

Grenzen regulärer Grammatiken und endlicher Automaten

Formale Sprachen begegnen uns ständig im Alltag, ohne dass wir sie bewusst wahrnehmen. So können häufig insbesondere bei der Kommunikation „Mensch-Maschine“ nur zu einer bestimmten Sprache gehörende „erlaubte“ Ausdrücke in ein Formular eingegeben oder verwendet werden: Beispiele sind syntaktisch korrekte Geburtsdaten, syntaktisch plausible Email-Adressen oder aber auch Taschenrechnereingaben, die auf eine fest vorgegebene Art zu erfolgen haben.

Wir haben bisher die Theorie der regulären Grammatiken zur Erzeugung regulärer Sprachen kennengelernt. Außerdem haben wir deterministische endlichen Automaten (DEA) zum Erkennen *bestimmter* formaler Sprachen genutzt. Vielleicht haben Sie bereits erkannt, dass diese *bestimmten* formalen Sprachen genau die regulären Sprachen sind, dass also DEAs und reguläre Grammatiken die gleiche Sprachklasse definieren. Jedoch ist nicht jede formale Sprache eine reguläre Sprache – bereits für eines der drei anfangs erwähnten Beispiele reicht diese Sprachdefinition nicht mehr aus.

Im Folgenden wollen wir die Grenzen endlicher Automaten zur Beschreibung formaler Sprachen genauer erkunden.

Beispiel: Zitate

In der deutschen Sprache wird eine direkte Rede oder ein Zitat jeweils von öffnenden und schließenden Anführungszeichen umschlossen: „Dies ist ein Beispiel“. Fehlt das öffnende oder schließende Anführungszeichen, so ist der Anfang oder das Ende einer direkten Rede oder eines Zitats schwer erfassbar und ein Satz syntaktisch falsch: „Beispiel

Die Sprache der „einfachen“ Zitate ist mithilfe einer regulären Grammatik noch leicht zu beschreiben:

Nichtterminalsymbole: {S, A, B}

Terminalsymbole: {z, „, “} z steht für ein Zeichen, welches kein Anführungszeichen ist

Startsymbol: S

Produktionen:

$S \rightarrow z S \mid „ A \mid z$

$A \rightarrow z B$

$B \rightarrow z B \mid “ S \mid “$

Aufgaben:

- 1) Entwickeln Sie einen deterministischen endlichen Automaten (DEA), der „einfache“ Zitate akzeptiert. Als Eingabealphabet können Sie die Terminalsymbole {z, u, o} verwenden, wobei z für ein Zeichen steht, welches kein Anführungszeichen ist und für die bessere Lesbarkeit im Automaten jeweils u für ein Anführungszeichen unten und o für ein Anführungszeichen oben steht.
- 2) Überprüfen Sie jeweils mithilfe der Angabe einer Zustandsübergangsfolge, ob die Eingabefolgen „Beispiel und Dies ist ein „Beispiel“ von Ihrem DEA akzeptiert werden.

- 3) Untersuchen Sie, welche Änderungen an Ihrem DEA nötig sind, damit dieser auch Zitate in Zitaten akzeptiert. Ein Beispiel für ein solches Zitat in einem Zitat ist: „Dies ist kein „einfaches“ Zitat“.
- 4) Geben Sie eine reguläre Grammatik zur Beschreibung der Sprache der Zitate in Zitaten an.

