

Deterministische endliche Automaten

Mit den **deterministischen endlichen Automaten (DEA)** lernen wir ein Maschinenmodell kennen, das systematisch prüft, ob ein Wort zu einer bestimmten formalen Sprache gehört.

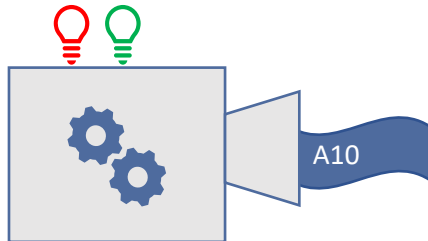


Abbildung 1: Schematische Darstellung eines DEA

Einen DEA können wir uns als eine Maschine wie in Abbildung 1 vorstellen, die eine Eingabe erhält, diese Zeichen für Zeichen verarbeitet und als Ausgabe immer ja (grüne Lampe leuchtet) oder nein (rote Lampe leuchtet) zurückgibt, je nachdem, ob das Wort zur Sprache gehört oder nicht. Man nennt sie deshalb auch *Akzeptor* oder *Erkennendensystem*.

Indem wir die Regeln, nach denen die Wörter einer Sprache aufgebaut sind, in einem DEA abbilden, definieren wir gleichzeitig, welche Wörter zu der formalen Sprache gehören. Die Frage, ob ein Wort zu einer Sprache gehört, ist zudem in der Praxis in unterschiedlichen Zusammenhängen von Bedeutung, z. B. wenn wir bei Eingabefeldern eines Formulars überprüfen, ob die Eingabe syntaktisch korrekt ist, wenn ein neues Passwort nur akzeptiert wird, wenn es bestimmten Sicherheitsstandards genügt, oder wenn der Compiler die Syntax eines Programms überprüft. Hier muss letztlich immer das gleiche Problem gelöst werden: Gehört das Eingabewort zur formalen Sprache der gültigen Eingaben?

Einen entsprechenden Algorithmus zu entwickeln ist je nachdem, wie die formale Sprache aufgebaut ist, mehr oder weniger komplex. Später schauen wir uns daher an, wie wir aus den Automaten systematisch einen geeigneten Algorithmus ableiten können.

Definition deterministischer endlicher Automaten

Schauen wir uns die Definition eines DEA am Beispiel der formalen Sprache der DIN-Formate an. DIN-Formate legen die Größe von Papier, Heften, Briefumschlägen usw. fest. Die formale Sprache der DIN-Formate beinhaltet folgende Wörter:

$L_{\text{DIN}} = \{A0, A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, B0, B1, B2, B3, B4, B5, B6, B7, B8, B9, B10, C0, C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D10\}$

Abbildung 2 zeigt den DEA_{DIN} , der genau die Wörter akzeptiert, die zur formalen Sprache der DIN-Formate gehören.

