Lernzettel

Kellerautomaten (Pushdown-Automaten) und kontextfreie Sprachen

Universität: Technische Universität Berlin
Kurs/Modul: Theoretische Grundlagen der Informatik
Erstellungsdatum: September 19, 2025

Lernzettel: Kellerautomaten (Pushdown-Automaten) und kontextfreie Sprachen

(1) Grundbegriffe.
Ein Pushdown-Automat (PDA) ist ein endliches Steuerwerk mit einem zusätzlichen Speicher in Form eines Stacks (Stackspeicher). Formale Definition:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

- Q: endliche Menge von Zuständen,
- Σ: Eingabealphabet,
- Γ: Stackalphabet,
- $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to \mathcal{P}(Q \times \Gamma^*)$ (Übergangsrelation),
- $q_0 \in Q$: Startzustand,
- $Z_0 \in \Gamma$: Startstacksymbol,
- $F \subseteq Q$: Endzustandsmenge.

(2) Übergänge und Akzeptanz.
Ein Schritt des PDA wird durch

$$(q, a, X) \to (q', \alpha) \quad \text{mit} \quad (q', \alpha) \in \delta(q, a, X)$$

beschrieben, wobei $a \in \Sigma \cup \{\epsilon\}$, $X \in \Gamma$ und $\alpha \in \Gamma^*$.
Akzeptanzarten:

$$L(M)_F = \{w \in \Sigma^* \mid (q_0, Z_0)w(q, \gamma) \text{ mit } q \in F\}$$
$$L(M)_{\text{empty}} = \{w \in \Sigma^* \mid (q_0, Z_0)w(q, \epsilon)\}$$

(3) Struktur eines kontextfreien Sprachsystems (CFG).
Eine kontextfreie Grammatik G wird beschrieben durch

$$G = (V, \Sigma, R, S)$$

mit

- V: endliche Menge von Variablen (Nichtterminalen),
- Σ: endliches Alphabet der Terme (Terminalsymbole),
- R: endliche Menge von Produktionsregeln ($A \to \alpha$ mit $A \in V$, $\alpha \in (V \cup \Sigma)^*$),
- $S \in V$: Startsymbol.

Die von S erzeugte Sprache ist

$$L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}.$$

(4) Zusammenhang PDA kontextfreie Sprachen.
Es gilt: Eine Sprache ist kontextfrei genau dann, wenn sie von einem PDA erkannt wird. Formal:

$$L \text{ ist kontextfrei} \iff \exists \text{ PDA } M \text{ mit } L = L(M).$$

Zudem gilt die Äquivalenz zwischen CFG und PDA (existenzielle Übersetzungen):

$$L(G) = L(M) \text{ für geeignete } G \text{ und } M.$$

(5) Beispielgrammatik: Dyck-Sprachen (ausgeglichene Klammern).
Die einfachste Dyck-Sprache über einer einzigen Klammerart (und) wird durch die CFG

$$S \to \varepsilon \mid S\,S \mid (\,S\,)$$

gegeben.
Es gilt

$$L = \{w \in \{(,)\}^* \mid \text{w ist korrekt gegliedert mit (und)}\}.$$

(6) Ein einfaches Beispiel für einen PDA (informell).
Zur Erkennung von Sprache $\{a^n b^n \mid n \geq 0\}$ benötigt man einen Stack, um die Anzahl der a zu speichern und dann mit b abzubauen.
Konzeptionell:

- Bei Eingabe von a wird ein Symbol auf den Stack gelegt.
- Bei Eingabe von b wird ein Symbol vom Stack entfernt.
- Akzeptiert wird nach Ablesen der Eingabe, wenn der Stack leer ist (und ggf der Endzustand erreicht wurde).

Daraus folgt, dass $a^n b^n$ kontextfrei ist.

(7) Typische Eigenschaften und Folgerungen.

- PDAs erkennen genau die kontextfreien Sprachen.
- Es gibt eine konstruktive Übersetzung von CFG zu PDA und umgekehrt.
- Akzeptanz durch Endzustand und Akzeptanz durch leeren Stack sind äquivalent, sofern man die nötigen epsilon-Übergänge zulässt bzw. den Stack entsprechend initialisiert.

(8) Notation und kurze Hilfsmittel.
Für Übergänge und Ableitungen arbeiten wir oft mit der Notation

$$(p, a, X) \Rightarrow (q, \gamma) \quad \text{bzw.} \quad (q, \gamma) \in \delta(p, a, X).$$

$$\Rightarrow^* \quad \text{bedeutet eine Folge von Schritten.}$$

(9) Übungsideen.
- Forme eine CFG für die Sprache $\{a^i b^j \mid i = j \text{ oder } i = 0\}$.
- Skizziere den PDA für $\{a^n b^n \mid n \geq 0\}$ und beschreibe die Stack-Operationen.
- Zeige, dass die Grammatik $S \to SS \mid a$ eine kontextfreie Sprache erzeugt und bestimme deren Sprache.