}^{*} (\{x_0 \mapsto 1\}, \varepsilon)$$

c) Eine Funktion $f \colon \mathbb{N}^k \to \mathbb{N}$ ist LOOPY-berechenbar, wenn es ein LOOPY-Programm P gibt, sodass es für alle $n_1, \ldots, n_k \in \mathbb{N}$ eine Variablenbelegung ρ gibt, für die gilt:

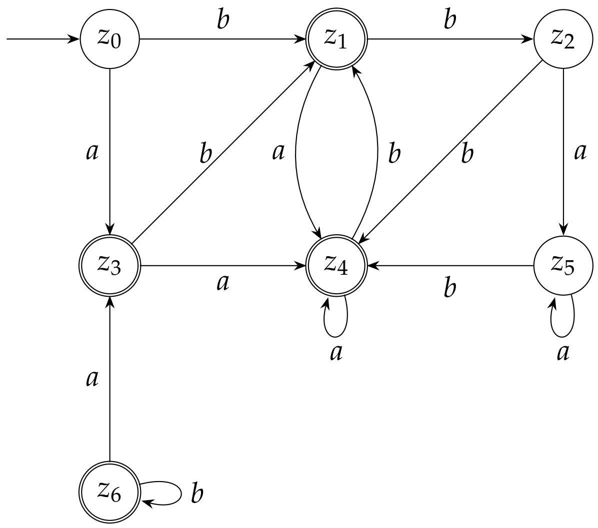
$$(\{x_1 \mapsto n_1, \ldots, x_k \mapsto n_k\}, P) \xrightarrow[\text{LOOPY}]{}{}^{*} (\rho, \varepsilon)$$

und $\rho(x_0) = f(n_1, \ldots, n_k)$.

Zeigen Sie: Jede Funktion, die LOOP-berechenbar ist, ist auch LOOPY-berechenbar.

FSK9-3 ***Algorithmen Wiederholung*** (2 Punkte)

a) Minimieren Sie den folgenden DFA. Verwenden Sie die tabellarische Variante des Algorithmus zur Minimierung von DFAs aus der Vorlesung (nicht die graphische Variante und nicht den Algorithmus von letztem Jahr). Geben Sie die Partitionstabelle und den minimalen DFA an.



Sei G die Grammatik $(\{A_1, A_2, A_3, A_4, A_5\}, \{\$, \#\}, P, A_1)$ mit

$$
\begin{aligned}
P = \{ & A_1 \to A_2A_5 \mid A_3A_5, \\
& A_2 \to A_4A_3 \mid \$, \\
& A_3 \to A_2A_5, \\
& A_4 \to \$, \\
& A_5 \to A_5A_1 \mid \#\}
\end{aligned}
$$

Prüfen Sie mit dem CYK-Algorithmus, ob das Wort \$##\$## in $L(G)$ ist. Erstellen Sie dazu die entsprechende Tabelle des Algorithmus und erklären Sie anhand der Tabelle, ob das Wort in $L(G)$ ist.

FSK9-4 ***Kontextfreie Sprachen*** (0 Punkte)

Sei L die formale Sprache aller Wörter $w \in \{a, b\}^*$, sodass w in der zweiten Hälfte mindestens ein b enthält:

$$L = \{ubv \mid u, v \in \{a, b\}^*, |u| > |v|\}$$

a) Geben Sie eine kontextfreie Grammatik an, die L erzeugt.

b) Geben Sei einen Kellerautomaten an, der L akzeptiert. Entsprechend der Definition aus der Vorlesung soll Ihr Kellerautomat durch leeren Keller akzeptieren.

Hinweis: Nutzen Sie den Nichtdeterminismus des Automaten aus.