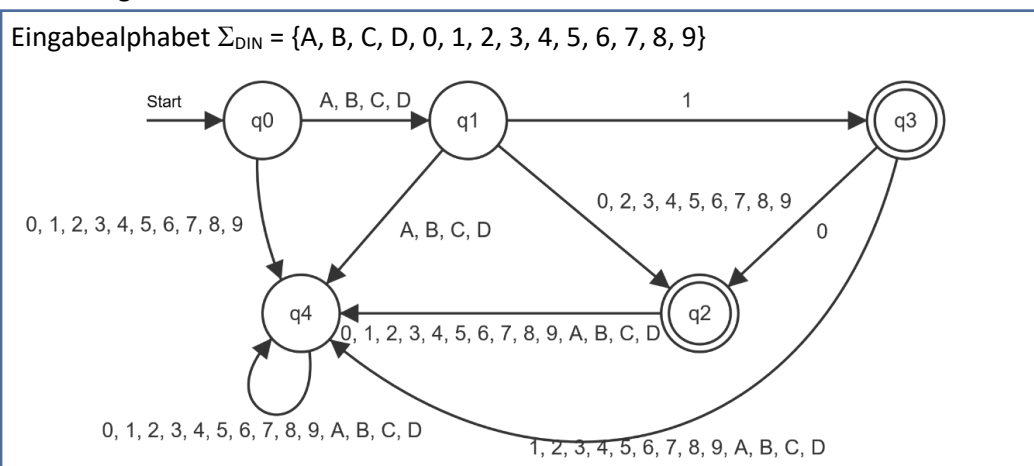


Abbildung 2: DEA_{DIN} für die Sprache der DIN-Formate

Die Definition eines DEA geben wir mithilfe eines Zustandsübergangsgraphen und eines Eingabealphabets an. Das **Eingabealphabet** entspricht dem Alphabet Σ der formalen Sprache, die der Automat akzeptiert. Der **Zustandsübergangsgraph** eines DEA besteht aus einer endlichen Menge von **Zuständen**, die als Kreise dargestellt werden. Der Automat arbeitet ein Eingabewort von links nach rechts Zeichen für Zeichen ab. Zwischen den Zuständen sind **Übergänge** definiert, die angeben, für welches Eingabezeichen man in welchen Folgezustand gelangt. Einer der Zustände ist durch einen kleinen Pfeil als **Startzustand** gekennzeichnet. Im Beispiel ist das der Zustand q_0 . Außerdem sind ein oder mehrere Zustände durch eine doppelte Umrandung als **Endzustände** definiert. Im Beispiel sind das die Zustände q_2 und q_3 . Ein Wort gehört genau dann zu der formalen Sprache, die der DEA akzeptiert, wenn die Bearbeitung des Wortes in einem Endzustand endet.

Betrachten wir als Beispiel die Verarbeitung der Eingabe A4. Der Automat startet in seinem Startzustand q_0 . Für das Eingabezeichen A wechselt er in den Zustand q_1 . Für das Eingabezeichen 4 wechselt er anschließend in den Zustand q_2 . Da das Wort nun vollständig verarbeitet wurde und q_2 ein Endzustand ist, wissen wir, dass der Automat das Wort akzeptiert und es somit zur formalen Sprache der DIN-Formate gehört.

Die Bearbeitung eines Eingabewortes durch einen DEA lässt sich kürzer als Folge von Konfigurationen darstellen, die der Automat nacheinander annimmt. Eine **Konfiguration** besteht dabei immer aus dem aktuellen Zustand und der noch zu verarbeitenden Eingabe. Für das Beispiel A4 ergibt sich also die Konfigurationsfolge: $(q_0, A4) \rightarrow (q_1, 4) \rightarrow (q_2, \epsilon)$

ϵ steht dabei für das leere Wort, das heißt, es sind keine Eingabezeichen mehr vorhanden.

Beispiele für gültige Eingaben, deren Bearbeitung in den Endzuständen q_2 oder q_3 endet:

1. $(q_0, B10) \rightarrow (q_1, 10) \rightarrow (q_3, 0) \rightarrow (q_2, \epsilon)$
2. $(q_0, D9) \rightarrow (q_1, 9) \rightarrow (q_2, \epsilon)$
3. $(q_0, D1) \rightarrow (q_1, 1) \rightarrow (q_3, \epsilon)$

Beispiele für ungültige Eingaben:

1. $(q_0, A20) \rightarrow (q_1, 20) \rightarrow (q_2, 0) \rightarrow (q_4, \epsilon)$
2. $(q_0, B100) \rightarrow (q_1, 100) \rightarrow (q_3, 00) \rightarrow (q_2, 0) \rightarrow (q_4, \epsilon)$
3. $(q_0, C) \rightarrow (q_1, \epsilon)$

Aufgabe 1: Die Eigenschaft *deterministisch* bedeutet in Bezug auf endliche Automaten, dass die Verarbeitung eines Wortes eindeutig ist. Anschaulich heißt das, es gibt für jedes Eingabewort genau einen Weg, der durch die Zustände des Automaten führt.

- a) Zeigen Sie für die Eingaben A13, B5 und D1234 und den DEA_{DIN} , dass die Verarbeitung eindeutig ist, und geben Sie an, ob die Wörter akzeptiert werden.
- b) Begründen Sie allgemein, dass der DEA_{DIN} deterministisch ist.
- c) Vergleichen Sie den Zustand q_4 mit den übrigen Zuständen. Welche Besonderheiten fallen hier auf?

