Karlsruher Institut für Technologie

Theoretische Grundlagen der Informatik
Vorlesung am 15.10.2019

INSTITUT FÜR THEORETISCHE INFORMATIK

Heute

Organisatorisches

- Team, Termine
- Vorlesungsaufzeichnungen
- Homepage, Literaturempfehlungen
- Übungsblätter, Klausur
- Tutorien, Einteilung

Wiederholung aus GBI

- Wörter, formale Sprachen
- Reguläre Sprachen, reguläre Ausdrücke
- Endliche Automaten
- Kontextfreie Grammatiken

Team

Vorlesung:	**Übung:**	ab Januar:
Prof. Dorothea Wagner	Jonas Sauer	Guido Brückner
	jonas.sauer2@kit.edu	brueckner@kit.edu

Tutorium:

Mo	8:00	Julius Häcker	Mi	11:30	Anne Bernhart
Mo	15:45	Liran Dattner	Mi	15:45	Wendy Yi
Mo	15:45	Vera Chekan	Mi	15:45	Vera Chekan
Di	8:00	Noah Wahl	Do	8:00	Nicholas Bieker
Di	8:00	Christoph Kleiser	Do	9:45	Maria Nübling
Di	8:00	Rafael Baur	Do	9:45	Mingyi Li
Di	14:00	Tobias Erthal	Do	14:00	Eric Hamann
Di	14:00	Valentin Quapil	Do	14:00	Michael Schrempp
Di	17:30	Sebastian Faller	Do	15:45	Liran Dattner
Mi	8:00	Tina Strößner	Do	15:45	Michael Schrempp
Mi	8:00	Bendix Sonnenberg	Do	17:30	Omar Elhusseini
Mi	8:00	Kolja Bauer	Fr	9:45	Vasil Papanchev
Mi	9:45	Moritz Potthoff	Fr	9:45	Alexander Linder
Mi	11:30	Max Willich	Fr	11:30	Vasil Papanchev

Termine (Änderungen vorbehalten)

Dienstags		Donnerstags	
15.10.	Vorlesung	17.10.	Vorlesung
22.10.	Vorlesung	24.10.	Übung
29.10.	Vorlesung	31.10.	Vorlesung
05.11.	Übung	07.11.	Vorlesung
12.11.	Vorlesung	14.11.	Übung
19.11.	Vorlesung	21.11.	Übung
26.11.	Vorlesung	28.11.	Vorlesung
03.12.	Übung	05.12.	Vorlesung
10.12.	Vorlesung	12.12.	Übung
17.12.	Vorlesung	19.12.	Vorlesung
07.01.	Übung	09.01.	Vorlesung
14.01.	Vorlesung	16.01.	Übung
21.01.		23.01.	Vorlesung
28.01.	Vorlesung	30.01.	Übung
04.02.	Vorlesung		

Aufzeichnungen der Vorlesung

- Ton und Folien der Vorlesungen werden aufgezeichnet.
- Aufnahmen sind abrufbar über YouTube-Kanal KIT | WEBCAST:
  - `https://www.youtube.com/channel/UC6AqaL6fH91U5YhyUCI0ZOQ`
  - Aufnahmen werden nach den Vorlesungen online gestellt (mit gewisser Verzögerung).
  - Direkte Links auf der Vorlesungshomepage

Homepage

- https://i11www.iti.kit.edu/teaching/winter2019/tgi/index
- Aktuelle Informationen / Termine
- Alte Klausuren
- Folien
- Übungsblätter
- ILIAS-Forum
  - Für Fragen an die Übungsleiter
  - Für Austausch untereinander
  - https://ilias.studium.kit.edu/goto_produktiv_crs_1017400.html
- Links zu Vorlesungsaufzeichnungen
- Literaturempfehlungen

