Formale Sprachen und Komplexität, SS 18, Prof. Dr. Volker Heun

Übungsblatt 3

Abgabe: bis Fr. 14.05.2018 8 Uhr

Institut für Informatik **Ludwig-Maximilians-Universität München**
Prof. Dr. Volker Heun

Formale Sprachen und Komplexität, SS 18
Übungsblatt 3

Abgabe: bis Fr. 14.05.2018 8 Uhr

Nach Bearbeitung dieses Übungsblattes sollten Sie:

	Check
Den Konfigurationsablauf eines deterministischen Automaten zu einem gegebenen Wort angeben können.	
Aus einem deterministischen Automaten eine reguläre Grammatik konstruieren können.	
Aus einer regulären Grammatik einen nichtdeterministischen Automaten konstruieren können.	
Einen nichtdeterministischen Automaten in einen Deterministischen umwandeln können.	
Begründen können, warum ein Automat (vollständig) deterministisch bzw. nichtdeterministisch ist.	

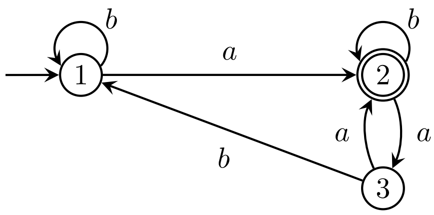
Diese Ziele sind wichtige Hinweise für die Klausur!

Aufgabe 3-1 schriftlich bearbeiten
Endliche Automaten

Sei $M = (Z, \Sigma, \delta, 1, E)$ der endliche Automat mit $Z = \{1, 2, 3\}$ und $\Sigma = \{a, b\}$ und $E = \{2\}$ und

$$\begin{array}{lllllllll} \delta(1,a) & = & 2 & \delta(2,a) & = & 3 & \delta(3,a) & = & 2 \\ \delta(1,b) & = & 1 & \delta(2,b) & = & 2 & \delta(3,b) & = & 1 \end{array}$$

M als Zustandsgraph:



Lässt man den Automaten mit einem Eingabewort „laufen“, kann man sein Verhalten durch die *Konfigurationen* beschreiben, die er dabei durchläuft. Eine *Konfiguration* ist ein Paar (aktueller Zustand, noch zu lesender Rest des Eingabeworts). Zum Beispiel durchläuft M für das Eingabewort bba die Konfigurationen $(1, bba) \vdash (1, ba) \vdash (1, a) \vdash (2, \varepsilon)$ und akzeptiert somit das Wort bba.

a) Geben Sie die Konfigurationsfolge an, die M mit dem Eingabewort *ababb* durchläuft. Akzeptiert der Automat dieses Wort?

Lösungsvorschlag:

$(1, ababb) \quad \vdash \quad (2, babb) \quad \vdash \quad (2, abb) \quad \vdash \quad (3, bb) \quad \vdash \quad (1, b) \quad \vdash \quad (1, \varepsilon)$

Da 1 kein Endzustand ist, akzeptiert der Automat dieses Wort nicht.

Achtung:
Der Automat hält **nicht** an, sobald er einen „Endzustand“ erreicht, sondern sobald das restliche Eingabewort leer ist.

Wenn der Zustand, den er dann erreicht hat, ein Endzustand ist, akzeptiert er das Wort. Wenn dieser Zustand kein Endzustand ist, akzeptiert er das Wort nicht.

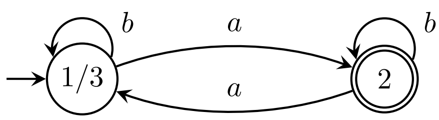
b) Welche Sprache akzeptiert M? (ohne Beweis)

Lösungsvorschlag:

$T(M) =$ die Menge aller Wörter $\in \{a, b\}^*$,
in denen eine ungerade Anzahl von a vorkommt.

Das ist dem gegebenen Automaten M nur schwer anzusehen.