Der Fehlerzustand

Zustand q_4 ist eine Art Fehlerzustand. Aus diesem Zustand führt kein Übergang heraus. Da es sich nicht um einen Endzustand handelt, steht fest, dass der Automat das Eingabewort nicht akzeptiert, sobald dieser Zustand erreicht wird. Der Automat wechselt in diesen Fehlerzustand, wenn bei der Verarbeitung des Eingabewortes ein Zeichen auftritt, dass nach den Regeln der formalen Sprache an dieser Stelle nicht vorkommen darf. Um den Zustandsübergangsgraphen des Automaten übersichtlicher zu gestalten, wird der Fehlerzustand in der Zeichnung häufig weggelassen und durch den Zusatz „Alle nicht aufgeführten Übergänge führen in einen Fehlerzustand“ ersetzt. Da ein endlicher Automat nur deterministisch ist, wenn für jedes Eingabezeichen und jeden Zustand genau ein Übergang definiert ist, darf dieser Satz bei der reduzierten Zeichnung des Zustandsübergangsgraphen auf keinen Fall fehlen, damit die Definition des DEA vollständig ist.

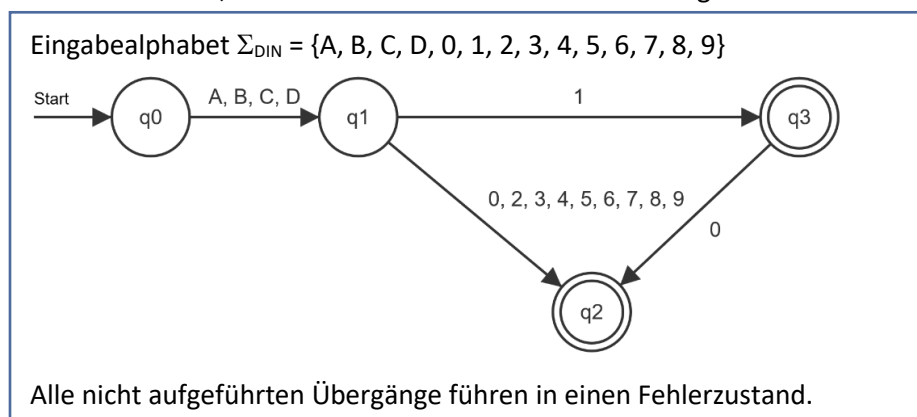


Abbildung 3: DEA_{DIN} mit Zustandsübergangsgraph ohne Fehlerzustand

Analyse und Konstruktion von DEAs

Aufgabe 2: Abbildung 4 zeigt den DEA_{Dezi} .

- a) Überprüfen Sie, ob der DEA die folgenden Eingabewörter akzeptiert.

- 1) 12,423
- 2) 0,3
- 3) 10
- 4) ,45
- 5) 0006,3

- b) Entscheiden Sie, ob es sich bei einem der Zustände um einen Fehlerzustand handelt.

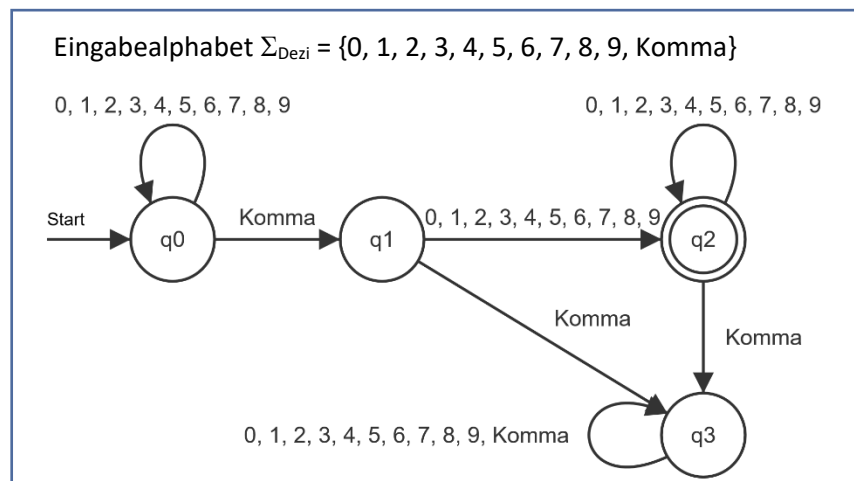


Abbildung 4: Der DEA_{Dezi} (die Eingabe „Komma“ steht für das Kommazeichen)

