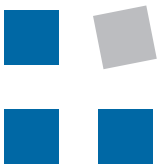
Zusammenfassung Math2I

Mathematische Grundlagen der Informatik 2

Emanuel Duss
emanuel.duss@gmail.com

12. April 2013



HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Zusammenfassung Math2I
Mathematische Grundlagen der Informatik 2

Dieses Dokument basiert auf der Vorlesung „Mathematische Grundlagen der Informatik 2“ der HSR (Hochschule für Technik Rapperswil) vom FS 2013.

Version 60ca18b vom 2013-04-01.

MITMACHEN
Falls Du an diesem Dokument mitarbeiten willst, kannst Du das Dokument auf GitHub unter http://github.com/HSR-Stud/Math2I forken.

MITWIRKENDE
Folgende Personen haben an diesem Dokument mitgewirkt:
Emanuel Duss (eduss@hsr.ch)

LIZENZ

Inhaltsverzeichnis

1 Sprachen

1.1 Grundbegriffe

- Ein Alphabet $\Sigma = \{0, 1\}$ ist eine endliche, nicht leere Menge
- Ein Wort w der Länge n über dem Alphabet Σ ist ein n-Tupel $w \in \Sigma^n$.
- Das Wort mit der Länge 0 wird als leeres Wort (ϵ) bezeichnet.
- Eine Zeichenkette ist ein n-Tupel von Elementen aus Σ
- Länge eines Wortes: $|w| = n$ falls $w \in \Sigma^n$
- Anzahl Buchstaben: $|w|_a =$ Anzahl a in w
- Eine Sprache L über dem Alphabet Σ ist eine Teilmenge von Σ^*.

2 Endliche Automaten und reguläre Sprachen

2.1 Deterministische endliche Automaten (DEA)

2.1.1 Definition

Deterministisch bedeutet, dass man in jedem Fall weiss, was passiert.

$$A = (Q, \Sigma, \delta, q_0, F)$$

- A: Automat
- Q: Mögliche Zustände
- Σ: Alphabet
- δ: Übergangsfunktion
- q_0: Startzustand
- F: Akzeptierzustände

2.2 Reguläre Sprache

Eine Sprache ist regulär, wenn es einen passenden DEA dazu gibt.

2.2.1 Minimaler Automat

Automaten können in eine Normalform gebracht werden:

1. Akzeptierzustände und andere sind nicht äquivalent.
2. Alle Paare untersuchen: Kommt man mit dem selben Zeichen in unterschiedliche Zustände, so sind diese Zustände nicht äquivalent.
3. Wiederholen, bis man nur noch äquivalente hat.

2.2.2 Pumping Lemma

Mit dem Pumping Lemma kann man nur zeigen, dass eine Sprache nicht regulär ist.

2.3 Nicht deterministische Automaten (NEA)

Beim NEA gibt es zu jedem Ausgangszustand und jedem Zeichen entweder keine, eine oder mehrere Zielzustände.

2.3.1 NEA zu DEA umwandeln

Mit dem Könnte-Automat kann man einen NEA zu einem DEA umwandeln.

- Zustände mit allen anderen Kombinieren (gibt 2^n Zustände)
- Startzustand festlegen
- Zustände, welche Zielzustände enthalten, so markieren
- Mögliche Zustände einzeichnen: Alle Kombinationen durchgehen, dabei sind alle Zustände möglich, wo der Pfeil hinziegen "könnte''. (alle Zustände nicht möglich = q_0)
- Nicht erreichbare Zustände können gestrichen werden.

2.4 Mengenoperationen

- Bei $L_1 \cup L_2$ gibt es einen ϵ-Übergang, der auf beide Automaten verweist.
- Bei $L_2 \cap L_2$ werden beide Automaten nacheinander durchlaufen.

2.4.1 Produkteautomat

Mit dem Kartesischen Produkt kann man aus zwei Automaten einen Automaten machen, welcher Zustände wie $L_1 \cup L_2$ abbildet.

2.5 Reguläre Ausdrücke

2.5.1 DEA zu regulärem Ausdruck

Um aus einem DEA einen regulären Ausdruck zu erstellen, kann man ein VNEA (Verallgemeinerten NEA) zeichnen, welcher reguläre Ausdrücke auf den Pfeilen hat.