Zitat in einem Zitat in einem Zitat

Eingabealphabet $\Sigma = \{z, u, o\}$

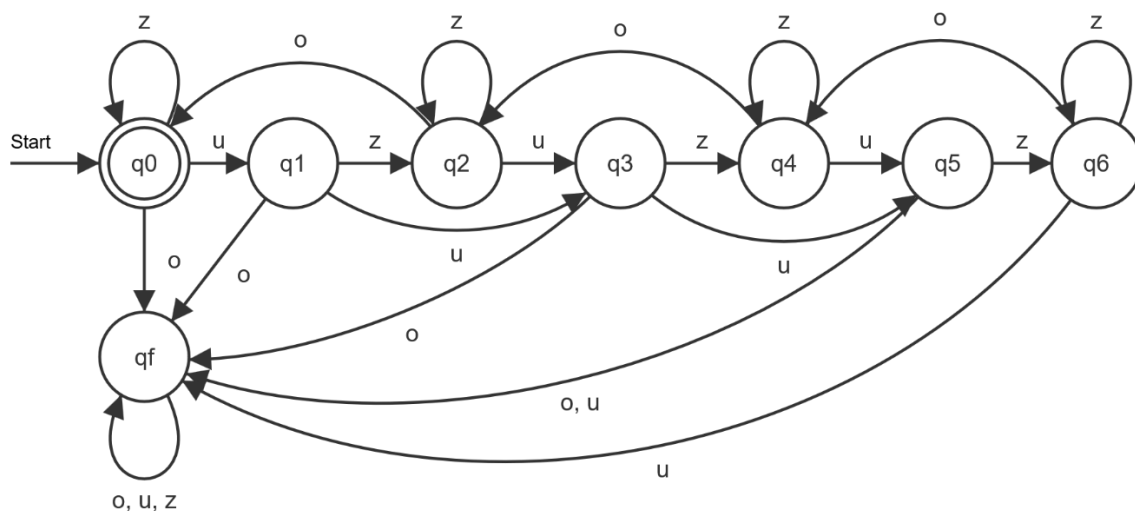


Abbildung 1

Der in Abbildung 1 dargestellte DEA akzeptiert auch ein Zitat in einem Zitat in einem Zitat.

Aufgaben:

- 5) Zeigen Sie, dass der Automat die Eingabefolge „„„Zitat“ in einem Zitat“ in einem Zitat“ akzeptiert.
- 6) Bei einem Zitat in einem Zitat spricht man von der Schachtelungstiefe 1, bei einem Zitat in einem Zitat in einem Zitat von der Schachtelungstiefe 2. Ein einfaches Zitat ist nicht geschachtelt, hat damit also die Schachtelungstiefe 0. Vergleichen Sie den endlichen Automaten aus Abbildung 1 mit Ihren endlichen Automaten aus den Aufgaben 1) und 3) und erklären Sie, wie sich eine Erhöhung der Schachtelungstiefe auf den Übergangsgraphen des DEA auswirkt.

Ein Zitat in einem Zitat in einem Zitat kommt in der deutschen Sprache sehr selten vor. Insofern machen weitere Schachtelungstiefen (Zitat in einem Zitat in einem Zitat in ...) kaum Sinn.

- 7) Überlegen Sie, ob Sie andere Beispiele für formale Sprachen finden, bei denen von „Schachtelungstiefen“ gesprochen werden kann.
- 8) Bei Zitaten kennen wir die maximale Schachtelungstiefe: wir gehen hier von der Zahl 2 aus. Deshalb kann die zugehörige Sprache mithilfe eines DEA modelliert werden. Diskutieren Sie, welche Auswirkungen es haben könnte, wenn man die maximale Schachtelungstiefe nicht kennt bzw. wenn es keine maximale Schachtelungstiefe gibt.

Klammersprachen

Die „Sprache“ der Zitate ist eine spezielle *Klammersprache*. Die öffnenden und schließenden Anführungszeichen *umklammern* einen Text – fehlt ein Anführungszeichen oder wird ein falsches Anführungszeichen gesetzt, so ist eine Zeichenfolge syntaktisch falsch.

Es gibt viele weitere Beispiele für Klammersprachen:

- Taschenrechnereingaben enthalten oft Klammern – dabei gehört jeweils zu jeder öffnenden eine schließende Klammer.
- Programmiersprachen enthalten Klammersausdrücke:
 - in Java werden Befehlsblöcke beispielsweise mit geschweiften Klammern { und } umschlossen
 - in Delphi werden Befehlsblöcke durch `begin` und `end` umschlossen, das Schlüsselwort `begin` entspricht also der öffnenden und `end` der schließenden Klammer.