- c) Verändern Sie den DEA_{Dezi} so, dass er die formale Sprache der positiven Dezimalzahlen akzeptiert. Dazu gehören auch Zahlen, die nur aus einem ganzzahligen Teil bestehen. Führende Nullen sind hingegen nicht erlaubt, es sei denn die Null ist die einzige Ziffer vor dem Komma.
- d) Erläutern Sie am Beispiel des DEA zum Erkennen von Dezimalzahlen, ...

... wann bei der Konstruktion eines DEA ein neuer Zustand eingerichtet werden muss.

... wie die beliebig häufige Wiederholung eines Zeichens modelliert wird. Unterscheiden Sie dabei die Fälle, dass das Zeichen „kein Mal“ oder „mindestens einmal“ vorkommen muss.

Aufgabe 3:

- Jahreszahlen bestehen aus vier Ziffern. Konstruieren Sie einen DEA, der alle nach dieser Vorgabe syntaktisch korrekten Jahreszahlen akzeptiert.
- Erläutern Sie allgemein, wie ein DEA zählen kann, also wie überprüft werden kann, ob ein Wort eine bestimmte Anzahl an Zeichen enthält.
- Verändern Sie Ihren Automaten so, dass nur Jahreszahlen, die mit 19 oder 20 beginnen, akzeptiert werden.
- Erweitern Sie Ihren Automaten aus Aufgabenteil a) so, dass er Datumsangaben der Form TT.MM.JJJJ akzeptiert.

Hinweis: Sie können dabei das Zeichen z für eine beliebige Ziffer von 0 bis 9 verwenden.

Aufgabe 4: Betriebssysteme bieten in der Regel die Möglichkeit, nach Dateien, Ordnern und Programmen anhand des Namens oder der Dateierdung zu suchen. Auch Textdokumente können nach bestimmten Wörtern durchsucht werden. Diese Suche kann mithilfe endlicher Automaten modelliert werden, indem alle Dateien bzw. Textdokumente, welche die Suchkriterien erfüllen, als formale Sprache aufgefasst werden.

- Abbildung 5 zeigt einen DEA mit dem Eingabealphabet $\Sigma = \{c, d, o, z, .\}$, wobei z für alle Buchstaben des Alphabets außer c, d und o steht. Analysieren Sie, welche Wörter der Automat akzeptiert.
- Konstruieren Sie für die folgenden Beispiele jeweils einen DEA. Gehen Sie vereinfachend davon aus, dass Dateinamen bzw. Texte nur aus Klein- und Großbuchstaben sowie Punkten bestehen. Alle Zeichen, die nicht unterschieden werden müssen, können Sie im Eingabealphabet mit dem Zeichen z zusammenfassen.