Literaturempfehlungen

- Ingo Wegener: **Theoretische Informatik**
  B.G. Teubner Verlag Stuttgart, 1993
- Uwe Schöning: **Theoretische Informatik - kurzgefasst**
  Hochschultaschenbuch, Spektrum Akademischer Verlag, 1997
- R. Garey und D. S. Johnson:
  Computers and Intractability: A Guide to the Theory of NP-Completeness
  W. H. Freeman, New York, 1979
- T. Cormen, C. Leiserson, R. Rivest: **Introduction to Algorithms**
  The MIT press, 1997, 2001.
- A. Asteroth, C. Baier:
  Theoretische Informatik: eine Einführung in Berechenbarkeit, Komplexität und formale Sprachen mit 101 Beispielen
  Pearson Studium, 2002

Übungsblätter

In der Regel:

- Ausgabe etwa jeden zweiten Dienstag
- In derselben Übung: Besprechung von ähnlichen Aufgaben
- Bearbeitungszeit: 1-2 Wochen (siehe Abgabedatum auf Übungsblatt!)
- Abgabe: Kasten im Untergeschoss des Informatik-Hauptgebäudes (50.34)
- Abgabe bis Dienstag, 11.00 Uhr im Kasten
- Rückgabe und Besprechung der korrigierten Blätter in den Tutorien
- Ausgabe des ersten Übungsblatts: Donnerstag 17.10.2019

Übungsblätter

- Doppelabgabe erlaubt, aber nur, wenn beide für das gleiche Tutorium eingeteilt sind
- Keine losen Blätter, bitte zusammenheften
- Abgaben müssen handschriftlich sein, also nicht gedruckt oder kopiert
- 25% der möglichen Gesamtpunktzahl $\rightsquigarrow$ 1 Bonuspunkt
  50% der möglichen Gesamtpunktzahl $\rightsquigarrow$ 2 Bonuspunkte
  75% der möglichen Gesamtpunktzahl $\rightsquigarrow$ 3 Bonuspunkte
- Der Klausurbonus wird nur auf bestandene Klausuren angerechnet
- Bei einmaligem Abschreiben: Keine Punkte auf das Blatt
- Bei wiederholtem Abschreiben: Keine Bonuspunkte

Klausur

- Hauptklausur: 23. März 2020 um 11:30 Uhr
- Dauer: 2 Stunden
- Orientierung: Altklausuren auf der Homepage
- Termin für Nachklausur wird noch bekannt gegeben

Tutorien

- Rückgabe und Besprechung der Übungsblätter
- Zusätzliche Aufgaben und Beispiele zum Stoff der Vorlesung
- Start: Nächste Woche (ab 21.10.2019)
- Einteilung über WebInScribe
  - `https://www.informatik.kit.edu/webinscribe/`
- Dort verlinkt: Merkblatt und Termine der Tutorien
- Anmeldebeginn: Dienstag, 15.10., 18:00 Uhr $\rightsquigarrow$ heute!
- Anmeldeschluss: Donnerstag, 17.10., 18:00 Uhr

Ziele der Vorlesung TGI

Welche Fragestellungen werden in TGI behandelt?

- Theoretische Grundlagen zu Algorithmen- und Programmentwurf: Wie kann man allgemeingültige (rechner- und programmiersprachenunabhängige) Aussagen zu gegebenen Problemstellungen machen?
- Gibt es Probleme die nicht von Computern gelöst werden können?
- Gibt es Probleme, für die Ausprobieren die beste Lösungsstrategie ist?
- Wie kann man konkrete Computer abstrakt betrachten oder modellieren?
- Wie kann man konkrete Programmiersprachen abstrakt betrachten oder modellieren?

Ziele der Vorlesung TGI

- Vertiefter Einblick in die Grundlagen der Theoretischen Informatik
- Beherrschen der Berechnungsmodelle und Beweistechniken der TI
- Verständnis für Grenzen und Möglichkeiten der Informatik in Bezug auf die Lösung von definierbaren, aber nur bedingt berechenbaren Problemen
- Abstraktionsvermögen für grundlegende Aspekte der Informatik von konkreten Gegebenheiten wie konkreten Rechnern oder Programmiersprachen
- Anwendung der erlernten Beweistechniken bei der Spezifikation von Systemen der Informatik und für den systematischen Entwurf von Programmen und Algorithmen

Wiederholung aus Grundbegriffe der Informatik

- Wörter
- Formale Sprachen
- Reguläre Ausdrücke
- Endliche Automaten
- Kontextfreie Grammatiken

Wörter

- Ein *Alphabet* Σ ist eine endliche Menge von Zeichen.

