

## Zusammenhang zwischen deterministischen endlichen Automaten und regulären Grammatiken

### Rückblick

Mit den deterministischen endlichen Automaten (DEA) und den Grammatiken haben wir zwei Konzepte kennengelernt, mit denen formale Sprachen eindeutig beschrieben werden können. Mithilfe von Grammatiken werden formale Sprachen erzeugt - man spricht von einem Erzeugendensystem. Mithilfe eines DEA wird dagegen überprüft, ob ein Wort zu einer bestimmten formalen Sprache gehört – man spricht von einem Akzeptor oder Erkennendensystem. Die Grammatiken und damit die Sprachen, die sie erzeugen, haben wir je nach der Komplexität ihrer Struktur in drei verschiedene Kategorien eingeteilt. Die rechtslinearen bzw. regulären Grammatiken, die kontextfreien und die kontextsensitiven Grammatiken bzw. Sprachen. Deterministische endliche Automaten haben wir nur für formale Sprachen der ersten Kategorie konstruiert, also für Sprachen, die mithilfe einer regulären Grammatik erzeugt werden können.

Es liegt daher die *Vermutung* nahe, dass sich die regulären Grammatiken und die deterministischen endlichen Automaten der gleichen Menge formaler Sprachen zuordnen lassen: den regulären Sprachen. Das hieße, für eine formale Sprache lässt sich genau dann ein DEA konstruieren, wenn sie von einer regulären Grammatik erzeugt werden kann und umgekehrt.

Diese Vermutung wäre bewiesen, wenn wir zeigen, dass sich jeder DEA systematisch in eine reguläre Grammatik umwandeln lässt und umgekehrt. Daher betrachten wir den Zusammenhang zwischen deterministischen endlichen Automaten und regulären Grammatiken im Folgenden genauer.

### Vom DEA zur regulären Grammatik

Schauen wir uns zunächst an einem Beispiel an, wie sich ein DEA systematisch in eine reguläre Grammatik transformieren lässt: Für Passwörter gibt es oft Vorgaben, die sicherstellen sollen, dass das Passwort nicht zu leicht erraten werden kann. Der DEA in Abbildung 1 akzeptiert die formale Sprache der Passwörter, die mindestens eine Ziffer und ein Sonderzeichen enthalten. Zur Vereinfachung steht *b* für einen beliebigen Buchstaben, *z* für eine beliebige Ziffer und *s* für ein Sonderzeichen. Neben dem Automaten finden Sie die zugehörige rechtslineare Grammatik  $G_{\text{Pass}}$ .

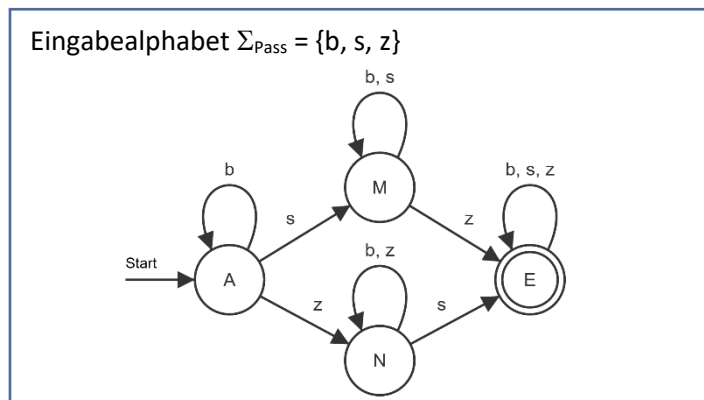


Abbildung 1: DEA, der alle Passwörter akzeptiert, die mindestens eine Zahl und ein Sonderzeichen enthalten

Grammatik  $G_{\text{Pass}} = (V, \Sigma, P, A)$  mit  
 Nichtterminale  $V = \{A, M, N, E\}$   
 Terminale  $\Sigma = \{b, s, z\}$   
 Startsymbol  $A$   
 Produktionen  $P$ :

