Berechenbarkeit und Komplexität
Die Chomsky-Hierarchie

Wolfgang Schreiner
Wolfgang.Schreiner@risc.jku.at

Research Institute for Symbolic Computation (RISC)
Johannes Kepler University, Linz, Austria
http://www.risc.jku.at

Sprachen und Automaten

- Reguläre Sprachen
  - Durch reguläre Ausdrücke beschreibbar.
  - Durch endliche Automaten erkennbar (und entscheidbar).
- Rekursiv aufzählbare Sprachenn
  - Durch . . . beschreibbar?
  - Durch Turing-Maschinen erkennbar (nicht unbedingt entscheidbar).
- Zusammenhang
  - Jede reguläre Sprache ist rekursiv aufzählbar (sogar rekursiv).
  - Jeder endliche Automat ist durch eine Turing-Maschine simulierbar.

Gibt es noch andere interessante Sprachklassen und dazugehörige Automatenmodelle und wie hängen diese mit den obigen zusammen?

Grammatiken

Grammatik $G = (N, \Sigma, P, S)$

- N ... die Menge der Nonterminalsymbole.
- Σ ... die Menge der Terminalsymbole
  - $N \cap \Sigma = \emptyset$
- P ... die Menge der Ersetzungsregeln der Form $\alpha \to \beta$
  - $\alpha \in (N \cup \Sigma)^* \circ N \circ (N \cup \Sigma)^*$
  - $\beta \in (N \cup \Sigma)^*$

  Die linke Seite α enthält mindestens ein Nonterminalsymbol; die rechte Seite β enthält beliebige Symbole.
- $S \in N$ ... das Satzsymbol.

G beschreibt eine Sprache über dem Alphabet Σ.

Satzformen und Sätze

Gegeben ist eine Grammatik $G = (N, \Sigma, P, S)$.

- Eine direkte Ableitung $\omega \Rightarrow \omega'$ in G:
  - $\omega = \omega_1 \ \alpha \ \omega_2$, $\omega' = \omega_1 \ \beta \ \omega_2$
  - $(\alpha \rightarrow \beta) \in P$
    - Für beliebige $\omega_1, \omega_2 \in (N \cup \Sigma)^*$.
- Eine Ableitung $\omega \Rightarrow^* \omega'$
  - Eine Verkettung direkter Ableitungen $\omega \Rightarrow \ldots \Rightarrow \omega'$.
- α ist eine Satzform in G:
  - $S \Rightarrow^* \alpha$
    - α kann aus dem Satzsymbol abgeleitet werden.
- α ist ein Satz in G:
  - α ist eine Satzform in G, die nur aus Terminalsymbolen besteht.
- Die Sprache $L(G)$ von G:
  - $L(G) := \{\alpha \mid \alpha \text{ ist ein Satz von } G\}$

Beispiel

- Grammatik $G = (N, \Sigma, P, S)$.
  - $N = \{S, A, B\}$
  - $\Sigma = \{a, b, c\}$
  - $P = \{S \rightarrow Ac,\ A \rightarrow aB, A \rightarrow BBb, B \rightarrow b, B \rightarrow ab\}$
    - $P = \{S \rightarrow Ac, A \rightarrow aB|BBb, B \rightarrow b|ab\}$
- Herleitung von Sätzen von G:
  - $S \Rightarrow Ac \Rightarrow aBc \Rightarrow abc$
  - $S \Rightarrow Ac \Rightarrow BBbc \Rightarrow abBbc \Rightarrow ababbc$
  - ...
- Die Sprache $L(G)$:
  - $L(G) = \{abc, aabc, bbbc, babbc, abbbc, ababbc\}$

Enthält eine Grammatik G eine "rekursive" Regel der Form $\omega_1 A \omega_2 \Rightarrow \omega_3 A \omega_4$, dann kann die Sprache $L(G)$ auch unendlich sein.

Reguläre und Rekursive Sprachen

- Satz: Jede rekursiv aufzählbare Sprache ist durch eine Grammatik beschreibbar.
  - Grammatiken sind gleich ausdrucksstark wie Turing-Maschinen.
- Satz: Jede reguläre Sprache ist durch eine rechtslineare Grammatik beschreibbar.
  - Jede Regel einer rechtslinearen Grammatik hat eine der beiden Formen $A \rightarrow \alpha$ oder $A \rightarrow \alpha B$.
    - $A, B \in N$.
    - $\alpha \in \Sigma^*$.

Man kann Sprachen anhand von Einschränkungen der zu ihrer Beschreibung erforderlichen Grammatiken kategorisieren.

Die Chomsky-Hierarchie

Noam Chomsky, 1959.

Typ	Grammatik	Erzeugte Sprache	Erkennender Automat
0	unbeschränkt	rekursiv aufzählbar	Turing-Maschine
1	kontextsensitiv	kontextsensitiv	linear beschränkter Automat
2	kontextfrei	kontextfrei	Kellerautomat
3	rechtslinear	regulär	endlicher Automat

- Die Menge der Sprachen vom Typ $i+1$ ist eine echte Teilmenge der Menge der Sprachen vom Typ i.
- Jeder Automat vom Typ $i+1$ kann durch einen Automaten vom Typ i simuliert werden (aber nicht umbekehrt).

