Automaten

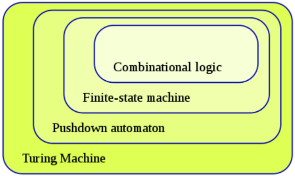
Andreas M. Chwatal

Programmieren und Software-Engineering
Homomorphismen, Formale Sprachen und Syntax-Analyse

30. Juni 2021

Automaten-Theorie

- Endliche Automaten können die syntaktische Korrektheit von Wörtern bezüglich regulärer Sprachen überprüfen.
- Das Ergebnis bezüglich eines Wortes lautet: akzeptiert oder nicht akzeptiert $\Rightarrow$ man nennt solche Automaten daher auch *Akzeptoren*.
- Der *Keller-Automat* (engl. Pushdown-Automaton) nutzt einen zusätzlichen Speicherbereich um auch Wörter kontextfreier Sprachen überprüfen zu können.
- Eine weitere Verallgemeinerung zur *Turing-Maschine* ermöglicht tatsächliche Berechnungen mittels des Automaten-Konzeptes vorzunehmen.



Keller-Automat

- Der Keller-Automat erweitert das Konzept des endlichen Automaten um einen Speicher
- Speicher: "Keller-Speicher" (=Stack)
- Englische Bezeichnung: pushdown automaton
- Bestandteile:
  - Zustandsmenge N
  - Eingabealphabet T
  - Kelleralphabet K
  - Übergangsfunktion $\delta : K \times (T \cup \{\varepsilon\}) \times K \to 2^{S \times K^*}$
  - Startzustand S
- Der Kellerspeicher wird mit $\perp$ initialisiert.
- In jedem Schritt wird ein Zeichen σ des zu verarbeitenden Wortes eingelesen.
- Ebenso wird das obeste Zeichen γ aus dem Kellerspeicher gelesen.
- Zusätzlich zum Zustandsübergang (δ) wird nun eine Zeichenkette aus K^* in den Kellerspeicher geschrieben.

Wir betrachten die reguläre Sprache

$$L = \{a^n b^n \mid n \in \mathbb{N}^+\}.$$

Der Kellerautomat $A = (N, T, K, \delta, S)$ ist nun wie folgt definiert:

$$\begin{aligned} N &= \{A, B\} \\ T &= \{a, b\} \\ K &= \{\perp, X\} \end{aligned}$$

Die Übergangsfunktion lautet:

$$\begin{aligned} \delta(A, a, \perp) &\rightarrow \{(A, X\perp)\} \\ \delta(A, a, X) &\rightarrow \{(A, XX)\} \\ \delta(A, b, X) &\rightarrow \{(B, \varepsilon)\} \\ \delta(B, b, X) &\rightarrow \{(B, \varepsilon)\} \\ \delta(B, \varepsilon, \perp) &\rightarrow \{(B, \varepsilon)\} \end{aligned}$$

Beispiel: Wort *aabb* wird akzeptiert:

Zustand	(Rest-)Wort	Keller
A	aabb	$\perp$
A	abb	$X\perp$
A	bb	$XX\perp$
B	b	$X\perp$
B	ε	$\perp$
B	ε	ε

Beispiel: Wort *aaabb* wird *nicht* akzeptiert:

Zustand	(Rest-)Wort	Keller
A	aaabb	$\perp$
A	aabb	$X\perp$
A	abb	$XX\perp$
A	bb	$XXX\perp$
B	b	$XX\perp$
B	ε	$X\perp$

Die Klasse der von (nichtdeterministischen) Kellerautomaten akzeptierten Sprachen ist mit der Klasse der kontextfreien Sprachen identisch.

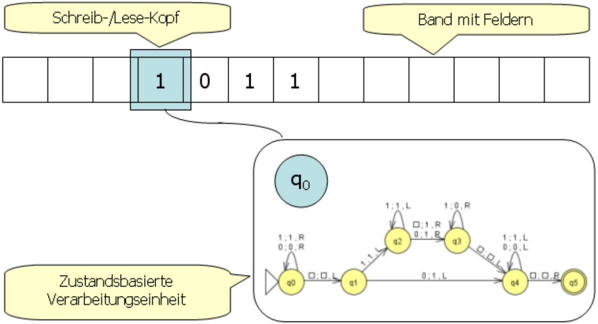
Die Sprachklasse der von deterministischen Kellerautomaten akzeptierten Sprachen ist eine echte Teilmenge der von nichtdeterministischen Kellerautomaten akzeptierten Sprachen.

Anmerkung: Grundsätzlich ist es möglich die

Turing-Maschine

- Die Turing-Maschine ist eine weitere Verallgemeinerung des Automatenkonzeptes
- Sie besteht aus einem Zustandsautomaten, einem Schreib- und Lesekopf und einem (unendlich langen) Band
- In jedem Schritt wird
  - ein Zeichen vom Band gelesen,
  - ein Zustandsübergang ausgeführt (abhängig vom gelesenen Zeichen und dem aktuellen Zustand),
  - ein Zeichen auf das Band geschrieben, und
  - der Schreib-/Lese-Kopf nach rechts oder links bewegt.

Turing-Maschine



Quelle: `https://www.inf-schule.de/`

Turing-Maschine

- Die Turing-Maschine ist universelles Modell für Computer
- Alle Berechnungen die mit einem Computer durchgeführt werden können, können (theoretisch) auch mit einer Turing-Maschine durchgeführt werden.
- Die Turing-Maschine ermöglicht somit eine theoretische Analyse von Berechenbarkeit und Komplexität.
- Bezüglich Typ-0 Grammatiken können Wörter mit der Turing-Maschine auf ihre syntaktische Korrektheit geprüft werden.

Literaturübersicht I

- Dirk W. Hoffmann: “Theoretische Informatik”, Hanser, 3. Auflage
- Gernot Salzer: “Einführung in die Theorie der Informatik”, Skriptum, TU Wien, 2001
- Wikipedia (Englisch): `https://en.wikipedia.org/`