  Beispiele: $\Sigma = \{0, 1\}$ oder $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \cup \{,\} \cup \{.\}$.

- Ein *Wort* w über einem Alphabet Σ ist eine (möglicherweise leere) Folge von Zeichen aus Σ.

  Beispiel: $\Sigma = \{0, 1\}$ und $w = 0010010$.

  $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \cup \{,\} \cup \{.\}$ und $w = 1.024,48$

- Das *leere Wort* wird mit ε symbolisiert.

- Die *Menge aller Wörter* über einem Alphabet Σ wird mit Σ^* abgekürzt.

  Beispiel: $\Sigma = \{0, 1\}$ und $\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\}$

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

  $\rightsquigarrow$ ε und $1..0.4$ und $,,.7$ etc. in Σ^*

Wörter

- Die *Menge aller Wörter* mit Länge n über einem Alphabet Σ wird mit Σ^n abgekürzt.

  Beispiel $\Sigma = \{0, 1\}$:

$$\begin{aligned} \Sigma^0 &= \{\varepsilon\} \\ \Sigma^1 &= \{0, 1\} \\ \Sigma^2 &= \{00, 01, 10, 11\} \end{aligned}$$

- Die *Konkatenation* (Aneinanderhängen) zweier Wörter w_1 und w_2 wird mit $w_1 \cdot w_2$ oder $w_1 w_2$ abgekürzt.

  Beispiel: $w_1 = 001$ und $w_2 = 110$ dann $w_1 \cdot w_2 = 001110$.

  $w_1 = 001$ und $w_2 = \varepsilon$ dann $w_1 \cdot w_2 = 001$.

Wörter

- Die *iterierte Konkatenation* eines Wortes w ist $w^k := \underbrace{w \cdot \ldots \cdot w}_{k\text{-mal}}$

  Beispiel: $w = 110$.

$$
\begin{aligned}
w^0 &= \varepsilon \\
w^1 &= 110 \\
w^2 &= 110 \cdot 110 = 110110 \\
w^3 &= 110 \cdot 110 \cdot 110 = 110110110
\end{aligned}
$$

Formale Sprachen

- Eine *formale Sprache* L über einem Alphabet Σ ist eine Teilmenge $L \subseteq \Sigma^*$.

  Beispiel: Sprache L' aller Wörter, deren vorletztes Zeichen 0 ist

$$L' = \{w0z \mid w \in \Sigma^*, z \in \Sigma\}$$

- Das Produkt $L_1 \cdot L_2$ der Sprachen L_1 und L_2 ist definiert als

$$L_1 \cdot L_2 := \{w_1 w_2 \mid w_1 \in L_1 \text{ und } w_2 \in L_2\}$$

  Beispiel:

$$L' = \Sigma^* \cdot \{0\} \cdot \Sigma$$

Formale Sprachen

- Produkte einer Sprache L mit sich selbst werden abgekürzt als

$$L^k := \underbrace{L \cdot \ldots \cdot L}_{k\text{-mal}}$$

Beispiel $L = \{00, 1\}$:

$$\begin{aligned}
L^0 &= \{\varepsilon\} \\
L^1 &= \{00, 1\} \\
L^2 &= \{00, 1\} \cdot \{00, 1\} = \{0000, 001, 100, 11\} \\
L^3 &= \{0000, 001, 100, 11\} \cdot \{00, 1\} \\
&= \{000000, 00100, 10000, 1100, 00001, 0011, 1001, 111\}
\end{aligned}$$

Formale Sprachen

Lässt sich ein Wort w schreiben als $w = u \cdot v \cdot x$, wobei u, v, x beliebige Wörter sind, so heißt:

$$\left.\begin{array}{ll} u & \text{Präfix} \\ v & \text{Teilwort} \\ x & \text{Suffix} \end{array}\right\} \text{von } w$$

Beispiel $w = \text{TAL}$

$$\begin{array}{ll} \text{Präfixe} & P = \{\varepsilon, \text{T}, \text{TA}, \text{TAL}\} \\ \text{Suffixe} & S = \{\varepsilon, \text{L}, \text{AL}, \text{TAL}\} \\ \text{Teilworte} & \{A\} \cup P \cup S \end{array}$$

Formale Sprachen

Seien $L, L_1, L_2 \subseteq \Sigma^*$ Sprachen.