$A \rightarrow bA \mid sM \mid zN$   
 $M \rightarrow bM \mid sM \mid zE \mid z$   
 $N \rightarrow bN \mid zN \mid sE \mid s$   
 $E \rightarrow bE \mid sE \mid zE \mid b \mid s \mid z$

### Aufgabe 1:

- Geben Sie für die Wörter *bsbzz* und *zs* sowohl die Konfigurationsfolge des  $\text{DEA}_{\text{Pass}}$  als auch die Ableitung aus dem Startsymbol  $A$  für die Grammatik  $G_{\text{Pass}}$  an.
- Vergleichen Sie den  $\text{DEA}_{\text{Pass}}$  mit der Grammatik  $G_{\text{Pass}}$ . Stellen Sie Regeln auf, nach denen ein DEA systematisch in eine rechtslineare Grammatik übersetzt werden kann.

**Aufgabe 2:** Abbildung 2 zeigt den  $\text{DEA}_{\text{Ganz}}$ . Darunter finden Sie die zugehörige rechtslineare Grammatik  $G_{\text{Ganz}}$ .

- Geben Sie für drei Beispiele sowohl die Konfigurationsfolge für den  $\text{DEA}_{\text{Ganz}}$  als auch die Ableitung aus dem Startsymbol  $S$  der regulären Grammatik  $G_{\text{Ganz}}$  an.
- Beschreiben Sie die Sprache, welche der  $\text{DEA}_{\text{Ganz}}$  akzeptiert, in Worten.
- Überprüfen Sie die Regeln, die Sie in Aufgabe 1b) aufgestellt haben, an diesem Beispiel. Ergänzen Sie ggf. weitere Regeln, um beliebige deterministische endliche Automaten systematisch in rechtslineare Grammatiken zu transformieren.

