

Übungen zu „Formale Methoden“

Aufgabe 1 Grundlegende Begriffe [LÖSUNG]

Sämtliche Begriffe sind sehr anschaulich und viel ausführlicher im Skript beschrieben. Hier sei deswegen nur eine kurze Zusammenfassung gegeben.

- *Zugriffssicht (Black Box View) & Realisierungssicht (Glass Box-View)*
Die *Zugriffssicht* ist die Sicht auf ein System von außen aus dem Blickwinkel des Benutzers. Den Benutzer interessiert hierbei *nicht* die konkrete Implementierung oder die verwendeten Datenstrukturen, die benutzt werden, sondern lediglich die Schnittstelle des Systems, also die Art und Weise, wie er mit dem System interagieren muss, um eine gewünschte Aktion auszuführen.
In der *Realisierungssicht* wird zusätzlich zu den Informationen der Zugriffssicht noch der innere Aufbau (Implementierung) des Systems angegeben. So werden jetzt konkrete Datenstrukturen und die zugehörigen Algorithmen angegeben, wie sie z.B. für einen Implementierer des Systems interessant sind.
- *Rechenstruktur (RS), Signatur, Funktion & Konstruktor*
Eine *RS* (auch *Algebra*, *mathematische Struktur*) ist eine Menge von Datenmengen und darauf operierenden Funktionen. Am Beispiel eines Automobils sind mögliche Funktionen z.B. Aktionen wie „*Bremsen*“, „*Fenster schließen*“, „*Zielort im NAVI eingeben*“, usw., die Eingaben (Aktionen) auf Ausgaben (Wirkungen) abbilden. *Konstrukturen* sind eine bestimmte Art von Funktionen, die zum (induktiven) Aufbau der Datenelemente einer *RS* (bzw. der dazugehörigen Sorte) dienen. Die Bezeichnungen der Mengen, auf denen die Funktionen operieren (Sortenbezeichnungen), bilden zusammen mit den Funktionsbezeichnungen und den entsprechenden Angaben über Eingabe- und Ergebnissorten dieser Funktionen die *Signatur*. Eine Signatur beschreibt also die Schnittstelle einer Rechenstruktur und stellt somit in gewisser Weise eine Abstraktion einer Rechenstruktur dar.
- *Sorte & Trägermenge*
Eine *Sorte* ist eine Bezeichnung für eine Datenmenge von Elementen eines bestimmten Typs (Die Begriffe *Sorte* und *Typ* werden in der Vorlesung synonym verwendet). Eine *Trägermenge* ist eine (Grund-) Menge von Datenelementen gleicher Sorte. Trägermengen zur Sorte *Nat* der natürlichen Zahlen wären z.B. die Mengen der Strichzahlen oder der römischen Zahlen.

Aufgabe 2 Modelle, Daten- und Rechenstrukturen [LÖSUNG]

- (1) Mögliche Beispielabläufe sind z.B. durch die Vielzahl der möglichen Interaktionen in einem Automobil gegeben. Geeignete Mittel zur Beschreibung von Beispielabläufen wären z.B. diverse Formen von Zustandsautomaten (Zustandsübergangsdiagramme) oder Message Sequence Charts (MSCs). Eine eher informellere Art der Beschreibung könnte in Form von Use-Cases bzw. natürlicher Sprache (z.B. Deutsch) gegeben werden. Je nach konkretem Beispielablauf bzw. Domäne sind die verschiedenen Beschreibungsmethoden jeweils besser bzw. weniger gut geeignet, da sie unterschiedliche Sichten auf einen Beispielablauf betonen (MSCs geben eher das Interaktionsverhalten wieder, während Zustandsautomaten die Gruppierung bzw. Äquivalenzrelation auf der Zustandsmenge betonen.)
- (2) Zur Implementierung der Konzepte und Funktionen in einem Automobil sind z.B. folgende Datenstrukturen denkbar:
 - *Record-Strukturen*: Um die aktuellen Verschleißwerte (z.B.: Ölstand, Verschleißgrad der Bremsbeläge, Luftdruck,..) des Autos zu einem bestimmten Zeitpunkt in Form eines einzelnen Tupels zu verwalten.
 - *Stack*: Um die Funktionalität eines Tempomats zu steuern (Kurzfristiges Beschleunigen mit anschließendem Zurückkehren auf die Grundgeschwindigkeit).
 - *Liste*: Liste der Wegpunkte einer Strecke im Navigationsgerät.

Aufgabe 3 Wörter vs. Terme [LÖSUNG]

- (a) Die Grammatik G ist kontextfrei (Typ 2), denn es gilt für alle Produktionen

$$p : w_l \rightarrow w_r$$

aus P , dass $|w_l| \leq |w_r|$ und dass alle linken Seiten w_l jeweils nur aus einer einzelnen Variablen bestehen.