Produktsprache	$L_1 \cdot L_2 := \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$
k-faches Produkt	$L^k := \{w_1 \cdot \ldots \cdot w_k \mid w_i \in L \text{ für } 1 \leq i \leq k\}$ $L^1 = L, \quad L^0 := \{\varepsilon\}$
Kleene'scher Abschluss	$L^* := \bigcup_{i \geq 0} L^i$
Positiver Abschluss	$L^+ := \bigcup_{i \geq 1} L^i$
Quotientensprache	$L_1 / L_2 := \{w \in \Sigma^* \mid \exists z \in L_2 \text{ mit } w \cdot z \in L_1\}$
Komplementsprache	$L^c := \Sigma^* \setminus L$

Reguläre Sprachen

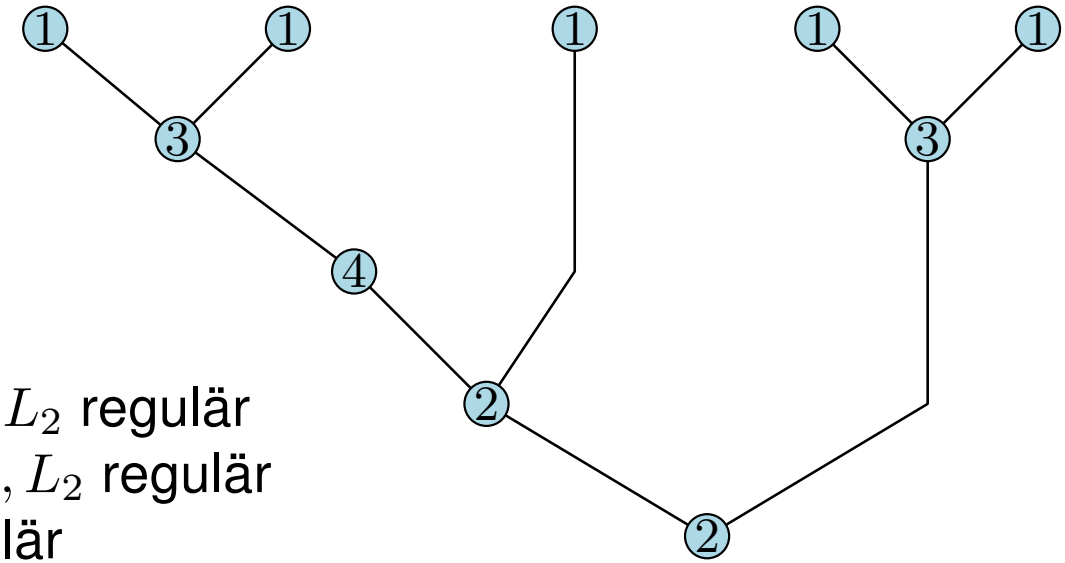
Eine Sprache $L \subseteq \Sigma^*$ heißt *regulär*, wenn für sie einer der folgenden Punkte gilt: (induktive Definition)

- Verankerung:
  - $L = \{a\}$ mit $a \in \Sigma$ oder
  - $L = \{\varepsilon\}$ oder
  - $L = \emptyset$
- Induktion: Es gibt reguläre Sprachen L_1, L_2, sodass
  - $L = L_1 \cdot L_2$ oder
  - $L = L_1 \cup L_2$ oder
  - $L = L_1^*$

Reguläre Sprachen

Beispiel: Die Sprache L' aller Wörter über $\{0, 1\}$, die als vorletztes Zeichen eine 0 haben

$$L' := (\{0\} \cup \{1\})^* \cdot \{0\} \cdot (\{0\} \cup \{1\})$$



Verankerung:

① $L = \{a\}$ mit $a \in \Sigma$

Induktion:

② $L = L_1 \cdot L_2$ mit L_1, L_2 regulär

③ $L = L_1 \cup L_2$ mit L_1, L_2 regulär

④ $L = L_1^*$ mit L_1 regulär

Reguläre Ausdrücke

- Wir benutzen eine leicht andere Schreibweise für reguläre Ausdrücke als in GBI.
- Sei Σ eine Alphabet. Eine reguläre Sprache über Σ kann durch einen *regulären Ausdruck* beschrieben werden.