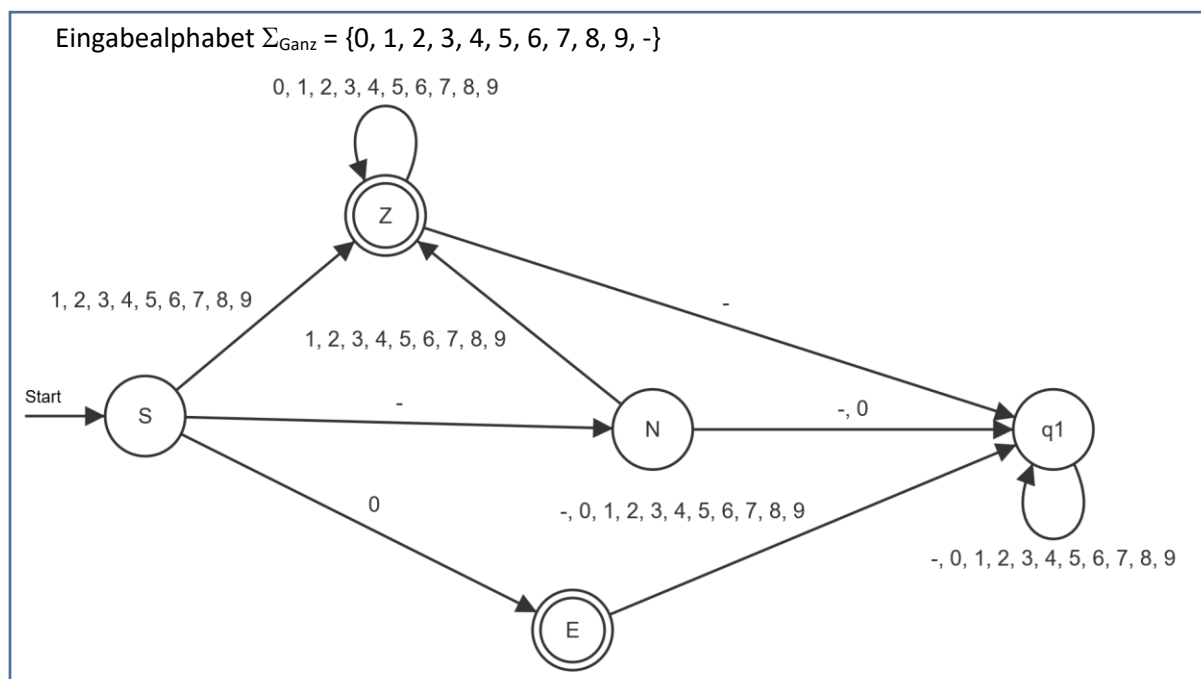


Abbildung 2:  $\text{DEA}_{\text{Ganz}}$  zu Aufgabe 2

$G_{\text{Ganz}} = (V, \Sigma, P, S)$  mit

Nichtterminale  $V = \{S, N, Z\}$

Terminale  $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -\}$

Startsymbol  $S$

Produktionen  $P$ :

$S \rightarrow -N \mid 1Z \mid 2Z \mid 3Z \mid 4Z \mid 5Z \mid 6Z \mid 7Z \mid 8Z \mid 9Z \mid 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$N \rightarrow 1Z \mid 2Z \mid 3Z \mid 4Z \mid 5Z \mid 6Z \mid 7Z \mid 8Z \mid 9Z \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$Z \rightarrow 0Z \mid 1Z \mid 2Z \mid 3Z \mid 4Z \mid 5Z \mid 6Z \mid 7Z \mid 8Z \mid 9Z \mid 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

**Aufgabe 3:** In der Oberstufe werden die Noten in Form von Punkten zwischen 00 und 15 vergeben. Abbildung 3 zeigt einen DEA, der die formale Sprache der Notenpunkte in zweistelliger Schreibweise akzeptiert.

Transformieren Sie den DEA systematisch in eine reguläre Grammatik.

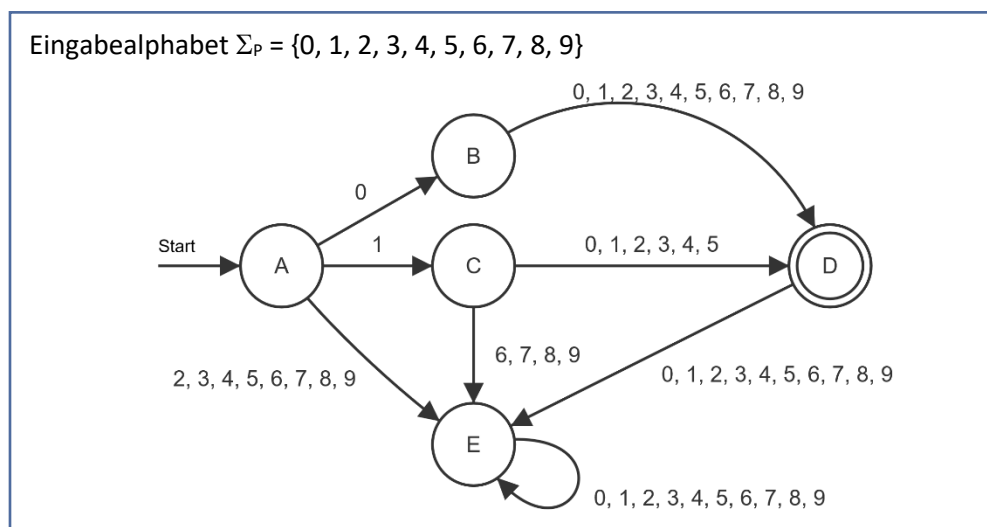


Abbildung 3: DEA, der die zweistelligen Notenpunkte der Oberstufe akzeptiert

**Aufgabe 4:** Die Adresse einer Zelle in einer Tabellenkalkulation wird durch eine Buchstabenkombination gefolgt von einer Zahl angegeben. Beispiele wären A1, BD2946, D312, CZ5.

- Konstruieren Sie einen DEA, der die formale Sprache der Zelladressen einer Tabellenkalkulation akzeptiert.
- Transformieren Sie den DEA systematisch in eine reguläre Grammatik.

