Formale Sprachen B
Beschreibungskomplexität

Benedikt Lüken-Winkels

6. Juni 2018

Inhaltsverzeichnis

1 Allgemeines

1.1 Chomsky Hierarchie

- Typ-0: rekursiv (aufzählbar) (beliebige formale Grammatik)
- Typ-1: Kontextsensitive Grammatik
- Typ-2: Kontextfreie Grammatik (CFG)
- Typ-3: Reguläre Grammatik

1.2 Begriffsklärung

1.2.1 rekursiv (aufzählbar) Typ-0

- L ist *rekursiv aufzählbar* bzw. *semientscheidbar*, wenn es eine Turingmaschine gibt, die alle $w \in L$ akzeptiert, aber keine Wörter, die nicht in L liegen.
- $L \subseteq \Sigma^*$ ist *rekursiv* bzw. *entscheidbar*, wenn es eine Turingmaschine gibt, die für jede Eingabe $w \in \Sigma^*$ hält und jedes w genau dann akzeptiert, wenn $w \in L$.

Beispiel: Das Halteproblem ist rekursiv aufzählbar, aber nicht rekursiv.

Beweisskizze

1.2.2 kontextsensitiv Typ-1

- G ist *kontextsensitiv*, wenn die Regeln der Form $\alpha A \beta \rightarrow \alpha \gamma \beta$ sind (wobei γ entweder NT oder T sein muss; α, β dürfen leer sein) oder $S \rightarrow \varepsilon$ (dann darf S allerdings nicht auf eine rechten Regelseite auftauchen), also
  - NT-Symbole auf der linken Regelseite,
  - keine Produktionsregel außer das Startsymbol darf ε erzeugen.

1.2.3 kontextfrei Typ-2

- G ist *kontextfrei*, wenn die Regeln der Form $A \rightarrow \alpha$ (wobei $A \in NT$ und α eine beliebige Folge von NT und T) oder $S \rightarrow \varepsilon$ (dann darf S allerdings nicht auf eine rechten Regelseite auftauchen) sind

1.3 Pumping Lemma

1.4 Push Down Automate (PDA)

2 1. Vorlesung

2.1 Ziel von Beschreibungskomplexität

- Wie kompakt können 'Gegenstände' oder 'Objekte' ausgedrückt werden?

- DEA als Beispiel mit kleinstmöglicher Zustandsmenge
- Minimierungsalgortihmen für Automaten
- Datenkomprimierung (Lempel-Ziv-Welch-Algorithmus, verlustfreies Komprimierungsverfahren)

$\Rightarrow$ knappe Beschreibung für Objekte für effizientere Arbeit

2.2 Beschreibungssystem

Ein Beschreibungssystem S besteht aus einer Menge endlicher Deskriptoren, sodass jeder Deskriptor $D \in S$ eine formale Sprache $L(D)$ beschreibt. Aus D kann man $alph(D)$ ablesen, sodass $L(D) \subseteq (alph(D))*$. Die durch S beschriebene Sprachfamilie $\ell(S)$ umgekehrt $L : S(L)$ beschreibt L.
Es existieren verschiedene Beschreiber für eine Sprache, eventuell abzählbar unendlich viele Möglichkeiten.

- Umsetzung einer Aufgabe/Funktion hat beliebig viele Implementierungen
- Verschiedene reguläre Ausdrücke für die selbe Sprache

2.3 Beschreibungsmaße

Natürliches Maß: # Bits einer Beschreibung
$\rightarrow$ Frage: Wie wird kodiert?
Alternativ: Betrachte die Struktur der Deskriptoren
Ist S_{mass} ein Beschreibungsmaß für S_{sys}, so meint
$S_{mass}(L) = min\{S_{mass}(D)|D \in S_{sys}, L(D) = L$

2.3.1 Beispiel reguläre Ausdrücke

- $\emptyset, \lambda, a \in \sum$ reguläre Ausdrücke
- Wenn r, s reguläre Ausdrücke, so auch $(r + s), (r \cdot s), (r*)$

Größenmaße

- Länge des Strings $|(\emptyset) * | = 4 > 1 = |\lambda|$
- $rpn(n)$ reversed polish notation
  - $r = ((0 + ((1 \cdot 0)*)) \cdot (1 + \lambda))$

  $\rightarrow$ $rpn(r) = |010 \cdot * + 1\lambda + \cdot| = 10$
- $a - width(r)$: Anzahl von Vorkommen von Zeichen aus $\sum$ in r
- Ressourcen: Nonterminalsymbole zählen zum Beispiel

3 2. und 3. Vorlesung

3.1 Typ-0 Grammatiken

$G = (N, T, P, S)$
N = non-Terminalsymbole; T = Terminalsymbole; S Startsymbole
$P \subseteq (N \cup T) * N(N \cup T) * \times (N \cup T)*$
Wenn alle Regeln bis auf Eine kontextfrei sind können alle rekursiv aufzählbaren formalen Sprachen dargestellt werden. Typ-0 Grammatiken können durch durch Turingmaschinen dargestellt werden.
Natürliche Maße für Turingmaschinen

- Laufzeit als Maß kann unentscheidbar sein, weil dynamisch
- Bandalphabet
- Anzahl der Bänder
- Grad des Nichtdeterminismus, zB wie oft gibt es nichtdeterministische Übergänge?

Fragen: In wie weit können Maßzahlen eingeschränkt werden, ohne das Modell zu ändern? Was sagen die Maße über die Mächtigkeit der Maschine aus?

3.1.1 Universelle Turingmaschine mit 2 Zuständen

Die Übergänge teilen sich in eine Kopierphase und einen Simulationszyklus: '+' bedeutet, dass der Simulationszyklus läuft; 'α' beendet die Kopierphase.
Zu den Regeln: Es existiert eine Bouncing-Regel zwischen den Regeln 1, 3 und Regeln 2,4, die für die Codierung auf dem Band sorgt.

3.2 Begriffseinführungen

Ziel ist eine minimale Beschreibung von Typ-0 Grammatiken

3.2.1 1-a-Transitor

Ein 1-a-Transitor $\tau = (Q, \sigma, \delta, H, q_I, q_F)$ ist ein endlicher übersetzender Automat.
σ = Eingabe, δ = Ausgabe, Q = Zustandsmenge, $H \subseteq Q \times \sigma \times \delta \times Q \Rightarrow$ nicht deterministisch.

- Allgemeiner, als Mealy und Moore Automaten

3.2.2 G-Systeme

Ein G-System $G = (N, T, P, S)$ wobei $P = (K, V, V, H, q_I, q_F)$ ein 1-a-Transitor ist und das Eingabe gleich dem Ausgabealphabet ist mit $V = N \cup T$.
Typ-0-Sprachen können durch G-Systeme simuliert werden.