Die nicht reguläre Sprache $\{a^n b^n \mid n=1,2,3,\dots\}$

In der Literatur findet man häufig als Beispiel für eine nicht reguläre Sprache die Sprache $\{a^n b^n \mid n=1, 2, 3, \dots\}$. Dabei steht die Bezeichnung $\{a^n b^n \mid n=1, 2, 3, \dots\}$ stellvertretend für eine bestimmte Art von Klammersausdrücken: eine gewisse Anzahl des Zeichens *a* gefolgt von der gleichen Anzahl des Zeichens *b*. So gehören beispielsweise die Zeichenfolgen *aabb* oder *aaabbb* zur Sprache $\{a^n b^n\}$, die Zeichenfolgen *aaab* und *bbaa* dagegen nicht.

Das *a* ist Stellvertreter für die öffnende Klammer (oder das öffnende Anführungszeichen, das `begin` eines Delphi-Befehlsblocks, die geschweifte Klammer eines Java-Befehlsblocks, die runde öffnende Klammer einer Taschenrechnereingabe...), das *b* für die schließende Klammer (oder das schließende Anführungszeichen, das `end` eines Delphi-Befehlsblocks, die schließende geschweifte Klammer eines Java-Befehlsblocks, die runde schließende Klammer einer Taschenrechnereingabe...).

Aufgaben:

- 9) Nennen Sie Gemeinsamkeiten und Unterschiede zwischen der Sprache $\{a^n b^n \mid n=1, 2, 3, \dots\}$ und der „Sprache“ der Zitate.
- 10) Können Sie für die formale Sprache $\{a^n b^n \mid n=1, 2, 3, \dots\}$ eine reguläre Grammatik entwerfen? Erläutern Sie, welche Probleme hier auftreten.

Wir werden im Folgenden zeigen, dass die Sprache $\{a^n b^n \mid n=1, 2, 3, \dots\}$ nicht durch einen DEA modelliert werden kann.

Begründung der Nichtexistenz eines DEA für die Sprache $\{a^n b^n \mid n=1,2,3,\dots\}$

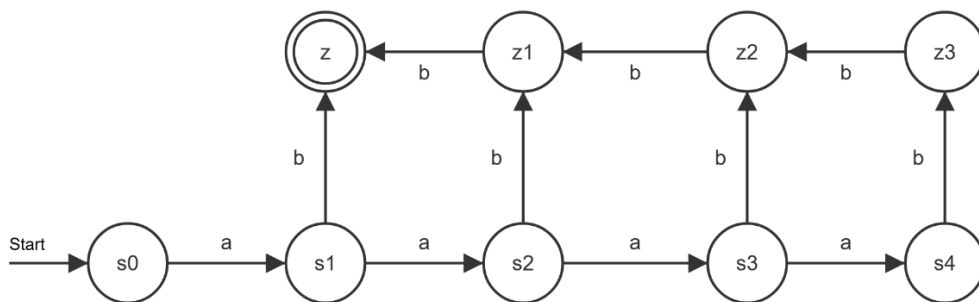


Abbildung 2: Zustandsgraph für Schachtelungstiefe 3, Eingabealphabet $\Sigma=\{a,b\}$. Nicht aufgeführte Übergänge führen in einen Fehlerzustand.

In Abbildung 2 ist der Übergangsgraph eines DEA der Sprache $\{a^n b^n \mid n=1, 2, 3, 4\}$, d.h. für eine vorgegebene maximale Schachtelungstiefe 3, angegeben. Man erkennt, dass man für jede Klammerebene zwingend ein zugehöriges Zustandspaar benötigt. Jeder Zustand „merkt“ sich die bereits erzielten Anzahlen der Zeichen a und b in einer Eingabe. Zum Beispiel steht der Zustand s2 für genau zwei eingelesene a, der Zustand z3 für noch genau drei einzulesende b. Möchte man jetzt eine weitere Schachtelungstiefe zulassen, also beispielsweise die Sprache $\{a^n b^n \mid n=1, 2, 3, 4, 5\}$ modellieren, so benötigt man zwei weitere Zustände s5 und z4. Dies lässt sich für weitere Schachtelungstiefen analog fortführen: für jede nächsthöhere maximale Schachtelungstiefe benötigt man zwei weitere Zustände.