### Von der regulären Grammatik zum DEA

Schauen wir uns nun an, ob sich die Regeln, die wir zum Transformieren eines DEA in eine reguläre Grammatik gefunden haben, auch umkehren lassen. Wir betrachten dazu wieder ein Beispiel.

Die Grammatik  $G_{\text{Euro}}$  erzeugt eine formale Sprache gültiger Geldbeträge.

$G_{\text{Euro}} = (V, \Sigma, P, S)$  mit

Nichtterminale  $V = \{A, B, C, D, S, Z\}$

Terminale  $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \text{Komma}, \text{€}\}$

Startsymbol  $S$

Produktionen  $P$ :

$S \rightarrow 0D \mid 1Z \mid 2Z \mid 3Z \mid 4Z \mid 5Z \mid 6Z \mid 7Z \mid 8Z \mid 9Z$

$Z \rightarrow 0Z \mid 1Z \mid 2Z \mid 3Z \mid 4Z \mid 5Z \mid 6Z \mid 7Z \mid 8Z \mid 9Z \mid ,A$

$D \rightarrow ,A$   
 $A \rightarrow 0B \mid 1B \mid 2B \mid 3B \mid 4B \mid 5B \mid 6B \mid 7B \mid 8B \mid 9B$   
 $B \rightarrow 0C \mid 1C \mid 2C \mid 3C \mid 4C \mid 5C \mid 6C \mid 7C \mid 8C \mid 9C$   
 $C \rightarrow \epsilon$

#### Aufgabe 5:

- Analysieren Sie die Grammatik  $G_{\text{Euro}}$  und beschreiben Sie die formale Sprache gültiger Geldbeträge in Worten.
- Transformieren Sie die Grammatik  $G_{\text{Euro}}$  systematisch in einen passenden DEA.
- Erläutern Sie Ihr Vorgehen und formulieren Sie allgemeine Regeln zur systematischen Transformation einer regulären Grammatik in einen DEA.

**Aufgabe 6:** Die Grammatik  $G_{\text{Nat}}$  erzeugt die formale Sprache der natürlichen Zahlen. Transformieren Sie die Grammatik  $G_{\text{Nat}}$  systematisch in einen passenden DEA.

$G_{\text{Nat}} = (V, \Sigma, P, S)$  mit

Nichtterminale  $V = \{S, Z\}$

Terminale  $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Startsymbol  $S$

Produktionen  $P$ :

$S \rightarrow 0 \mid 1Z \mid 2Z \mid 3Z \mid 4Z \mid 5Z \mid 6Z \mid 7Z \mid 8Z \mid 9Z$

$Z \rightarrow 0Z \mid 1Z \mid 2Z \mid 3Z \mid 4Z \mid 5Z \mid 6Z \mid 7Z \mid 8Z \mid 9Z \mid 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

#### Zusammenfassung

Die vorhergehenden Betrachtungen haben an verschiedenen Beispielen gezeigt, dass ein deterministischer endlicher Automat systematisch in eine reguläre Grammatik transformiert werden kann und umgekehrt. Es handelt sich daher um zwei verschiedene Modelle, welche die Menge der regulären Sprachen aus unterschiedlichen Blickwinkeln beschreiben, aber die gleichen Grenzen aufweisen.

In Aufgabe 7 soll der Zusammenhang zwischen den deterministischen endlichen Automaten und den regulären Grammatiken noch einmal allgemein herausgearbeitet und zusammengefasst werden.

**Aufgabe 7:** Vergleichen Sie allgemein das Modell der regulären Grammatiken mit dem der deterministischen endlichen Automaten, indem Sie Gemeinsamkeiten beschreiben und Analogien herstellen.

#### Exkurs: Nichtdeterministische endliche Automaten

Im Kapitel „Grammatiken zur Beschreibung formaler Sprachen“ haben Sie die formale Sprache der Halbstrukturformeln für Alkene kennengelernt. Die Grammatik  $G_{\text{AlkeneReg}}$  ist eine reguläre Grammatik, welche diese Sprache erzeugt.