Reguläre Ausdrücke

Dabei bezeichnet

- $\emptyset$ den regulären Ausdruck, der die leere Menge beschreibt.
  $\rightsquigarrow L(\emptyset) = \{\}$
- ε den regulären Ausdruck, der die Menge $\{\varepsilon\}$ beschreibt.
  $\rightsquigarrow L(\varepsilon) = \{\varepsilon\}$
- a den regulären Ausdruck, der die Menge $\{a\}$ beschreibt.
  $\rightsquigarrow L(a) = \{a\}$

Wenn α, β reguläre Ausdrücke sind, die die Sprachen $L(\alpha), L(\beta)$ beschreiben, so schreiben wir

- $(\alpha) \cup (\beta)$ für $L(\alpha) \cup L(\beta)$
- $(\alpha) \cdot (\beta)$ für $L(\alpha) \cdot L(\beta)$
- $(\alpha)^+$ für $L(\alpha)^+$
- $(\alpha)^*$ für $L(\alpha)^*$

Reguläre Ausdrücke

Notation

- Wir schreiben auch α statt $L(\alpha)$ und $w \in \alpha$ statt $w \in L(\alpha)$.
- * bindet stärker als $\cdot$ und $\cdot$ stärker als $\cup$
- Das heißt zum Beispiel:

$$a \cup b \cdot c = a \cup (b \cdot c) = a \cup bc$$

- Wir lassen Konkatenationspunkte und unnötige Klammern weg.

Reguläre Ausdrücke – Beispiele

- L' ist der reguläre Ausdruck für die Sprache aller Wörter über $\{0,1\}$, die als vorletztes Zeichen eine 0 haben

$$L' = (0 \cup 1)^* 0 (0 \cup 1)$$

- $L := \{w \in \{0,1\}^* \mid w \text{ enthält 10 als Teilwort}\}$

$$L = (0 \cup 1)^* 10 (0 \cup 1)^*$$

- $L := \{w \in \{0,1\}^* \mid w \text{ enthält 101 als Teilwort}\}$

$$L = (0 \cup 1)^* 101 (0 \cup 1)^*$$

Reguläre Ausdrücke – Beispiele

- $L := \{w \in \{0,1\}^* \mid w \text{ enthält 10 nicht als Teilwort}\} = 0^*1^*$, denn
  - Sei $w \in 0^*1^*$. Dann kommt in w nach keiner 1 eine 0.
    Also ist $w \in L$ und damit $0^*1^* \subseteq L$.
  - Sei $w \in L$. Dann kommen nach der ersten Eins keine Nullen mehr vor, d.h. $w = w'1 \ldots 1$, wobei w' keine 1 enthält.
    Also ist $w \in 0^*1^*$ und damit $L \subseteq 0^*1^*$.

Reguläre Ausdrücke – Beispiele

- $L := \{w \in \{0,1\}^* \mid w \text{ enthält 10 nicht als Teilwort}\} = 0^*1^*$, denn
  - Sei $w \in 0^*1^*$. Dann kommt in w nach keiner 1 eine 0.
    Also ist $w \in L$ und damit $0^*1^* \subseteq L$.
  - Sei $w \in L$. Dann kommen nach der ersten Eins keine Nullen mehr vor, d.h. $w = w'1\ldots1$, wobei w' keine 1 enthält.
    Also ist $w \in 0^*1^*$ und damit $L \subseteq 0^*1^*$.

Testen Sie sich:

Sei $\Sigma = \{0,1,2,3,4,5,6,7,8,9\} \cup \{,\} \cup \{.\}$.
Sei $L_{\mathbb{R}} \subseteq \Sigma^*$ die Sprache aller endlichen Dezimaldarstellungen von positiven reellen Zahlen.

