Grundlagen der Theoretischen Informatik

15. Juli 2020

Terminologie

- Alphabete, Wörter, Operationen auf Wörtern
  (Konkatenation, i-te Potenz, Reverse)
- Sprachen, Operationen auf Sprachen
  (Konkatenation, i-te Potenz, Reverse, Kleene Hülle)
- Reguläre Ausdrücke

- **Grammatik** (Definition)
  - Ableitung, Rechnung; Erzeugte Sprache, Äquivalenz
  - Beispiele: Dycksprache

- **Sprachen und Darstellung von Problemen**

- **Die Chomsky-Hierarchie**
  - Rechtlineare Grammatik; Kontextfreie Grammatik; Kontextsensitive Grammatik; Beschränkte Grammatik
  - Sprachklassen

- **Abzählbarkeit**; Beispiele von abzählbare Mengen: $\mathbb{N}$, $\mathbb{Q}$, alle endlichen Mengen, die Vereinigung zweier abzählbarer Mengen, die Vereinigung abzählbar vieler abzählbarer Mengen.

- **Überabzählbarkeit**
  - Die Menge $\mathbb{R}$ der rellen Zahlen ist überabzählbar.
  - Es gibt überabzählbar viele Funktionen $f : \mathbb{N} \to \mathbb{N}$.
  - Es gibt überabzählbar viele Funktionen $f : \mathbb{N} \to \{0, 1\}$.

- Gegeben: Signatur Σ, endlich oder abzählbar unendlich.
  - Σ^* ist abzählbar unendlich.
  - Die Menge aller Grammatiken über Σ ist abzählbar unendlich.
  - Es gibt (nur) abzählbar viele Algorithmen.
  - Die Menge der Sprachen über Σ ist überabzählbar.
  - Nicht jede Sprache kann durch eine Grammatik dargestellt werden.

Endliche Automaten

- **Determinierte endliche Automaten (DEAs)**;
  Darstellung als Graph; Übergangsfunktion; Erweiterung δ^*
- Von einem endlichen Automaten akzeptierte Sprache; RAT

- **Indeterminierte endliche Automaten (NDEAs)** ;
  Darstellung als Graph; Übergangsrelation; Erweiterung Δ^*
- Von einem NDEA akzeptierte Sprache
- Gleichmächtigkeit von DEAs und NDEAs:
  Eine Sprache ist rational (es gibt einen determinierten endlichen Automaten, der sie akzeptiert) genau dann, wenn es einen indeterminierten endlichen Automaten gibt, der sie akzeptiert.
  Konstruktion!

- **Automaten mit ε-Kanten**; Akzeptierte Sprache
- Gleichmächtigkeit von Automaten mit ε-Kanten und NDEAs
  Konstruktion!

- Satz von Kleene: **RAT** $= \mathbf{L_3}$
  Konstruktionen!
- **Pumping-Lemma** (beide Varianten);
  Anwendungen (Nichtregularitätsbeweise)

- **Abschlusseigenschafte von RAT**
  (Komplement, Vereinigung, Konkatenation, Kleene Hülle, Durchschnitt)
  Konstruktion der Automaten!

  Anwendungen (z.B. Nichtregularitätsbeweise)

- **Wortprobleme**

- **Hauptsatz von Kleene**
  Konstruktionen!

Kellerautomaten und kontextfreie Sprachen

- Definition: Kontextfreie Grammatik
- Ableitungsbaum zu einer Grammatik; Linksableitung, Rechtsableitung; Mehrdeutigkeit; Inhärente Mehrdeutigkeit
- Definition: erreichbare/co-erreichbare/nutzlose Symbole in einer Grammatik

  Theorem: Für jede cf-Grammatik G mit $L(G) \neq \emptyset$ existiert eine cf-Grammatik G' ohne nutzlose Symbole, mit $L(G) = L(G')$.

- Umformung von Grammatiken
  - Startsymbol nur links
  - Eliminierung nutzloser Symbole
  - Normalformen für die Regel
    (für jede Regel $P \to Q$ gilt: entweder $Q \in V^*$ oder $Q \in T$)
  - Elimination von ε-Regeln
  - Elimination von Kettenproduktionen
- Chomsky-Normalform (Definition, Konstruktion einer äquivalenter cf-Grammatik in Chomsky-Normalform)
- Greibach-Normalform (nur Definition)
- $\mathcal{L}_2 \subseteq \mathcal{L}_1$ (Jede kontextfreie Sprache ist auch kontextsensitiv)
- Pumping-Lemma für kontextfreie Sprachen
- Pushdown-Automaten (PDAs)
  - Definition; Konfiguration,
    Startkonfiguration, Nachfolgekonfiguration; Rechnung
  - Von PDA akzeptierte Sprache
    ($L_f(M)$: über finale Zustände/ $L_l(M)$: über leeren Keller)
  - Zu jedem PDA M_1 existiert ein PDA M_2 mit $L_f(M_1) = L_l(M_2)$.
  - Zu jedem PDA M_1 existiert ein PDA M_2 mit $L_l(M_1) = L_f(M_2)$.
- Die Klasse der PDA-akzeptierten Sprachen ist $\mathcal{L}_2$
- Abschlusseigenschaften von $\mathcal{L}_2$
  - Abgeschlossen gegen $\cup, \circ, *$
  - Nicht abgeschlossen gegen $\cap, \neg$
- Wortprobleme
  CYK-Algorithmus (Cocke-Younger-Kasami)