$G_{\text{AlkeneReg}} = (V, \Sigma, P, A)$  mit

Nichtterminale  $V = \{A, B, D, E, F, J, G, K, L, M, N, O, R, S, T\}$

Terminale  $\Sigma = \{C, H, 2, 3, =, -\}$

Startsymbol  $A$

Produktionen  $P$ :

$A \rightarrow CB$

$E \rightarrow = F$

$K \rightarrow 2$

$M \rightarrow CN$

$R \rightarrow CS$

$B \rightarrow HD$

$F \rightarrow CG \mid CJ$

$J \rightarrow HL$

$N \rightarrow HO$

$S \rightarrow HT$

$D \rightarrow 2E$

$G \rightarrow HK$

$L \rightarrow - M \mid -R$

$O \rightarrow 2L$

$T \rightarrow 3$

Wenn wir die Grammatik  $G_{\text{AlkeneReg}}$  systematisch in einen endlichen Automaten transformieren, erhalten wir den Automaten in Abbildung 4.

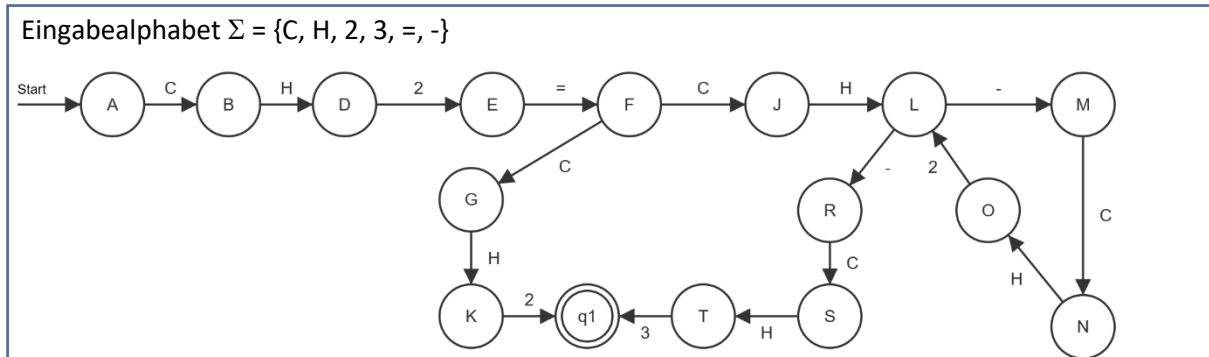


Abbildung 4: Endlicher Automat zu der Grammatik  $G_{\text{AlkeneReg}}$

**Aufgabe 8:** Überprüfen Sie, ob es sich bei dem endlichen Automaten aus Abbildung 4 um einen deterministischen endlichen Automaten handelt.

Eine reguläre Grammatik darf für ein Nichtterminalsymbol auf der linken Seite und ein Terminalsymbol auf der rechten Seite mehrere Produktionen enthalten. Im Beispiel finden z. B. sowohl die Regel  $F \rightarrow CG$  als auch  $F \rightarrow CJ$ . Dadurch können sich bei der Transformation in einen Automaten für einen Zustand und ein Terminalsymbol mehrere Übergänge ergeben. Der Automat ist dann nicht mehr deterministisch. Man spricht stattdessen von einem *nichtdeterministischen endlichen Automaten* (NEA).

Heißt das, dass es formale Sprachen gibt, die sich zwar mithilfe einer regulären Grammatik erzeugen lassen, zu denen es jedoch keinen deterministischen endlichen Automaten gibt? Nein! Die Transformation der regulären Grammatik in einen DEA verläuft in diesem Fall lediglich über einen Umweg. Wenn wir unsere Regeln zur Transformation einer Grammatik in einen Automaten anwenden, erhalten wir zwar zunächst einen NEA. Man kann jedoch zeigen, dass sich jeder NEA in einen DEA transformieren lässt und die Modelle gleich mächtig sind, das heißt die gleichen Grenzen aufweisen. Abbildung 5 zeigt den DEA, der sich aus dem NEA in Abbildung 4 für die Grammatik  $G_{\text{AlkeneReg}}$  ergibt.