Beispiele: $1.024,48 \in L_{\mathbb{R}}$ und $42 \in L_{\mathbb{R}}$
aber $1024,48 \notin L_{\mathbb{R}}$ und $042,00 \notin L_{\mathbb{R}}$ und $,,.123 \notin L_{\mathbb{R}}$

$\rightsquigarrow$ Finden Sie einen regulären Ausdruck für $L_{\mathbb{R}}$?

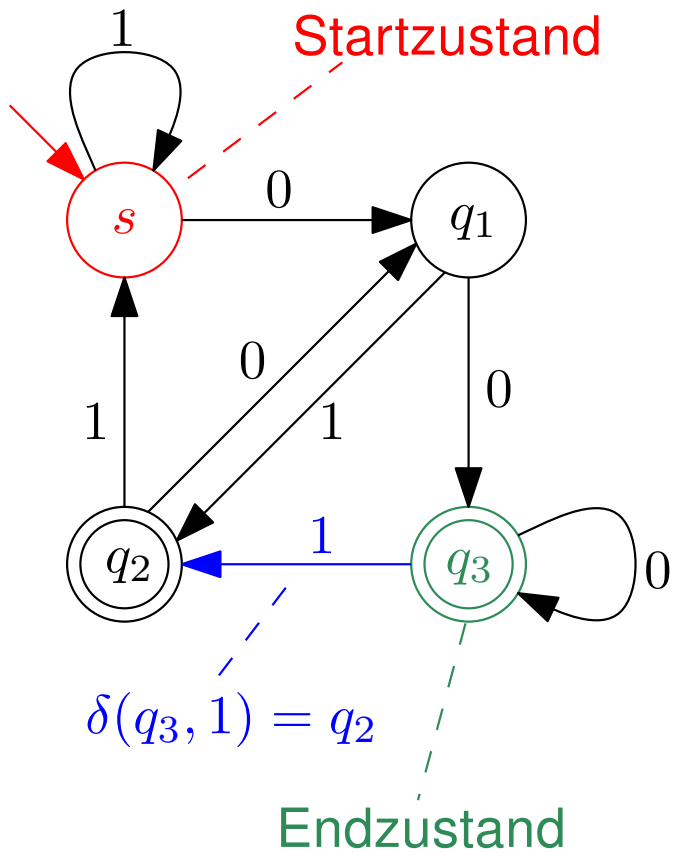
Endliche Automaten

- In GBI wurden Mealy- und Moore-Automaten behandelt.
- In TGI werden nur endliche Akzeptoren benötigt.

Endliche Automaten

Ein (deterministischer) endlicher Automat DEA $(Q, \Sigma, \delta, s, F)$ besteht aus:

- Q, einer endlichen Menge von *Zuständen*
- Σ, einer endlichen Menge von *Eingabesymbolen*
- $\delta\colon Q \times \Sigma \to Q$, einer *Übergangsfunktion*
- $s \in Q$, einem *Startzustand*
- $F \subseteq Q$, einer Menge von *Endzuständen*.



Endliche Automaten

Der Automat heißt

- endlich, da die Zustandsmenge (vgl. mit Speicher, Gedächtnis) endlich ist.
- deterministisch, da δ eine Funktion ist und der Automat somit in jedem Schritt eindeutig arbeitet. Es gibt keine Zufälligkeiten oder Wahlmöglichkeiten.

Was kann ein endlicher Automat?

- Gegeben ist eine Eingabe als endliche Folge von Eingabesymbolen. Der Automat entscheidet, ob die Eingabe zulässig ist oder nicht, indem er in einem Endzustand endet oder nicht.

