Proseminar Theoretische Informatik, WS 2010/11

Chomsky-Hierarchie

von Wojciech Lukasiewicz

Grammatiken

Eine Grammatik G nennt man einen 4-Tupel

$G = (V, T, P, S)$,

wobei

V – eine Menge der Nichtterminalsymbole

T – eine Menge der Terminalsymbole

$P \subseteq (V \cup T)^+ \times (V \cup T)^*$ – eine Menge der Produktionsregeln

$S \in V$ – ein Startsymbol.

Eine Produktion ist ein Paar (l, r) (linke/rechte Seite der Produktion),

mit

$l \in (V \cup T)^+, \; r \in (V \cup T)^*$ (linke Seite darf nicht leer sein).

Notation:

$l \to r$, wenn sich r in einem Schritt (durch Anwendung einer Produktion) aus l ableiten lässt.

Ableitbarkeit

Ein Wort β ist *ableitbar* aus dem Wort α in einer Grammatik G , was wir folgendermaßen notieren:

$\alpha \Rightarrow_G^* \beta$,

falls es solche $\phi_0, \phi_1, \ldots, \phi_n \in (V \cup T)^*$ gibt, dass

$\phi_0 = \alpha$

$\phi_n = \beta$

$\phi_{i-1} \Rightarrow_G \phi_i, \quad \text{für } i = 1, 2, \ldots, n$.

Die Sequenz $\phi_0, \phi_1, \ldots, \phi_n$ bezeichnen wir als eine *Ableitung* der Länge n .

Eine von einer Grammatik erzeugte Sprache

Eine Grammatik ist eine Art, auf die man eine formale Sprache definieren kann. Haben wir eine gegebene Grammatik G , bezeichnen wir als $L(G)$ die Menge aller Wörter, die in dieser Grammatik vom Startsymbol erzeugt werden können. Diese Menge nennt man eine Sprache, die von gegebener Grammatik G generiert wird.

$$L(G)=\{x\in T^* | S \overset{*}{\Rightarrow}_G x\}$$

Chomsky-Hierarchie

Noam Chomsky, amerikanischer Sprachforscher hat Mitte 50er Jahren vier Klassen formaler Grammatiken und vier Klassen formaler Sprachen definiert. Die Stufen dieser 4-Klassen-Hierarchie hat er mit Nummern 0-3 bezeichnet.

Typ-0-Grammatiken

Eine Grammatik $G = (V, T, P, S)$, in der jede Produktion die Form $\alpha \rightarrow \beta$, mit $\alpha, \beta \in (V \cup T)^*$ beliebigen Strings und $\alpha \neq \epsilon$ nennen wir eine unbeschränkte Grammatik oder Phrasenstrukturgrammatik.

Wie wir sehen, werden die Produktionsregeln im Vergleich zur allgemeinen Definition einer Grammatik gar nicht beschränkt.

Die von den Typ-0-Grammatiken generierten Sprachen heißen *rekursiv aufzählbare* Sprachen.

Das Automatenmodell, das die rekursiv aufzählbaren Sprachen erkennt, ist die Turingmaschine.

Beweis:

z.z: Die Klasse der von Typ-0-Grammatiken erzeugten Sprachen und die Klasse der rekursiv aufzählbaren Sprachen stimmen überein.

Idee: Turingmaschinen, die rückwärts arbeiten sind Grammatiken, und Grammatiken, die rückwärts arbeiten, sind Turingmaschinen.

Typ-1-Grammatiken

Eine Grammatik $G = (V, T, P, S)$, in der jede Produktion die Form $\alpha \rightarrow \beta$, mit $\alpha, \beta \in (V \cup T)^*$ beliebigen Strings, β ist mindestens so lang, wie α ($|\alpha| \leq |\beta|$) und zusätzlich ist die Produktion $S \rightarrow \epsilon$ erlaubt (falls die Sprache das leere Wort beinhaltet),

nennen wir eine kontextsensitive Grammatik oder monotone Grammatik.
„Kontextsensitiv", weil für jede Grammatik obiger Art, kann man eine äquivalente (also dieselbe Sprache erzeugende) Grammatik finden, deren Produktionen (mit eventueller Ausnahme $S \to \epsilon$) die Form $\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2$, mit $A \in V, \alpha_1, \alpha_2, \beta \in (V \cup T)^*$ und $\beta \neq \epsilon$.
Solche Produktionen erlauben die Ersetzung des Nichtterminals A mit der Zeichenkette β nur dann, wenn „Linkskontext" α_1 und „Rechtskontext" α_2 , in denen A steht, vorhanden sind.
Die von solchen Grammatiken erzeugten Sprachen heißen *kontextsensitiv*.
Das Automatenmodell, das die kontextsensitiven Sprachen erkennt, ist eine linar platzbeschränkte Turingmaschine.

Typ-2-Grammatiken

Eine Grammatik $G = (V, T, P, S)$, in der jede Produktion die Form $A \to \beta$, mit $A \in V$ und $\beta \in (V \cup T)^*$ einem beliebigen String, nennen wir eine kontextfreie Grammatik. Die Bezeichnung „kontextfrei" kommt davon her, dass die Produktionen die bedingungslose (ohne den Rücksicht auf den Kontext zu nehmen) Ersetzung des Nichtterminals A mit der Zeichenkette β erlauben.
Die von solchen Grammatiken erzeugten Sprachen heißen *kontextfrei*.
Die Klasse der kontextfreien Grammatiken mag die wichtigste (aus unserer Sicht) Grammatikenklasse sein, denn mit Hilfe den Typ-2-Grammatiken beschreibt man die Syntax von meisten Programmiersprachen.
Das Automatenmodell, das die kontextfreien Sprachen erkennt, ist ein Kellerautomat.

Typ-3-Grammatiken

Eine Grammatik $G = (V, T, P, S)$, in der jede Produktion die Form $A \to xB$, oder $A \to x$, mit $A, B \in V, x \in T^+$, nennen wir eine rechtslineare Grammatik.
Eine Grammatik $G = (V, T, P, S)$, in der jede Produktion die Form $A \to Bx$, oder $A \to x$, mit $A, B \in N, x \in T^+$, nennen wir, dementsprechend, linkslinear.
Die rechts- und linkslineare Grammatiken nennen wir lineare Grammatiken, reguläre Grammatiken oder Typ-3-Grammatiken.
Die von ihnen erzeugten Sprachen werden *regulär* genannt.
Das Automatenmodell, das die regulären Sprachen erkennt, ist ein endlicher Automat.

Anmerkungen

Eine Sprache kann gleichzeitig von Grammatiken unterschiedlicher Typen erzeugt werden. Zur Bezeichnung dieser Grammatik benutzt man daher den „höchsten", am meisten eingeschränkten Typ. Formal:

Eine Sprache, die von einer Grammatik des Typs k erzeugt wird, aber von keiner Grammatik eines geringeren Typs $l > k$ erzeugt werden kann, heißt eine Sprache des Typs k .

Weil (bis auf das Verhältnis zwischen kontextfreien und kontextsensitiven Grammatiken [Beweisidee: eine kfG in eine ksG umwandeln]) eine Grammatik des niedrigeren Typs lässt alle Regeln der höheren Grammatiken zu, schließen sich die aufeinanderfolgenden Klassen ineinander ein.

$$L_3 \subset L_2 \subset L_1 \subset L_0$$

$$REG \subset CF \subset CS \subset \mathrm{RE}$$

In der obigen Relation sind die Sprachklassen echte Untermengen.

Das Wortproblem ist für die Grammatiken der Typen 1-3 entscheidbar, für Typ-0 nicht.