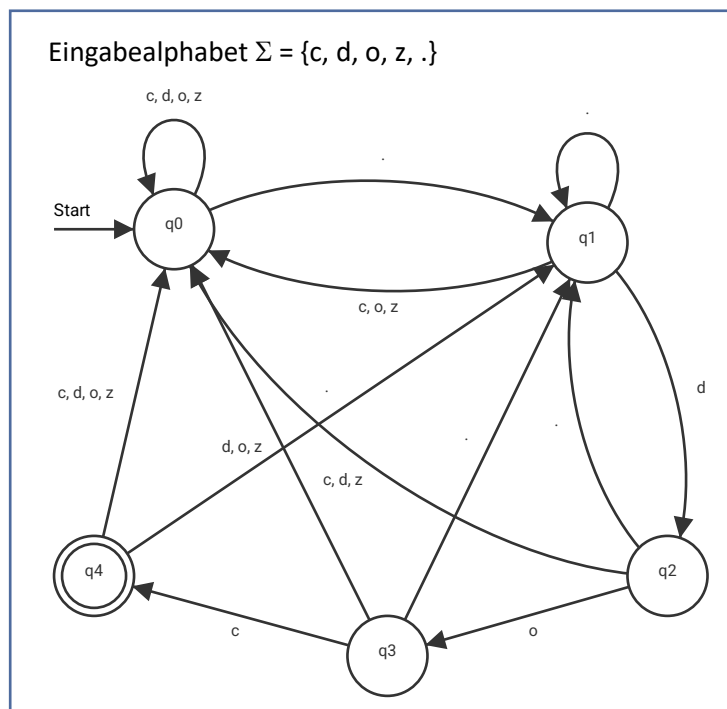


Abbildung 5: DEA zu Aufgabe 4a

- Gesucht werden alle Dateien, deren Name mit *AB* beginnt.
- Gesucht werden alle Dateinamen mit der Endung *ppt* oder *pptx*.
- Gesucht werden alle Texte, die mit der Zeichenkombination *LOL* enden.
- Gesucht werden alle Texte, die das Kürzel *JIT* enthalten.

Aufgabe 5: Ziffernkolonnen werden zur besseren Lesbarkeit häufig in Viererblöcken, die durch Leerzeichen getrennt sind, gedruckt. Bei der Eingabe zur digitalen Verarbeitung werden die Leerzeichen hingegen weggelassen.

- a) Konstruieren Sie einen DEA, der überprüft, ob die Länge des Eingabewortes ein Vielfaches von vier ist.
- b) Begründen Sie, dass ein DEA, der die hier betrachtete Sprache akzeptiert, keinen Fehlerzustand benötigt.

Aufgabe 6: Um ungültige Eingaben zu minimieren, soll bei einem Eingabefeld für eine E-Mail-Adresse überprüft werden, ob die Adresse syntaktisch korrekt ist.

- a) Sammeln Sie Regeln, nach denen eine syntaktisch korrekte E-Mail-Adresse aufgebaut sein muss.
- b) Konstruieren Sie einen zu Ihren Regeln passenden DEA, der die Sprache der syntaktisch korrekten E-Mail-Adressen akzeptiert.
- c) Informieren Sie sich im Internet, welche Verfahren noch eingesetzt werden, um die Gültigkeit von E-Mail-Adressen zu überprüfen.

Aufgabe 7:

- a) Durch fünf teilbare Zahlen enden mit den Ziffern 0 oder 5. Konstruieren Sie einen DEA, der genau die natürlichen Zahlen akzeptiert, die durch 5 teilbar sind.
- b) Sammeln Sie in Kleingruppen weitere Teilbarkeitsregeln und diskutieren Sie, ob sich dazu ein DEA konstruieren lässt. Wählen Sie geeignete Teilbarkeitsregeln aus und konstruieren Sie arbeitsteilig jeweils einen DEA der überprüft, ob eine natürliche Zahl durch die ausgewählte Zahl teilbar ist.

Aufgabe 8:

- a) Sammeln Sie Regeln für sichere Passwörter. Diskutieren Sie, welche der Regeln sich mithilfe eines DEA überprüfen lassen.
- b) Wählen Sie geeignete Regeln zur Definition einer formalen Sprache sicherer Passwörter aus und konstruieren Sie einen entsprechenden DEA.

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: Abbildung 1 wurde mithilfe von Formen und Piktogrammen in Microsoft Word 2016 erstellt.

Die Abbildungen 2 bis 5 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.

Die Liste der DIN-Formate basiert auf den Angaben auf: R. Sachs. *Übersicht über die Maße der Papierformate DIN A, DIN B, DIN C und DIN D, Größen in Millimeter mm und Zentimeter cm*

<https://www.din-formate.de/uebersicht-tabelle-masse-din-a-b-c-d-din-lang-groessen-zb-a4-b4-c4-d4-zusammenfassung-papierformate-in-mm-cm.html> (Abgerufen: 3. Juli 2020)