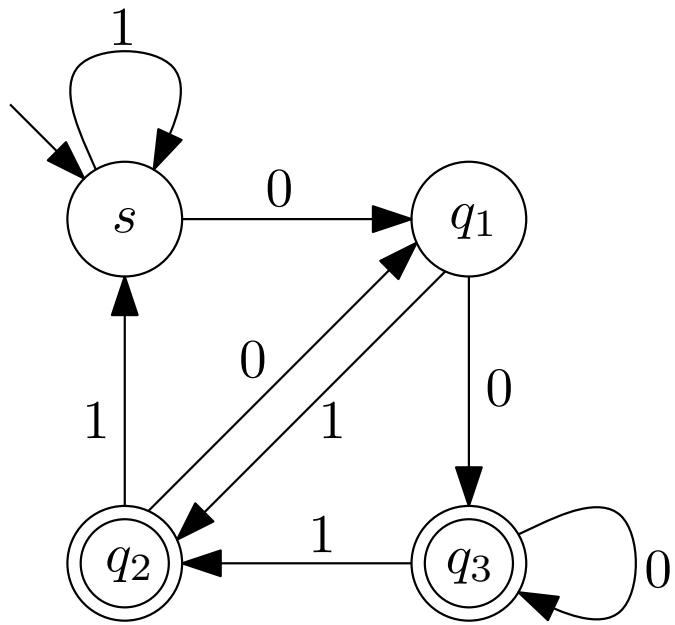
Endliche Automaten

- Ein endlicher Automat **erkennt** oder **akzeptiert** eine Sprache L, d.h. eine Menge von Wörtern über dem Alphabet Σ des Automaten, wenn er nach Abarbeitung eines Wortes w genau dann in einem Endzustand ist, wenn das Wort w in der Sprache L ist ($w \in L$).
- Eine formale Sprache heißt *endliche Automatensprache*, wenn es einen endlichen Automaten gibt, der sie erkennt.

Endliche Automaten

Der Automat erkennt
$w = 001$.

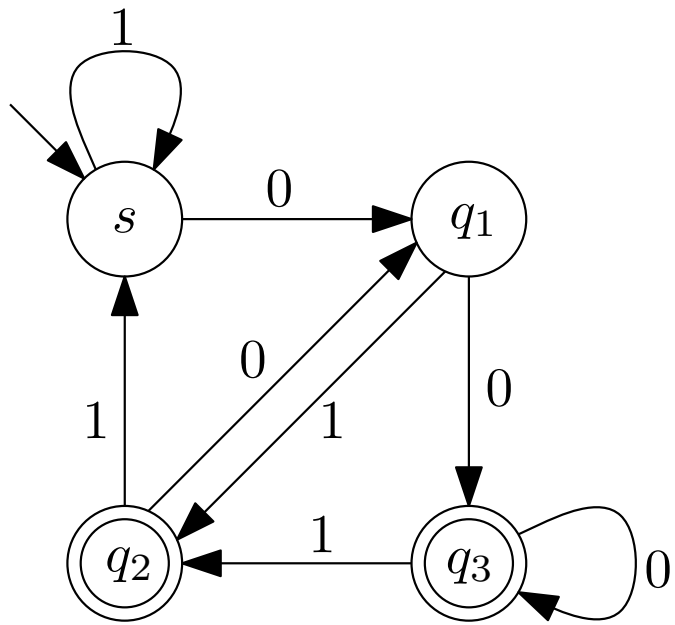
Der Automat erkennt nicht
$w = 110$.



Endliche Automaten

Der Automat erkennt
$w = 001$.

Der Automat erkennt nicht
$w = 110$.



Der Automat erkennt die Sprache L' aller Wörter, deren vorletztes Zeichen 0 ist:

$$L' = (0 \cup 1)^* 0 (0 \cup 1)$$

$\rightsquigarrow$ L' ist eine endliche Automatensprache.

Kontextfreie Grammatiken

Eine kontextfreie Grammatik $G = (\Sigma, V, S, R)$ ist gegeben durch

- ein endliches Alphabet Σ (auch Terminalalphabet genannt)
- eine endlichen Menge V mit $V \cap \Sigma = \emptyset$ von Variablen (auch Nichtterminale genannt)
- ein Startsymbol $S \in V$
- eine endlichen Menge von Ableitungsregeln R, d.h. durch eine Menge von Tupeln $(l, r) \in V \times (\Sigma \cup V)^*$ (auch Produktionen genannt)

Wir schreiben Produktionen auch in der Form $l \to r$.

Kontextfreie Grammatiken

Gegeben ist eine Regel $l \to r$.