- (b) Eine mögliche Ableitung des Wortes $a + a * a$ ist z.B. gegeben durch:

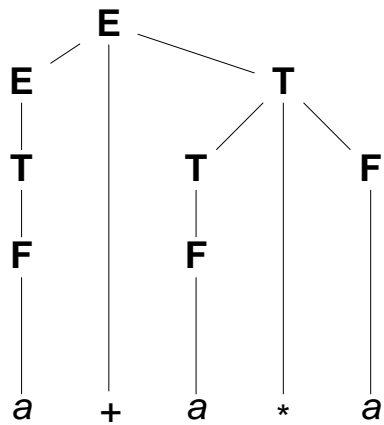
$$\begin{aligned} A_1 &= E \rightarrow E + T \rightarrow T + T \rightarrow F + T \rightarrow a + T \rightarrow a + T * F \\ &\rightarrow a + F * F \rightarrow a + a * F \rightarrow a + a * a \end{aligned}$$

Die Ableitung A_1 stellt die Linksableitung dar. Jede Ableitung besitzt einen eindeutigen Syntaxbaum, jedoch kann derselbe Syntaxbaum zu mehreren unterschiedlichen Ableitungen gehören. Der entsprechende Syntaxbaum für die Linksableitung A_1 von w ist:

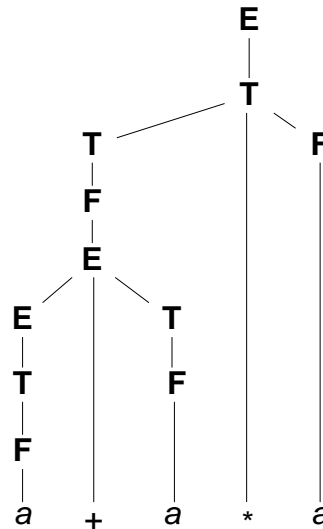
Die Ableitung von w ist nicht eindeutig, da für w auch noch weitere Ableitungen existieren, z.B. die Rechtsableitung oder die folgende Ableitung A_2 , deren Syntaxbaum sich von dem der Linksableitung unterscheidet:

$$\begin{aligned} A_2 &= E \rightarrow T \rightarrow T * F \rightarrow F * F \rightarrow E * F \rightarrow E + T * F \rightarrow T + T * F \\ &\rightarrow F + T * F \rightarrow a + T * F \rightarrow a + F * F \rightarrow a + a * F \rightarrow a + a * a \end{aligned}$$

Da in der Sprache $\mathcal{L}(G)$ für das Wort w verschiedene Syntaxbäume existieren, nennt man die Sprache $\mathcal{L}(G)$ auch mehrdeutig.



(a) Syntaxbaum der Linksableitung A_1 von w .



(b) Syntaxbaum der Ableitung A_2 .

- (c) Legt man für die Sorte D z.B. die Trägermenge $D_1 = \{1, 2, 3\}$ zugrunde, so lassen sich z.B. folgende Terme über der Signatur Σ erzeugen:

- es
- $ph(es, tp(pp(es)))$
- $ph(ph(es, tp(ph(es, 3))), 2)$

- (d) Über der Signatur $\Sigma_{arith} = (S, F)$, mit z.B.

$$S = \{S_1\} \quad \text{und} \quad F = \{ +, * : S_1, S_1 \rightarrow S_1 \}$$

lassen sich Terme erzeugen, die die gleiche syntaktische Gestalt wie Wörter aus $\mathcal{L}(G)$ haben. Um konkrete (variablenfreie) Terme zu erzeugen muss weiterhin noch eine Konstante $a : S_1$ mit in die Signatur aufgenommen werden. Läßt man Terme mit Identifikatoren (Formeln) zu, so reicht die Angabe aus, dass a ein Identifikator der Sorte S_1 sei. Die Sortenbezeichnung S_1 steht hierbei für eine beliebige Sorte.

- (e) Über der Signatur Σ_{arith} entsprechen die beiden folgenden Term t_1 und t_2 dem Wort $w = a + a * a \in \mathcal{L}(G)$ (Zur Unterscheidung der Struktur sei hier die Prefix-Notation für die Funktionen $+$ und $*$ verwendet.):

$$t_1: +(a, *(a, a))$$

(Entspricht $a + (a * a)$ in Infix-Notation mit expliziter Klammerung.)

$$t_2: *(+(a, a), a)$$

(Entspricht $(a + a) * a$ in Infix-Notation mit expliziter Klammerung.)

Während dasselbe Wort w der kontextfreien Sprache $\mathcal{L}(G)$ auf unterschiedliche Arten abgeleitet werden kann (w besitzt unterschiedliche Syntaxbäume) existiert für die Terme t_1 bzw. t_2 über Σ_{arith} jeweils immer eine eindeutige Verschachtelung von Funktionsaufrufen. In Infix-Notation ohne explizite Klammerung sind die beiden Terme jedoch identisch.