Die Chomsky-Hierarchie (eingebettet)

Typ	Grammatik	Erzeugte Sprache	Erkennender Automat
-	-	*überabzählbar unendlich*	-
-	-	*abzählbar unendlich*	-
0	unbeschränkt	rekursiv aufzählbar	Turing-Maschine (TM)
-	-	*rekursiv*	*stets haltende TM*
1	kontextsensitiv	kontextsensitiv	linear beschränkter Automat
2	kontextfrei	kontextfrei	Kellerautomat
3	rechtslinear	regulär	endlicher Automat
-	-	*endlich*	*(endliche Tabelle)*

endlich $\subset$ regulär $\subset$ kontextfrei $\subset$ kontextsensitiv $\subset$ rekursiv $\subset$ rekursiv aufzählbar $\subset$ abzählbar unendlich $\subset$ überabzählbar unendlich

Beispiele

- Endlich:
  - $\{a, aa, aaa\}$
- Typ 3 (regulär, nicht endlich):
  - $\{a^i \mid i \in \mathbb{N}\}$
- Typ 2 (kontextfrei, nicht regulär):
  - $\{a^i b^i \mid i \in \mathbb{N}\}$
- Typ 1 (kontextsensitiv, nicht kontextfrei):
  - $\{a^i b^i c^i \mid i \in \mathbb{N}\}$
- Rekursiv (nicht kontextsensitiv):
  - $\{\alpha_i \mid \alpha_i$ ist nicht Satz der kontextsensitiven Grammatik i$\}$
- Typ 0 (rekursiv aufzählbar, nicht rekursiv):
  - $\{\alpha_i \mid \alpha_i$ wird von Turingmaschine i erkannt$\}$
- Abzählbar unendlich (nicht rekursiv aufzählbar):
  - $\{\alpha_i \mid \alpha_i$ wird nicht von Turingmaschine i erkannt$\}$
- Überabzählbar unendlich (nicht abzählbar unendlich):
  - Die Menge aller formalen Sprachen.

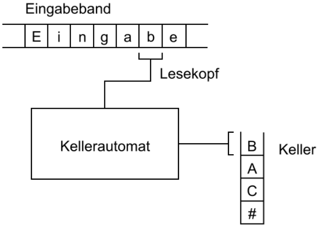
Kontextsensitive Sprachen (Typ 1)

- Kontextsensitive Grammatik:
  - In jeder Regel $\alpha \rightarrow \beta$ gilt $|\alpha| \leq |\beta|$.
  - Die Regel $S \rightarrow \epsilon$ ist nur zugelassen, wenn S in keiner Regel auf der rechten Seite vorkommt.
- Beispiel: $L = \{a^i b^i c^i \mid i \in \mathbb{N}\}$
  - $S \rightarrow aSBC|aBC, CB \rightarrow HB, HB \rightarrow HC, HC \rightarrow BC, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc$
  - $S \Rightarrow a\underline{S}BC \Rightarrow aaB\underline{CB}C \Rightarrow aaB\underline{HB}C \Rightarrow aaB\underline{HC}C \Rightarrow a\underline{aB}BCC \Rightarrow aa\underline{bB}CC \Rightarrow aab\underline{bC}C \Rightarrow aabb\underline{cC} \Rightarrow aabbcc$
- Linear beschränkter Automat: eine nichtdeterministische Turingmaschine mit k-Bändern, die sich auf jedem Band nur auf dem Bereich der Länge n des Eingabeworts bewegt.
  - Der Automat hat $n * k$ Bandzellen (also nur ein linear Vielfaches der Länge des Eingabeworts) als "Arbeitsspeicher" zur Verfügung.

Kontextsensitive Sprachen heute haben keine praktische Bedeutung.

Kontextfreie Sprachen (Typ 2)

- Kontextfreie Grammatik:
  - Jede Regel hat die Form $A \rightarrow \beta$ mit $A \in N$.
    - Das Nonterminal-Symbol A kann unabhängig von dem Kontext, in dem es vorkommt, ersetzt werden.
- Beispiel: $\{a^i b^i \mid i \in \mathbb{N}\}$
  - $S \rightarrow \epsilon | aSb$
  - $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$.
- Kellerautomat: ein nichtdeterministischer endlicher Automat mit einem Keller (stack) als Arbeitsspeicher:
  - In jedem Zug ist die Aktion auch vom obersten Symbol des Kellers abhängig.
  - Das oberstes Symbol wird durch eine (auch leere) Symbolfolge ersetzt.
  - Das Eingabesymbol muss nicht unbedingt gelesen werden.



Praktisch alle in der Informatik verwendeten Sprachen sind kontextfrei.

Erzeugung von Syntaxanalysatoren

"Compilergeneratoren" zur Erzeugung von Syntaxanalysatoren.

- Eingabe: eine (deterministische) kontextfreie Grammatik.

```
statement: assignment | conditional | whileloop | ... ;
whileloop: 'while' '(' valexp ')' statement ;
```

- Ausgabe: ein (deterministischer) Kellerautomat (als Programm).

```
public final LoopStatement whileloop() throws ... {
  ...
  pushFollow(FOLLOW_valexp_in_whileloop1457);
  valexp();
  state._fsp--;
  if (state.failed) return value;
  ...
  pushFollow(FOLLOW_statement_in_whileloop1484);
  statement();
  state._fsp--;
  if (state.failed) return value;
  ...
}
```