Nachklausurtutorium Theoretische Informatik

Karl Jochen Micheel & Christopher Happe

Sommersemester 2019

Überblick

1. Grundlagen
2. Reguläre Sprachen
3. Kontextfreie Sprachen
4. Kontextsensitive und L0 Sprachen
5. Berechenbarkeit
6. Primitive und Partielle Rekursion
7. Entscheidbarkeit

Inhaltsverzeichnis

- Pumping Lemma für reguläre Sprachen
- Myhill Nerode Relation
- Minimalautomaten
- Abschlusseigenschaften

Das Pumping-Lemma für reguläre Sprachen

- **Wird nur benutzt, um nachzuweisen, dass eine Sprache NICHT regulär ist und FÜR NICHTS ANDERES!**

Beweis der nicht-regularität mithilfe des Pumping-Lemmas

- Angenommen $L \in REG$:
- *Dann* $\exists\, n \geq 1 : \forall\, x \in L$ *mit* $|x| \geq n$:
- *x lässt sich zerlegen in* $x = uvw$ *mit:*
  1. $|uv| \leq n$,
  2. $|v| \geq 1$,
  3. $(\forall i \geq 0)$: $uv^iw \in L$
- Finde ein x, für das 1. und 2. aber nicht 3. gilt.

Beispiele des Pumping-Lemmas für reguläre Sprachen

- $L_1 = \{a^n b^n \mid n \geq 0\}$
- $L_2 = \{a^x \mid x = 2^n, n \geq 0\}$
- *Aufgabe:* $L_3 = \{a^n b^m \mid n < m, n,m \geq 0\}$
- *Aufgabe:* $L_4 = \{a^x \mid x \text{ ist Quadratzahl}\}$

Die Myhill-Nerode-Relation

- Gegeben:
  - Eine Sprache L
  - Zwei Wörter $x, y \in \Sigma^*$ (müssen nicht Teil der Sprache sein
- $x\ R_L\ y$ (R_L ist die Myhill-Nerode-Relation) g.d.w.:
- $\forall z \in \Sigma^*: xz \in L \Leftrightarrow yz \in L$

Beispiele der Myhill-Nerode-Relation

- $L = \{a^n b^m \mid n, m \geq 1\}$
- $x = aa$, $y = a$
- $x R_L y$
- $x = aa$, $y = ab$
- $x R_L y$ *gilt nicht!*
- $x = bab$, $y = b$
- $x R_L y$

Der Satz von Myhill und Nerode

- $[x] = \{y \in \Sigma^* \mid x R_L y\}$
- $Index(R_L) = ||\{[x] \mid x R_L y\}||$
- $L \in REG \Leftrightarrow Index(R_L) < \infty$
- Beispiel zu $L_1 = \{a^n b^m \mid n, m \geq 1\}$ und $L_2 = \{a^n b^n \mid n \geq 1\}$ an der Tafel

Minimalautomaten

- Ein DFA *M* mit totaler Überfuhrungsfunktion mit minimaler Anzahl Zuständen
- D.h. es gibt keinen anderen totalen DFA, der die gleiche Sprache erzeugt und weniger Zustände hat
- Für eine Sprache *L* ist die Anzahl Zustände eines Minimalautomaten gleich $Index(R_L)$

Minimalautomat zu gegebenem DFA

- Gegeben: DFA $M = (\Sigma, Z, \delta, z_0, F)$ mit totaler Überführungsfunktion
  1. Entferne alle von z_0 aus nicht erreichbare Zustände
  2. Erstelle eine Tabelle aller Zustandspaare $\{z, z'\}$ von M mit $z \neq z'$
  3. Markiere alle Paare mit $z \in F \Leftrightarrow z' \notin F$
  4. Sei $\{z, z'\}$ unmarkiert. Prüfe für alle $a \in \Sigma$, ob mind. ein $\{\delta(z, a), \delta(z', a)\}$ markiert ist. Falls ja markiere $\{z, z'\}$

Minimalautomat zu gegebenem DFA

5. Widerhole Schritt 4, bis keine Änderungen mehr eintreten
6. Bilde maximale Mengen paarweise nicht disjunkter unmarkierter Zustandspaar und verschmelze jeweils alle Zustände einer Menge zu einem neuen Zustand

- Beispiel und Aufgabe an der Tafel

Abschlusseigenschaften regulärer Sprachen

- REG ist abgeschlossen unter:
- Vereinigung
- Komplement
- Schnitt
- Differenz
- Konkatenation
- Iteration
- Spiegelung

Beispiel Abschlusseigenschaften REG

- Betrachte $L_1 = \{a^n \mid n \geq 0\}$, $L_2 = \{b^n \mid n \geq 0\}$
- Die Konkatenation $L_1L_2 = \{a^nb^m \mid n, m \geq 0\}$ ist regulär
- Der Schnitt $L_1 \cap L_2 = \emptyset$ ist regulär

Fragen?