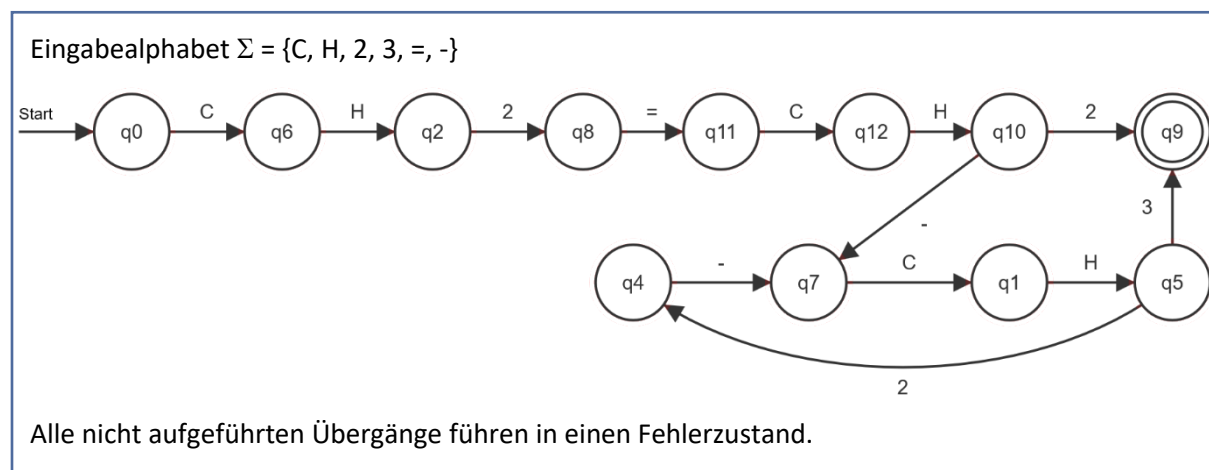


Abbildung 5: DEA zum NEA aus Abbildung 4

Das Modell des NEA wollen wir hier nicht vertiefen. Werkzeuge wie *AtoCC*<sup>1</sup> oder der Nachfolger *flaci.com*<sup>2</sup> bieten jedoch die Möglichkeit eine reguläre Grammatik in einen NEA und einen NEA in einen DEA zu transformieren. Mit diesen Werkzeugen können Sie sich anhand einiger Beispiele davon überzeugen, dass auch eine Grammatik, deren Produktionen zunächst zu einem NEA führen, in einen DEA transformiert werden kann.

<sup>1</sup> AtoCC wird von M. Hielscher zur Verfügung gestellt: [www.atocc.de](http://www.atocc.de)

<sup>2</sup> FLACI.com wird von der Pädagogischen Hochschule Schwyz kostenlos für Bildungszwecke zur Verfügung gestellt: <https://flaci.com>

## Hinweis

Die Materialien erheben keinen Anspruch auf Vollständigkeit hinsichtlich der für die Abiturprüfung erwarteten Kompetenzen. Verbindlich für das Abitur in Niedersachsen sind allein das niedersächsische Kerncurriculum für die gymnasiale Oberstufe sowie die ergänzenden Hinweise in der jeweils aktuellen Fassung.

## Lizenz

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#). Sie erlaubt Bearbeitungen und Weiterverteilung des Werks unter Nennung meines Namens und unter gleichen Bedingungen, jedoch keinerlei kommerzielle Nutzung.

**Abbildungsnachweise:** Die Abbildungen wurden mithilfe der Software FLACI.com erzeugt. FLACI.com wird von der Pädagogischen Hochschule Schwyz kostenlos für Bildungszwecke zur Verfügung gestellt: <https://flaci.com>

## Quellen

D. W. Hoffmann (2011). *Theoretische Informatik*. (2. Aufl.) Carl Hanser Verlag München.