Vorgriff auf späteren Stoff:
Wenn man den Minimalautomaten zu M konstruiert, verschmelzen die Zustände 1 und 3, und es entsteht:

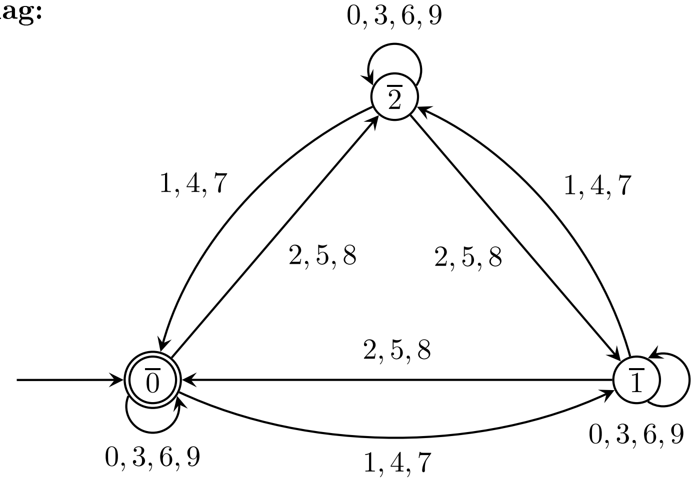


Hier erkennt man die akzeptierte Sprache viel einfacher.

Aufgabe 3-2 Endliche Automaten

Geben Sie einen endlichen deterministischen Automaten an, der die Sprache
$L = \{\omega \in \{0, 1, 2, \ldots, 9\}^* \mid \omega \textit{ als Dezimalzahl interpretiert ist durch drei teilbar}\}$ akzeptiert. In diesem Fall darf ein ω aus L führende Nullen enthalten.

Lösungsvorschlag:



Aufgabe 3-3 schriftlich bearbeiten
(Nicht)deterministische endliche Automaten und reguläre Grammatiken

a) Geben Sie einen deterministischen endlichen Automaten M an, der die Menge der ungeraden natürlichen Zahlen in Dezimaldarstellung akzeptiert.

Lösungsvorschlag:

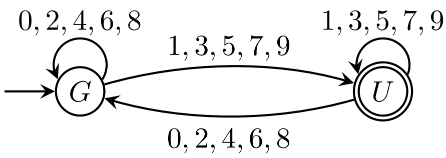
Vorüberlegung:
Die Parität hängt in Dezimaldarstellung nur von der letzten Ziffer ab.
Also zwei Zustände G (letzte Ziffer gerade) und U (letzte Ziffer ungerade).

Endlicher Automat:

$M = (Z, \Sigma, \delta, G, E)$ mit

$Z = \{G, U\}$ $\quad \Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ $\quad E = \{U\}$

δ:



Lösungsvorschlag:

In den weiteren Teilaufgaben verwendete Konstruktionen aus dem Lehrbuch:

determ. endlicher Automat	$\longrightarrow$	Grammatik	S. 22
Zustand A	wird zu	Variable A	
δ-Übergang $A \xrightarrow{a} B$	wird zu	Produktion $A \to aB$	
δ-Übergang $A \xrightarrow{a} \circledcirc B$	wird zu	Produktionen $A \to aB$, $A \to a$	

Grammatik	$\longrightarrow$	nichtdeterm. endlicher Automat	S. 27
Variable A	wird zu	Zustand A. Ein Extrazustand X	
Produktion $A \to aB$	wird zu	δ-Übergang $A \xrightarrow{a} B$	
Produktion $A \to a$	wird zu	δ-Übergang $A \xrightarrow{a} \circledcirc X$	

nichtdeterm. endlicher Automat	$\longrightarrow$	determ. endlicher Automat	S. 24
Teilmenge der Zustandsmenge	wird zu	Zustand	

b) Konstruieren Sie aus M eine Grammatik G vom Typ 3 (regulär) mit $L(G) = T(M)$. Verwenden Sie die Konstruktion aus dem Lehrbuch.

Lösungsvorschlag:

Wir wollen eine Variable G nennen. Deshalb nennen wir die Grammatik hier $\mathbb{G}$ statt G.