Turingmaschinen/rekursiv aufzählbare Sprachen

- **Determinierte Turing-Maschinen (DTMs)**
  - Definition; Konfiguration; Rechnung; Halten/Hängen.
  - TM-berechenbare Funktion (TM^{part}, TM).
  - Von einer DTM akzeptierte Sprache.
- TM-Flussdiagramme.
- Varianten von Turing-Maschinen:
  - Standard-DTM
  - Turing-Maschinen, die nie hängen.
    Theorem: Zu jeder DTM M existiert eine DTM M', die nie hängt und L(M) akzeptiert.
  - DTM mit zweiseitig unbeschränktem Band.
    Theorem: Zu jeder zw-DTM M, die eine Funktion f berechnet oder eine Sprache L akzeptiert, existiert eine Standard-DTM M, die ebenfalls f berechnet bzw. L akzeptiert.

- DTM mit k Halbbändern (k-DTM).
  Theorem: Zu jeder k-DTM M, die eine Funktion f berechnet oder eine Sprache L akzeptiert, existiert eine Standard-DTM M, die ebenfalls f berechnet bzw. L akzeptiert.
- Indeterminierte Turing Maschinen (NDTMs) ;
  Von einem NDTM akzeptierte Sprache
  Theorem: Jede Sprache, die von einer indeterminierten Turing-Maschine akzeptiert wird, wird auch von einer Standard-DTM akzeptiert.

- Universelle determinierte Turing-Maschinen (Idee; Kodierung einer DTM als Wort (Idee); Gödelisierung (Idee))
- Aufzählbarkeit, Akzeptierbarkeit, Entscheidbarkeit
  - Akzeptierbar = Rekursiv Aufzählbar.
  - Entscheidbar $\Rightarrow$ Akzeptierbar.
  - Das Komplement einer entscheidbaren Sprache ist entscheidbar
  - Eine Sprache L ist genau dann entscheidbar, wenn sie und ihr Komplement akzeptierbar sind.
  - Rekursiv Aufzählbar = Typ 0.
- Unentscheidbarkeit, Nicht-berechenbare Funktionen
  - Beispiel: The Busy Beaver Funktion - nicht berechenbar.
  - Gödelnummern von DTMs (Idee; Jede Zahl ist Gödelnummer)
  - Halteprobleme (Allgemeines Halteproblem, Spezielles Halteproblem, Null-Halteproblem)
  - Leerheitsproblem, Totalitätsproblem, Gleichheitsproblem, Entscheidbarkeitsproblem.
  - Unentscheidbarkeit des speziellen Halteproblems.
    Akzeptierbarkeit des speziellen Halteproblems.
  - Reduktion von Problemen (Definition)
    Theorem: Ist L_1 auf L_2 reduzierbar ($L_1 \preceq L_2$), und ist L_1 unentscheidbar, so ist auch L_2 unentscheidbar.
  - Das Null-Halteproblem ist unentscheidbar.

Komplexitätstheorie

- DTIME und NTIME; DSPACE und NSPACE
- Falls f berechenbar, so ist jede Sprache in DTIME($f(n)$) entscheidbar.
- Falls f berechenbar, so ist jede Sprache in DSPACE($f(n)$) entscheidbar.
- DTIME($f(n)$) $\subseteq$ NTIME($f(n)$); DSPACE($f(n)$) $\subseteq$ NSPACE($f(n)$);
- NTIME($f(n)$) $\subseteq$ DTIME($2^{h(n)}$) where $h \in O(f)$; NSPACE($f(n)$) $\subseteq$ DSPACE($f(n)^2$)

- Konstante Faktoren werden ignoriert.
- **P, NP, PSPACE (Definitionen)**
- **P** $\subseteq$ **NP** $\subseteq$ **PSPACE**
- Polynomial-Zeit-reduzibilität (Definition)

– Wenn $L_1 \preceq_{\mathsf{pol}} L_2$ und $L_2 \in \mathbf{P}$ (bzw. $\mathbf{NP}$) dann $L_1 \in \mathbf{P}$ (bzw. $\mathbf{NP}$).
– Die Komposition zweier Polynomial-Zeit-Reduktionen ist wieder eine Polynomial-Zeit-Reduktion.

- **NP-hart, NP-vollständig (Definition)**
  Beispiel: SAT Problem (NP-vollständig)
- PSPACE-hart, PSPACE-vollständig (Definition)
- Abgschlossenheit der Komplexitätsklassen:
  P, PSPACE sind abgeschlossen unter Komplement
  NP: nicht bekannt, ob unter Komplement abgeschlossen (co-NP, Definition)
- Beziehungen, die momentan noch unbekannt sind:
  1. P =? NP.
  2. NP =? co-NP.
  3. P =? PSPACE.
  4. NP =? PSPACE.