- Ein neuen Start- und Akzeptierzustand mit dem Automaten dazwischen zeichnen.
- Alle Zwischenzustände entfernen (Jeden einzelnen Zustand eliminieren, bis nur noch der neue Start- und Akzeptierzustand vorhanden ist.

2.6 Reguläre Operationen

- Alternative: $L_1 \cap L_2$
- Verkettung: $L_1 L_2$
- Wiederholung (*-Operation): L_1*

Alle Sprachen lassen sich aus einzelnen Zeichen mit regulären Operationen zusammensetzen. Zu jedem DEA gibt es einen regulären Ausdruck.

Zu jedem regulären Ausdruck gibt es einen NEA, welchen man in einzelne Zeichen zerlegen kann. Daraus lässt sich wiederum einen DEA bauen.

3 Stackautomaten und kontextfreie Grammatik

3.1 Kontextfreie Sprachen

Mit regulären Ausdrücken kann z. B. keine Klammern prüfen. Mit der kontextfreien Grammatik ist das hingegen möglich.

3.1.1 Kontextfreie Grammatiken

$$G = \{V, \Sigma, R, S\}$$

- V: Endliche menge von Variablen ($V = \{A\}$)
- Σ: Alphabet ($\Sigma = \{(,)\}$)
- R: Regeln
- S: Start

3.1.2 Kontextfreie Sprachen

Jede reguläre Sprache L hat eine kontextfreie Grammatik G mit $L(G) = L$. Reguläre Sprache $\in$ kontextfreie Sprache. Hat eine Sprache einen regulären Ausdruck, ist sie kontextfrei.

3.1.3 Reguläre Operationen

L_1 und L_2 sind kontextfreie Sprachen:

- $L_1 = L(G_1); G_1 = \{V_1, \Sigma, R_1, S_1\}$
- $L_2 = L(G_2); G_2 = \{V_2, \Sigma, R_2, S_2\}$

Alternative $G_1 \cup G_2 = \{V_1 \cup V_2 \cup \{S\}, \Sigma, R_1 \cup R_2 \cup \{S \to S_1 | S_2\}, S\}$

Verkettung $G_1 G_2 = \{V_1 \cup V_2 \cup \{S\}, \Sigma, R_1 \cup R_2 \cup \{S \to S_1 S_2\}, S\}$

***-Operation** $G_1* = \{V_1 \cup \{S_0\}, \Sigma, R_1 \cup \{S_0 \to \epsilon | S_0 S\}, S_0\}$

3.1.4 Chomsky Normalform

Jede kontextfreie Grammatik (CFG) lässt sich in die Chomsky Normalform bringen.
Dabei sind nur noch folgende Regeln zugelassen: $A \to BC$ (Variable $\to$ Paar von Variablen), $A \to a$ (Terminalsymbol) und $S \to \epsilon$.

- Kommt S auf der rechten Seite vor, neue Startvariable mit $S_0 \to S$ definieren.
- ϵ-Regel eliminieren: Aus $A \to \epsilon, B \to AC$ wird $B \to AC, B \to C$.
- Unit Rules eliminieren: Aus $A \to B, B \to CD$ wird $A \to C, B \to CD$.
- Verketung eliminieren: Aus $A \to ABA$ wird $A \to AT, T \to BA$.
- Terminalsymbole eliminieren: Aus $A \to 0B$ wird $A \to NB, N \to 0$.

3.1.5 Parse Tree

Zwei Ableitungen eines Wortes w einer kontextfreien Sprache $L(G)$ heissen äquivalent, wenn sie den gleichen Ableitungsbaum haben. Eine Sprache mit mehreren Ableitungsbäumen heisst mehrdeutig.

3.2 Stackautomaten

Jede kontextfreie Grammatik hat ein Stackautomat $P = (Q, \Sigma, T, \delta, q_0, F)$ mit dem Stackalphabet T.

3.3 Nicht kontextfreie Sprachen

Nicht kontextfreie Sprachen (z. B. $L = \{a^n b^n c^n | n \in \mathbb{N}\}$ können mit dem pumping Lemma für kontextfreie Sprachen widerlegt werden.

Das Pumping Lemma sagt, es gibt eine Pumping Length (N), so dass jedes Wort $w \in L$ mit $|w| \geq N$ zerlegt werden kann in fünf Teile $w = uvxyz$

- $|vy| \geq 0$
- $|vxy| \leq N$

- $uv^kxy^kz \in L$ für alle $k \in \mathbb{N}$