$\mathbb{G} = (V, \Sigma, P, G)$ mit

$V = \{G, U\}$ $\qquad \Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$$
P: \quad
\begin{array}{lllllllll}
G & \to & 0G & G & \to & 1U & G & \to & 1 \\
G & \to & 2G & G & \to & 3U & G & \to & 3 \\
G & \to & 4G & G & \to & 5U & G & \to & 5 \\
G & \to & 6G & G & \to & 7U & G & \to & 7 \\
G & \to & 8G & G & \to & 9U & G & \to & 9 \\
U & \to & 0G & U & \to & 1U & U & \to & 1 \\
U & \to & 2G & U & \to & 3U & U & \to & 3 \\
U & \to & 4G & U & \to & 5U & U & \to & 5 \\
U & \to & 6G & U & \to & 7U & U & \to & 7 \\
U & \to & 8G & U & \to & 9U & U & \to & 9
\end{array}
$$

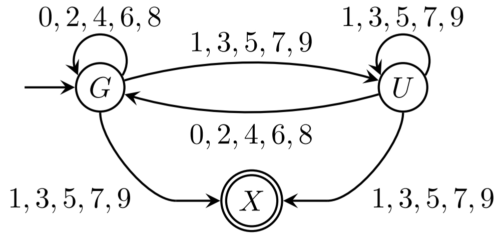
c) Konstruieren Sie aus G einen nichtdeterministischen endlichen Automaten N mit $T(N) = L(G)$. Verwenden Sie die Konstruktion aus dem Lehrbuch.

Lösungsvorschlag:

$N = (Z, \Sigma, \delta, S, E)$ mit

$Z = \{G, U, X\}$ $\qquad \Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ $\qquad S = \{G\}$ $\qquad E = \{X\}$

δ:



d) Konstruieren Sie aus N einen deterministischen endlichen Automaten M' mit $T(M') = T(N)$.

Lösungsvorschlag:

Offensichtlich ist der Zustand $\{G\}$ der Startzustand von M'. Ausgehend vom Startzustand bestimmen wir für alle Eingabewörter die Menge der möglichen Folgezustände. Diese Mengen sind ebenfalls Zustände von M'. Wir bestimmen für alle neuen Zustände die Mengen der Folgezustände für alle Wörter. So verfahren wir weiter, bis keine Zustände von M' mehr dazu kommen. u Steht für alle ungeraden Zahlen und g für alle geraden.

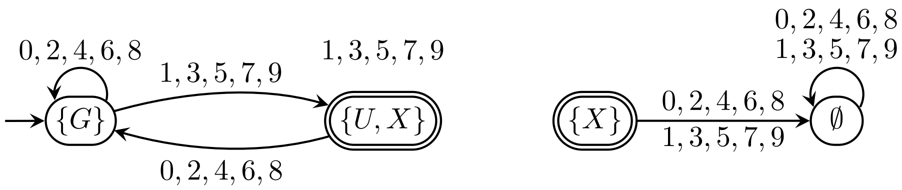
Zustand von M'	Eingabewort	Folgezustand von M'
$\{G\}$	g	$\{G\}$
$\{G\}$	u	$\{U, X\}$
$\{U, X\}$	u	$\{U, X\}$
$\{U, X\}$	g	$\{G\}$

$M' = (Z, \Sigma, \delta, z_0, E)$ mit

$Z = \{\,\emptyset,\ \{G\},\ \{X\},\ \{U, X\}\,\}$

$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ $\qquad z_0 = \{G\}$ $\qquad E = \{\,\{X\},\ \{U, X\}\,\}$

δ:



Aufgabe 3-4 Endliche Automaten

Literale für `int`-Konstanten im Dezimalsystem bestehen in Java aus einem optionalen Vorzeichen gefolgt von einer nichtleeren Folge von Dezimalziffern ohne führende Nullen: 0 und +0 und −0 sind erlaubt, aber 00 und +08 und −009 nicht (vgl. Aufgabe 1-1).

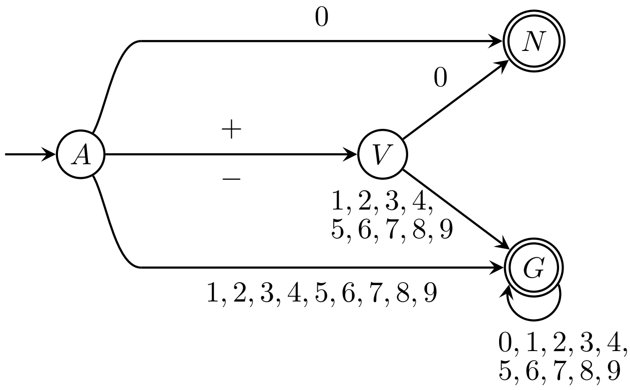
Geben Sie einen endlichen Automaten M an, der diese formale Sprache akzeptiert.

