

Formale Sprachen und Automatentheorie II

Ergänzende Inhalte der theoretischen Informatik

Roman Wetenkamp

19. April 2022

Inhaltsverzeichnis

1	Sprachklassen	4
1.1	Die CHOMSKY-Hierarchie	4
1.2	Typ 3: Reguläre Grammatiken	4
1.3	Typ 2: Kontextfreie Grammatiken	6
1.4	Typ 1: Kontextsensitive Grammatiken	6
1.5	Typ 0: Rekursiv aufzählbare Grammatiken	6

Vorwort

Liebe*r Leser*in,

die Theorie der Informatik – sie begeistert Studierende vom ersten Semester an! Vielleicht unterschreiben Sie das, vielleicht interpretieren Sie es als ironisch – in beiden Fällen sind Sie herzlich willkommen. Dieses Skript ist das Ergebnis meiner Beschäftigung mit weiteren, ergänzenden Themen der theoretischen Informatik, die über den Stoff der Vorlesung zu Formalen Sprachen und Automaten hinaus gingen. Damit versuche ich einerseits, mögliche Defizite meines dualen Studiums im Vergleich zu einem regulären Universitätsstudiums auszugleichen und andererseits ermöglicht es mir auch, mich didaktisch auszuprobieren. In meiner fernen Zielvorstellung bietet es Ihnen ebenfalls die Möglichkeit, sich weitere Inhalte der theoretischen Informatik anzueignen – falls Sie dann noch Hochschullehrende finden sollten, die bereit sind, Ihnen für Ihre Arbeit ein paar ECTS-Punkte anzuerkennen, schließt dies vielleicht genau eine der Lücken, die sich aus dem Modulplan Ihres Bachelors und den Zulassungsvoraussetzungen der Universitäten für Informatik-Masterstudiengänge ergeben. Ich freue mich sehr über Ihre Erfahrungsberichte!

Doch nun zu den Inhalten des Skriptes: Zunächst ordnen wir die Erkenntnisse zu regulären Sprachen und kontextfreien Grammatiken in die Hierarchie der Sprachen (Chomsky) ein. Anschließend beschäftigen wir uns mit Kellerautomaten und widmen uns der Entscheidbarkeit und Berechenbarkeitstheorie.

Didaktisches Konzept Der Zugang zu den Stoffinhalten folgt folgender Trias:

1. Verständliche, normalsprachliche Motivation (auch durch Bezüge zur Praxis)
2. Formale Beschreibung und gegebenenfalls Beweise
3. Anwendung und Vertiefung durch Aufgaben und Hintergrundinformationen

Zu den Aufgaben befinden sich am Ende des Skriptes Lösungen. Das Skript ist für das Selbststudium konzipiert und wirkt daher stellenweise länglich. Durch farbliche Hervorhebungen versuche ich, vernachlässigbare Textpassagen kenntlich zu machen.

Projekt „Masterplan“ Ich habe während meines dualen Bachelorstudiums gemerkt, dass ich mein Studium nach dem Bachelor an einer Universität fortsetzen möchte. Nun unterscheiden sich die Modulpläne von HAWs und Universitäten jedoch stellenweise stark, sodass viele Universitäten spezielle Voraussetzungen für ein konsekutives Masterstudium angeben und so die Durchlässigkeit des Hochschulbildungssystems hemmen. Inwiefern dies auf deutsche Hochschulen zutrifft und inwieweit es Bachelorabsolventen von HAWs oder Dualen Hochschulen in ihrer Studienwahl einschränkt, sind Forschungsgegenstände des Projektes „Masterplan“. Die Ergebnisse des Projektes finden Sie unter Masterplan.

Mitwirken Ich freue mich jederzeit über Hinweise zu Fehlern oder Ungenauigkeiten. Auch, falls Sie weitere Aufgaben oder Inhalte anfügen möchten, sind Sie dazu herzlich eingeladen! Dieses Skript wird in einem GitHub-Repository verwaltet. Vielen Dank für Ihre Unterstützung!

Viel Erfolg!

Ihr Roman Wetenkamp
Mannheim, den 19. April 2022

1 Sprachklassen

Sprachen sind spannend. In unseren ersten Lebensjahren lernen wir die Sprache(n) unserer Eltern, noch so mühelos wie nie wieder. In der Schulzeit kommen Fremdsprachen hinzu, die wir mal mehr, mal weniger gerne lernen. Als Jugendliche experimentieren wir mit Sprache: Wir verändern sie, um uns von Erwachsenen abzugrenzen, nutzen Sie als Ausdruck unserer kulturellen Vielfalt. Später lernen Sie eine Programmiersprache und fügen Sie vielleicht scherzhaft der Liste Ihrer Sprachkenntnisse hinzu. Doch das ist weit mehr als ein plumper Nerdwitz: Diese und jene Sprachen eint viel!