Ein deterministischer endlicher Automat verfügt, wie der Name schon sagt, immer nur über endlich viele Zustände. Ein konkreter Übergangsgraph hat also immer nur eine feste, endliche Anzahl an Zuständen. Charakteristisch für die Sprache $\{a^n b^n \mid n=1, 2, 3, \dots\}$ ist jedoch, dass keine maximale Schachtelungstiefe existiert, dass sie also beliebig lange Ausdrücke $a \dots b \dots$ enthält. Hat man nun einen konkreten DEA mit einer gewissen Anzahl an Zuständen, so findet man immer eine längere Zeichenfolgen der Sprache $\{a^n b^n \mid n=1, 2, 3, \dots\}$, für die der vorhandene DEA nicht ausreicht und eine größere Anzahl an Zuständen zum Merken der bereits erzielten Anzahlen der Zeichen a und b nötig wäre. Es gibt somit keinen DEA, der die Sprache $\{a^n b^n \mid n=1, 2, 3, \dots\}$ akzeptiert.

Reguläre Sprache oder nicht?

Reguläre Grammatiken und deterministische endliche Automaten modellieren reguläre Sprachen und sind häufig Grundlage für die Entwicklung von Programmen, bei denen eine bestimmte Art von Eingaben erforderlich ist (vgl. Endliche Automaten als Modellierungswerkzeug – Implementierung endlicher Automaten). Aber nicht jede formale Sprache ist eine reguläre Sprache. Zum Nachweis, dass eine formale Sprache regulär ist, genügt es, eine reguläre Grammatik zur Erzeugung anzugeben. Alternativ kann stattdessen auch ein DEA zum Erkennen der Sprache angegeben werden.

Will man dagegen begründen, dass eine gegebene Sprache nicht regulär ist, reicht die Angabe einer nicht-regulären Grammatik nicht aus – es könnte ja eine andere, möglicherweise reguläre Grammatik geben, die die Sprache ebenfalls erzeugt. Zum Nachweis muss daher allgemeiner begründet werden, beispielsweise, dass es keine zugehörige reguläre Grammatik geben kann. Analog lässt sich über die Begründung der Nicht-Existenz eines DEAs nachweisen, dass eine gegebene Sprache nicht regulär ist. Die Argumentation erfolgt häufig analog zu obiger Begründung der Nichtexistenz eines DEA für die Sprache $\{a^n b^n \mid n=1, 2, 3, \dots\}$.

Aufgaben:

Entscheiden Sie für die folgenden Sprachen jeweils begründet, ob sie regulär sind oder nicht. Beschreiben Sie die jeweilige Sprache durch eine zugehörige Grammatik. Entwickeln Sie, falls möglich, den Übergangsgraphen eines DEA zur Modellierung der Sprache.

- 11) Die Sprache gültiger Klammerausdrücke für Taschenrechnereingaben. Dabei soll nur auf korrekte Klammerung mit runden Klammern geachtet werden, andere Eingabezeichen können ignoriert werden. Gültige Klammerausdrücke wären beispielsweise $(())()$ oder $()()()()$.
- 12) In Kaufhäusern findet man häufig für Kundenkarteninhaber günstigere Preise als für andere Kunden. Die günstigeren Preise werden dann in Klammern hinter dem Normalpreis angegeben.

Ausblick

Wir haben in unseren Überlegungen bisher nur angedeutet, dass DEAs und reguläre Grammatiken die gleiche Sprachklasse, nämlich die Menge der regulären Sprachen, definieren. Dieser Zusammenhang zwischen deterministischen endlichen Automaten und regulären Grammatiken wird in anderen Materialien vertieft.

Neben regulären Sprachen gibt es weitere Sprachklassen. So gehören Klammersprachen zu den sogenannten kontextfreien Sprachen. Das Automatenmodell des DEA kann dafür auf das Modell von Kellerautomaten erweitert werden, welche dann beispielsweise Klammersprachen akzeptieren.

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.