Ist Ihr Automat nichtdeterministisch, deterministisch oder vollständig deterministisch?

Lösungsvorschlag:

Alphabet:	$\Sigma = \{+, -, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
Zustände:	A Startzustand
	V Vorzeichen gelesen
	N Null am Anfang gelesen
	G Ganze Zahl in zulässiger Form gelesen

Erste Idee für die Zustandsüberführungsfunktion:



Der endliche Automat mit dieser Zustandsüberführungsfunktion ist **deterministisch**, jedoch nicht **vollständig deterministisch**: Kein Paar (Zustand,Terminalsymbol) hat mehr als einen Nachfolgezustand, manche haben jedoch gar keinen, z.B. ist $\delta(V, +) = \emptyset$.

Hinweis:

In einem *vollständig deterministischen* endlichen Automaten hat jedes Paar (Zustand,Terminalsymbol) genau einen Nachfolgezustand, also **höchstens und mindestens** einen.

Schwächt man die Definition ab zu „**höchstens** einen Nachfolgezustand“, ist der endliche Automat *deterministisch*.

Für nichtvollständige deterministische endliche Automaten funktionieren aber einige Verfahren nicht, zum Beispiel das Verfahren zur Konstruktion des Minimalautomaten.

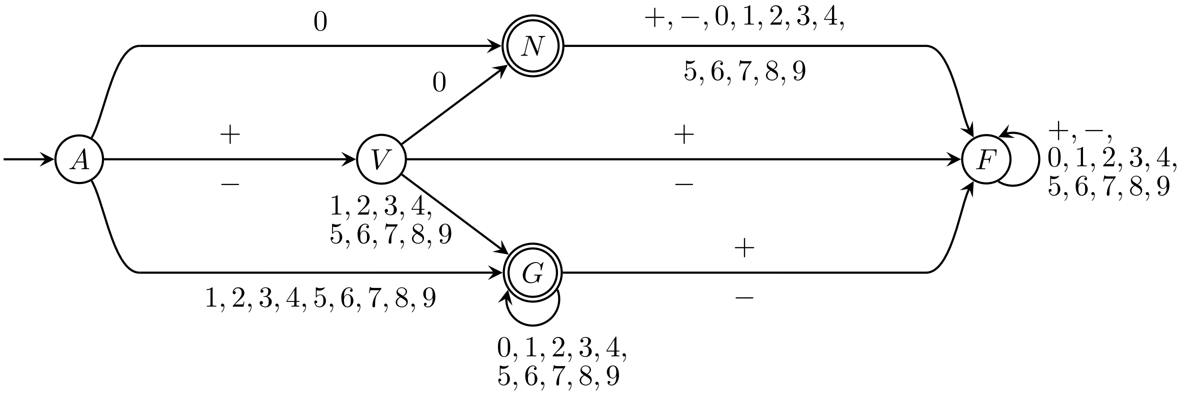
Deshalb fordert man gelegentlich die Vollständigkeit, obwohl nichtvollständige deterministische Automaten häufig deutlich übersichtlicher sind.

Lösungsvorschlag:

Lösung mit einem vollständigen deterministischen endlichen Automaten:

$M = (Z, \Sigma, \delta, A, E)$ mit
$Z = \{A, V, N, G, F\}$ und $\Sigma = \{+, -, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ und $E = \{N, G\}$ und

δ:



F ist ein sogenannter „Fangzustand“ für alle „falschen“ Wörter.

Beispiele:

$(A, +18)$	$\vdash$	$(V, 18)$	$\vdash$	$(G, 8)$	$\vdash$	(G, ε)	akzeptiert	weil $G \in E$
$(A, +0)$	$\vdash$	$(V, 0)$	$\vdash$	(N, ε)			akzeptiert	weil $N \in E$
$(A, +08)$	$\vdash$	$(V, 08)$	$\vdash$	$(N, 8)$	$\vdash$	(F, ε)	nicht akzeptiert	weil $F \notin E$