I.2 Grundlagen von Programmiersprachen

Dienstag, 10. Oktober 2017 08:30

Idee der Lösungsbeschreibung: Algorithmus

Konkrete Formulierung in maschinenausführbarer Form: Programm

Algorithmus

- in endlichem Text beschreibbar
- effektiv (durch Maschine) ausführbar
- besteht aus Elementaroperationen, wobei eindeutig festliegt, welche Elementarop. als nächstes ausgeführt wird. (Determinismus)
- ermöglicht Ein- u. Ausgabe, wobei jeder Eingabe genau eine Ausgabe zugeordnet wird (Determiniertheit)

Alg. wird schrittweise durch Maschine ausge-

führt. Alg. terminiert, wenn er nach endlich vielen Schritten abbricht.

Terminierung u. Determinismus u. Determiniertheit werden nicht immer verlangt.

Bsp - Alg ist indeterministisch, aber determiniert.

Eigenschaften eines guten Algorithmus

- Allgemeinheit: Alg. sollte nicht nur ein einziges Problem lösen, sondern Klasse von Problemen.
- Änderbarkeit: sollte leicht modifizierbar sein
- Effizienz: Anzahl d. Rechenschritte sollte mögl. klein sein.
- Robustheit: sollte sich auch bei unzulässigen Eingaben voll definiert verhalten.

Beispiel für Syntax + Semantik: Sprache der natürlichen Zahlen ohne 0

Syntax: Zahlen sind Sequenzen von Ziffern

0, 1, ..., 9, wobei
erste Ziffer ≠ 0

Semantik: Wert einer Zahl
ist letzte Ziffer +
zehnfacher Wert der links
davor stehenden Zahl
(rekursive Def.)

Bsp: 367 ist syntakt.
korrekt

Wert(367) = 7 + 10 · Wert(36)
= 7 + 10 · (6 + 10 · Wert(3))
= 7 + 10 · (6 + 10 · 3)
= 367

007 ist syntakt.
falsch

Bsp. für Alphabet:

- lateinisches Alphabet
(a, b, ..., z)
- ASCII-Code (128
Zeichen)
- $A_1 = \{0, 1\}$
- $A_2 = \{(,), +, -, *, /, a\}$

Bsp für Wörter:

- $A_1^* = \{\varepsilon, 0, 1, 01, 00, 1101, \ldots\}$
- $A_2^* = \{\varepsilon, (), (a+(a-a)), (a+, +-(a), \ldots\}$

Bsp. für Sprachen:

- $L = \{\varepsilon, 1, 10, 11, 1101, \ldots\}$
  $\subseteq A_1^*$ Binärzahlen ohne führende 0
- $EXPR = \{\varepsilon, ((a)), (a+a), (a+(a-a)), \ldots\}$
  $\subseteq A_2^*$ Menge der korrekt geklammerten Ausdrücke

Hier nur Grammatiken, Automaten kommen in Vorlesg.

FoSAP

Bsp - Grammatik:

Satz $\Rightarrow$ Subj Präd Obj
$\Rightarrow$ Art Attr Sub Präd Obj
$\Rightarrow$ der Attr Sub Präd Obj
$\Rightarrow$ der Adj Sub Präd Obj
$\Rightarrow$ ...

⇒ der kleine Hund jagt die große Katze

das große Katze jagt der kleine bissige Katze

ist auch in der Sprache

die Katze der Hund ist nicht in der Sprache

Grammatik ist Vier-Tupel (N, T, P, S)

N: endl. Menge von Nicht-Terminalsymbolen
- Symbole f. syntakt. Abstraktionen
- Bsp: Satz, Subjekt, ...
- treten nicht in Wörtern der Sprache auf
- werden durch Produktionsregeln ersetzt, bis nur noch Terminalsymbole übrig sind

T: endl. Menge von Terminalsymbolen mit $N \cap T = \emptyset$
- Zeichen, aus denen Wörter d. Sprache bestehen
- Bsp: der, die, Hund, jagt, ...

P: endl. Menge von Produktionsregeln $x \to y$

- Regel $x \to y$ bedeutet, dass man Teilwort x durch Teilwort y ersetzen kann.
- $x \in V^* N V^*$, wobei:
  $V = N \cup T$ (Vokabular)
  $y \in V^*$
  d.h: auf der linken Seite jeder Regel ist mind. ein Nichtterminalsymbol.
- Bsp: Prädikat → jagt

  der Hund Prädikat die Katze ⇒
  der Hund jagt die Katze

S: Startsymbol

- ist ein spezielles Nichtterminalsymbol aus N, aus dem alle Wörter d. Sprache hergeleitet werden können.
- Bsp: Satz

Ableitung

- Relation ⇒ auf V^*
- Für $u, v, y \in V^*$ und $x \in V^* N V^*$ gilt:

$u x v \Rightarrow u y v$

falls $x \to y \in P$

Die von Grammatik $G = (N, T, P, S)$ erzeugte Sprache ist

$L(G) = \{w \mid w \in T^*, S \Rightarrow \ldots \Rightarrow w\}$

Zwei Grammatiken G_1, G_2 sind äquivalent gdw. $L(G_1) = L(G_2)$.

Sprache der Beispiel-Grammatik

$A \Rightarrow aBbc \Rightarrow dc$

$\Rightarrow aaBbbc \Rightarrow adbc$

$\Rightarrow a^3Bb^3c \Rightarrow a^2db^2c$

$\Rightarrow a^4Bb^4c \ldots$

$L(G) = \{a^n d b^n c \mid n \in \mathbb{N}\}$

Diese Sprache ist kontextfrei, denn es gibt auch eine kontextfreie Grammatik,

die sie erzeugt:

$A \to B\ c$

$B \to C\ d\ D$

$C \to \varepsilon \qquad D \to \varepsilon$

$C \to a\,C \qquad D \to b\,D$

geht nicht, denn diese Sprache ist $\{a^n d\, b^m c \mid n, m \in \mathbb{N}\}$

$A \to B\ c$

$B \to a\ B\ b$

$B \to d$

EBNF

- aus $A \to y$ wird $A = y$
- aus $A \to y_1, \ldots, A \to y_n$ wird $A = (y_1 \mid y_2 \mid \ldots \mid y_n)$
- aus $A \to x\,z$, $A \to x\,y\,z$ wird $A = x\ [y]\ z$
- aus $A \to x\,A$, $A \to y$ wird $A = \{x\}\ y$

Bsp-Grammatik in EBNF:

Satz = Subj Präd Obj

Subj = Art Attr Subst

Artikel = [("der" | "die" | "das")]

Attribut = {Adjektiv}

Adj = ("kleine" | "große" | "bissige")

⋮