Grundlagen der Programmierung (Vorlesung 20)

Ralf Möller, FH-Wedel

- Vorige Vorlesung
  - (Asymptotische) Komplexität von Algorithmen
- Inhalt dieser Vorlesung
  - Automatentheorie und Formale Sprachen
- Lernziele
  - Grundkenntnisse in der Beschreibung von Programmiersprachen (sowie Teilen davon)
  - Kennenlernen von abstrakten Maschinen zur Charakterisierung von Problemen

Danksagung

- Die Präsentationen sind an den Inhalt des Buches "Theoretische Informatik kurzgefaßt" von Uwe Schöning angeleht und wurden aus den Unterlagen zu der Vorlesung "Informatik IV - Theoretische Informatik" an der TU München von Angelika Steger übernommen
- Die Originalunterlagen befinden sich unter: http://www14.in.tum.de/lehre/2000SS/info4/

Festlegung von Sprachen: Ein erstes Beispiel

<Satz> → <Subjekt> <Prädikat> <Objekt>

<Subjekt> → <Artikel> <Attribut> <Substantiv>

<Artikel> → ε

<Artikel> → der

<Artikel> → die

<Artikel> → das

<Attribut> → ε

<Attribut> → <Adjektiv>

<Attribut> → <Adjektiv> <Attribut>

<Adjektiv> → klein

<Adjektiv> → groß

u.s.w.

Vorbemerkung: Alphabet, Zeichen, Worte

- Im Zusammenhang mit Formalen Sprachen heißt eine endliche nicht-leere Menge <u>Alphabet</u>
  - Beispiel: $\Sigma = \{a, b\}$
- Die Elemente eines Alphabets heißen <u>Zeichen</u> oder <u>Symbole</u>
  - Also: in unserem Beispiel sind a und b Zeichen
- Zeichen können "hintereinandergefügt" werden und bilden durch die Reihung <u>Worte</u>
- Der Operator hierzu heißt <u>Konkatenation</u>
  - Auch Worte können zu neuen Worten konkateniert werden
- Eine Reihung ohne Zeichen heißt <u>leeres Wort</u> (Notation: ε)

Vorbemerkung: Monoid

- Die Menge aller Worte, die sich durch Konkatenation aus einer Menge Σ bilden lassen, wird mit $\underline{\Sigma^*}$ bezeichnet:
  - $\Sigma^* = \{ \varepsilon, a, b, aa, ab, ba, bb, aaa, aab, \ldots \}$
- Mit $\underline{\Sigma^+}$ bezeichnen wir $\Sigma^* - \{\varepsilon\}$
- Sei "o" die Bezeichnung für die Konkatenation.
- In der algebraischen Struktur (Σ^*, o) sind die folgenden Axiome erfüllt (die Struktur wird <u>Monoid</u> genannt):
  - $x \in \Sigma^* \wedge y \in \Sigma^* \rightarrow x \circ y = xy \in \Sigma^*$ (Abgeschlossenheit)
  - $(x \circ y) \circ z = x \circ (y \circ z) = xyz$ (Assoziativität)
  - $\varepsilon \circ x = x \circ \varepsilon = x$ (neutrales Element)

Vorbemerkungen: Länge, Mächtigkeit, Sprache

- Für $n \in N_0$ ist w^n definiert als $w \circ (w \circ (w \circ ..)..)$, $w^0 = \varepsilon$
- Für ein Wort w bezeichnet $|w|$ die <u>Länge</u> (also die Anzahl der Zeichen des Wortes) und für eine Menge M bezeichnet $|M|$ die <u>Mächtigkeit</u> (also die Anzahl der Elemente)
- Eine <u>Formale Sprache</u> (über einem Alphabet Σ) ist eine Teilmenge von Σ^*

Grammatik

Definition. Eine *Grammatik* ist ein 4-Tupel $G = (V, \Sigma, P, S)$ das folgende Bedingungen erfüllt:

- V ist eine endliche Menge, die Menge der *Variablen*.
- Σ ist eine endliche Menge, das *Terminalalphabet*, wobei $V \cap \Sigma = \emptyset$.
- P ist die Menge der *Produktionen* oder *Regeln*. P ist eine endliche Teilmenge von $(V \cup \Sigma)^+ \times (V \cup \Sigma)^*$. (Schreibweise: $(u, v) \in P$ schreibt man meist als $u \to v$.)
- $S \in V$ ist die *Startvariable*.

Übergang

Seien $u, v \in (V \cup \Sigma)^*$. Wir definieren die Relation $u \Rightarrow_G v$ (in Worten: u geht unter G unmittelbar über in v), falls u und v die Form haben

$$\begin{array}{lcl} u & = & xyz \\ v & = & xy'z \quad \text{mit } x, z \in (V \cup \Sigma)^* \text{ und} \\ y & \to & y' \quad \text{eine Regel in } P \text{ ist.} \end{array}$$

(Bem: Falls klar ist, welche Grammatik gemeint ist, so schreiben wir oft auch einfach kurz $u \Rightarrow v$ anstelle von $u \Rightarrow_G v$.)

Transitive Hülle einer Relation: Motivation

- Ein Wort wird durch Produktionsregeln in ein neues Wort abgebildet
- Was in was abgebildet wird, ist durch die Übergangsrelation $\Rightarrow$ gegeben
- Im Kontext einer Grammatik können Produktionsregeln mehrfach hintereinander angewendet werden
- Es soll nun alles, was auch mehrschrittig abgeleitet werden kann, betrachtet werden
- Es ist in diesem Fall nicht die direkte Übergangsrelation zu betrachten, sondern die sogenannte <u>Hülle</u> der Übergangsrelation

Transitive Hülle einer Relation: Definition

- Seien R und S zwei zweistellige Relationen über einer Menge M, so daß $R \subseteq M \times M$ und $S \subseteq M \times M$
- Wir definieren

  $$RS := \{ (x, y) \mid \exists z \in M.\ (x, z) \in R \wedge (z, y) \in S \}$$

- Wir definieren $R^0 := \{ (x, x) \mid x \in M \}$ und dann $R^{n+1} := RR^n$
- Damit können wir dann definieren ($n \in N_0$):
  - $R^* := \bigcup_{n \geq 0} R^n$ (transitive, reflexive Hülle)
  - $R^+ := \bigcup_{n \geq 1} R^n$ (transitive Hülle)

Erzeugte Sprache, Ableitung

Die von G definierte (erzeugte, dargestellte) *Sprache* ist

$$L(G) := \{w \in \Sigma^* \mid S \Rightarrow_G^* w\},$$

wobei $\Rightarrow_G^*$ die reflexive und transitive Hülle von $\Rightarrow_G$ ist.

Eine Folge von Worten $(w_0, w_1, \ldots, w_n)$ mit $w_0 = S$, $w_n \in \Sigma^*$ und $w_i \Rightarrow w_{i+1}$ für $i = 0, \ldots, n-1$ heißt *Ableitung* von w_n.

Was kommt beim nächsten Mal?

- Fortsetzung der Theorie der Formalen Sprachen
- BNF-Grammatik für "unsere" Programmiersprache
- Entscheidungsprobleme