Definition 1.0.1.

Eine (formale) Sprache ist eine Menge von Zeichenketten, die aus den Symbolen eines beliebigen Alphabets aufgebaut sind.

Wenn Sie an das Sprachenlernen im schulischen Kontext denken, vergleichen Sie diese Definition vielleicht mit dem Vokabellernen. Demnach wäre Englisch die Menge aller englischen Wörter. Nun werden Sie vielleicht protestieren: Die bloße Kenntnis aller englischen Begriffe lässt Sie noch lange kein korrektes Englisch sprechen. Dafür gibt es die Grammatiken – Regelwerke, die aussagen, wann ein Satz korrekt oder inkorrekt ist.

1.1 Die Chomsky-Hierarchie

Wenden wir uns nun wieder den formalen Sprachen zu, müssen wir zunächst feststellen, dass es verschiedene Typen von Grammatiken gibt. In Ihrer Vorlesung haben Sie höchstwahrscheinlich kontextfreie Grammatiken eingeführt – Diese Klassen erweitern wir nun entsprechend einer Hierarchie von Noam Chomsky, der Chomsky-Hierarchie. Zunächst wiederholen wir reguläre Sprachen und kontextfreie Grammatiken, bevor wir zu kontextsensitiven und rekursiv aufzählbaren Grammatiken gelangen. Stellen Sie sich diesen Aufbau vor wie in der Schule: Sie beginnen mit natürlichen Zahlen, wagen irgendwann einen Schritt in die negative Richtung hin zu den ganzen Zahlen, dann treten Sie zwischen einzelne Stufen und erreichen rationale Zahlen, bevor Sie im Wahn die irrationalen, reellen und komplexen Zahlen einführen. Ähnlich dieser Teilmengenbeziehungen konstruieren wir nun auch die Sprachklassen der Chomsky-Hierarchie.

1.2 Typ 3: Reguläre Grammatiken

Zu Beginn dieses Themengebiets der Theoretischen Informatik widmeten wir uns Automaten. Wir unterschieden zunächst deterministische endliche (DEA, *engl.*: DFA) und nicht-deterministische endliche Automaten (NEA, *engl.*: NFA). Ebenso definierten wir reguläre Sprachen und stellten die Äquivalenz der drei Objekte fest. Nun ist ein kleiner Vorgriff von Nöten: Wir werden reguläre Grammatiken anhand der nächstgrößeren Klasse, der kontextfreien Grammatiken, definieren. So werden reguläre Grammatiken zu kontextfreien Grammatiken, die spezielle Bedingungen erfüllen.

Definition 1.2.1.

Eine **kontextfreie Grammatik** G ist ein 4-Tupel $G = (V, T, P, S)$, wobei gilt:

- V bezeichnet eine Menge von Variablen.
- T bezeichnet eine Menge von Terminalen.
- P bezeichnet eine Menge von Produktionen (also Regeln), wobei eine Produktion A der Form $A \rightarrow \alpha$ mit $\alpha \in (V \cup T)^*$ ist.