- Wenn in einem Wort w das Zeichen l vorhanden ist, darf l in w durch r ersetzt werden.
- Wir schreiben $w \to z$, wenn w durch Anwendung *einer* Ableitungsregel in z umgewandelt werden kann.
- Wir schreiben $w \overset{*}{\to} z$, wenn w durch Anwendung einer, keiner oder *mehrerer* Ableitungsregeln in z umgewandelt werden kann.

Die von einer Grammatik $G = (\Sigma, V, S, R)$ erzeugte Sprache $L(G)$ ist die Menge aller Wörter $z \in \Sigma^*$, für die gilt: $S \overset{*}{\to} z$.

Kontextfreie Grammatiken – Beispiele

Gegeben ist die Grammatik $G = (\Sigma, V, S, R)$ mit

$$\begin{aligned} \Sigma &:= \{0, 1\} \\ V &:= \{S\} \\ R &:= \{S \to 01, S \to 0S1\}. \text{ Wir schreiben dafür kurz } \{S \to 01 \mid 0S1\}. \end{aligned}$$

Kontextfreie Grammatiken – Beispiele

Gegeben ist die Grammatik $G = (\Sigma, V, S, R)$ mit

$$\begin{aligned} \Sigma &:= \{0, 1\} \\ V &:= \{S\} \\ R &:= \{S \to 01, S \to 0S1\}. \text{ Wir schreiben dafür kurz } \{S \to 01 \mid 0S1\}. \end{aligned}$$

Welche Wörter erzeugt G, d.h. was ist $L(G)$?

Kontextfreie Grammatiken – Beispiele

Gegeben ist die Grammatik $G = (\Sigma, V, S, R)$ mit

$$\begin{aligned} \Sigma &:= \{0, 1\} \\ V &:= \{S\} \\ R &:= \{S \to 01, S \to 0S1\}. \text{ Wir schreiben dafür kurz } \{S \to 01 \mid 0S1\}. \end{aligned}$$

Welche Wörter erzeugt G, d.h. was ist $L(G)$?

$$\begin{aligned} S &\to 01 \mid 0S1 \to 01 \mid 0011 \mid 00S11 \\ &\to 01 \mid 0011 \mid 000111 \mid 000S111 \to \dots \end{aligned}$$

Kontextfreie Grammatiken – Beispiele

Gegeben ist die Grammatik $G = (\Sigma, V, S, R)$ mit

$$\begin{aligned} \Sigma &:= \{0, 1\} \\ V &:= \{S\} \\ R &:= \{S \to 01, S \to 0S1\}. \text{ Wir schreiben dafür kurz } \{S \to 01 \mid 0S1\}. \end{aligned}$$

Welche Wörter erzeugt G, d.h. was ist $L(G)$?

$$\begin{aligned} S &\to 01 \mid 0S1 \to 01 \mid 0011 \mid 00S11 \\ &\to 01 \mid 0011 \mid 000111 \mid 000S111 \to \dots \end{aligned}$$

Es ist $L(G) = \{0^n 1^n \mid n \in \mathbb{N}\}$.

Kontextfreie Grammatiken – Beispiele

Welche Grammatik erzeugt die Sprache über dem Alphabet $\Sigma = \{0, 1\}$, deren vorletztes Zeichen 0 ist?

Kontextfreie Grammatiken – Beispiele

Welche Grammatik erzeugt die Sprache über dem Alphabet $\Sigma = \{0, 1\}$, deren vorletztes Zeichen 0 ist?

Grammatik $G = (\Sigma, V, S, R)$ mit

$$\begin{aligned} \Sigma &:= \{0, 1\} \\ V &:= \{S, A, B\} \\ R &:= \{S \to A0|A1, \quad A \to B0, \quad B \to \varepsilon|B0|B1\} \end{aligned}$$

Ableitung:

$$S \to A0|A1 \to B00|B01 \to \ldots$$

Kontextfreie Grammatiken

In TGI werden wir Grammatiken kennenlernen, deren Produktionen *mächtiger* sind als die von kontextfreien Grammatiken.

Wir werden auch Sprachen kennenlernen, die nicht von Grammatiken erzeugt werden können.