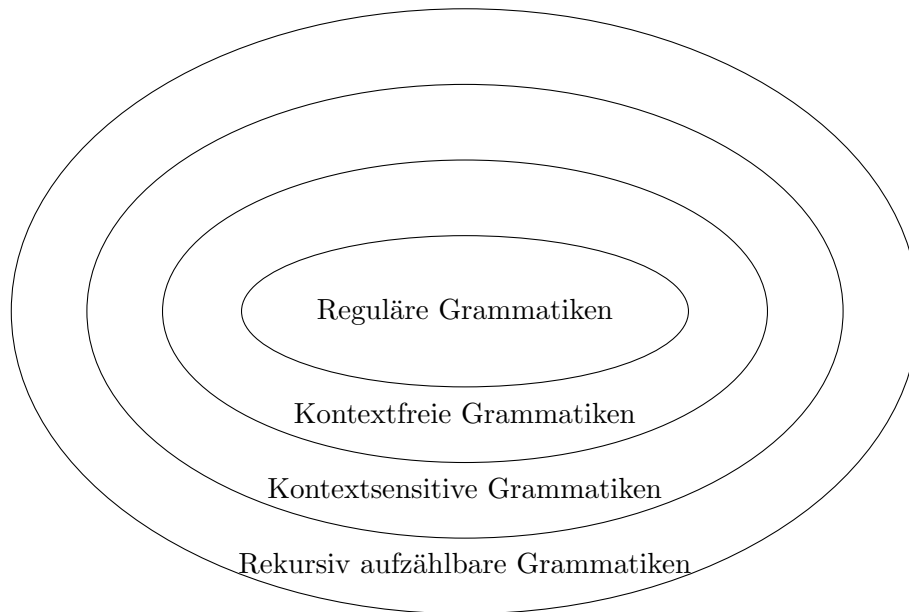


Abbildung 1.1: Chomsky-Hierarchie formaler Sprachen (Quelle: Eigene Darstellung)

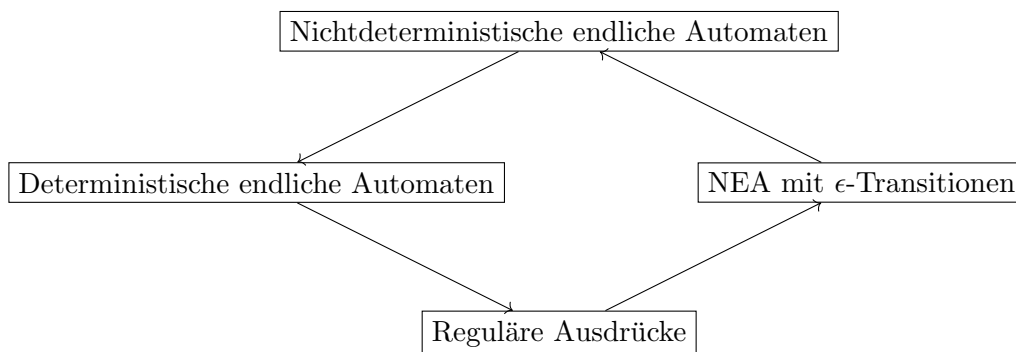


Abbildung 1.2: Äquivalenzkonstruktionen (Quelle: [Hopcroft])

- S bezeichnet ein Startsymbol. $S \in V$.

Es gilt: $V \cap T = \emptyset$.

Betrachten wir beispielsweise eine Grammatik, die den Aufbau von Log-Einträgen beschreibt:

Beispiel 1.2.1.

$G = (\{\text{Start, Type, Text, ErrorCode, TimeStamp}\}, \{\text{INFO, WARNING, ERROR, :, ., , --, (,)}, \text{NUMBER, STRING}\}, P, \text{Start})$

Wobei P aus folgenden Regeln besteht:

$\text{Start} \rightarrow \text{TimeStamp} \text{ -- Type : Text (ErrorCode)}$
 $\text{Type} \rightarrow \text{INFO} \mid \text{WARNING} \mid \text{ERROR}$
 $\text{TimeStamp} \rightarrow \text{NUMBER}.\text{NUMBER}.\text{NUMBER} \text{ NUMBER}:\text{NUMBER}:\text{NUMBER}$
 $\text{Text} \rightarrow \text{STRING Text} \mid \text{NUMBER Text} \mid \epsilon$
 $\text{ErrorCode} \rightarrow \text{NUMBER} \mid \epsilon$

Häufig wird die Grammatik nicht mehr als Tupel angegeben, sondern lediglich die Produktionen. Dies ist dann möglich, wenn die Definition entsprechend beschränkt wird – dann ist das Startsymbol direkt ablesbar und ebenso auch die Mengen der Terminale und Variablen.

Kommen wir nun zu **regulären Grammatiken**.

Definition 1.2.2.

Eine **kontextfreie Grammatik** ist genau dann **regulär**, falls sie rechts-linear oder links-lineare ist. Rechts-linear ist eine Grammatik genau dann, wenn ihre Produktionen von der Form

$$A \rightarrow w B$$

oder

$$A \rightarrow w$$

und A, B Variablen sind und w eine Zeichenkette von Terminalen ist. Entsprechend definieren wir die Links-Linearität: Falls alle Produktionen der Grammatik der Form

$$A \rightarrow B w$$

oder

$$A \rightarrow w$$

sind, so ist die Grammatik links-linear.

1.3 Typ 2: Kontextfreie Grammatiken**1.4 Typ 1: Kontextsensitive Grammatiken****1.5 Typ 0: Rekursiv aufzählbare Grammatiken**