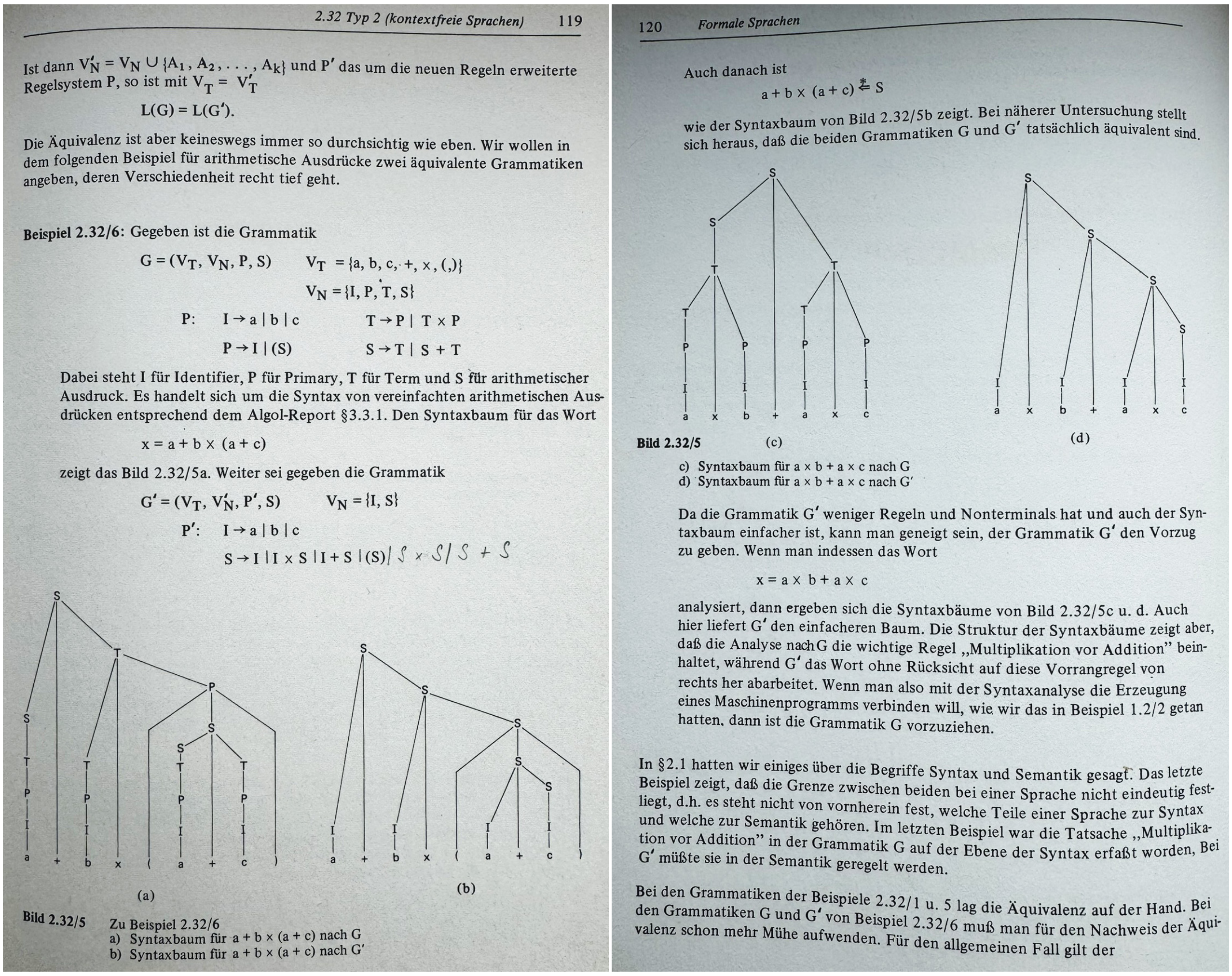
Aufgabenblatt Nr. 14 inf12 13.05.2024

35. Die in Aufgabe 34 formulierten Grammatiken $\mathbf{G_1}$ und $\mathbf{G_2}$ sind entnommen dem Werk „Rudolf Herschel: Einführung in die Theorie der Automaten, Sprachen und Algorithmen", Verlag Oldenbourg 1986:



Ist dann $V_N' = V_N \cup \{A_1, A_2, \ldots, A_k\}$ und P' das um die neuen Regeln erweiterte Regelsystem P, so ist mit $V_T = V_T'$

$$L(G) = L(G').$$

Die Äquivalenz ist aber keineswegs immer so durchsichtig wie eben. Wir wollen in dem folgenden Beispiel für arithmetische Ausdrücke zwei äquivalente Grammatiken angeben, deren Verschiedenheit recht tief geht.

Beispiel 2.32/6: Gegeben ist die Grammatik

$$G = (V_T, V_N, P, S) \qquad V_T = \{a, b, c, +, \times, (,)\}$$
$$V_N = \{I, P, T, S\}$$

$$P: \quad I \to a \mid b \mid c \qquad T \to P \mid T \times P$$
$$P \to I \mid (S) \qquad S \to T \mid S + T$$

Dabei steht I für Identifier, P für Primary, T für Term und S für arithmetischer Ausdruck. Es handelt sich um die Syntax von vereinfachten arithmetischen Ausdrücken entsprechend dem Algol-Report §3.3.1. Den Syntaxbaum für das Wort

$$x = a + b \times (a + c)$$

zeigt das Bild 2.32/5a. Weiter sei gegeben die Grammatik

$$G' = (V_T, V_N', P', S) \qquad V_N = \{I, S\}$$

$$P': \quad I \to a \mid b \mid c$$
$$S \to I \mid I \times S \mid I + S \mid (S) \mid S \times S \mid S + S$$

(a) (b)

Bild 2.32/5 Zu Beispiel 2.32/6
a) Syntaxbaum für $a + b \times (a + c)$ nach G
b) Syntaxbaum für $a + b \times (a + c)$ nach G'

Auch danach ist

$$a + b \times (a + c) \stackrel{*}{\Leftarrow} S$$

wie der Syntaxbaum von Bild 2.32/5b zeigt. Bei näherer Untersuchung stellt sich heraus, daß die beiden Grammatiken G und G' tatsächlich äquivalent sind.

Bild 2.32/5 (c) (d)

c) Syntaxbaum für $a \times b + a \times c$ nach G
d) Syntaxbaum für $a \times b + a \times c$ nach G'

Da die Grammatik G' weniger Regeln und Nonterminals hat und auch der Syntaxbaum einfacher ist, kann man geneigt sein, der Grammatik G' den Vorzug zu geben. Wenn man indessen das Wort

$$x = a \times b + a \times c$$

analysiert, dann ergeben sich die Syntaxbäume von Bild 2.32/5c u. d. Auch hier liefert G' den einfacheren Baum. Die Struktur der Syntaxbäume zeigt aber, daß die Analyse nach G die wichtige Regel „Multiplikation vor Addition" beinhaltet, während G' das Wort ohne Rücksicht auf diese Vorrangregel von rechts her abarbeitet. Wenn man also mit der Syntaxanalyse die Erzeugung eines Maschinenprogramms verbinden will, wie wir das in Beispiel 1.2/2 getan hatten, dann ist die Grammatik G vorzuziehen.

In §2.1 hatten wir einiges über die Begriffe Syntax und Semantik gesagt. Das letzte Beispiel zeigt, daß die Grenze zwischen beiden bei einer Sprache nicht eindeutig festliegt, d.h. es steht nicht von vornherein fest, welche Teile einer Sprache zur Syntax und welche zur Semantik gehören. Im letzten Beispiel war die Tatsache „Multiplikation vor Addition" in der Grammatik G auf der Ebene der Syntax erfaßt worden, Bei G' müßte sie in der Semantik geregelt werden.

Bei den Grammatiken der Beispiele 2.32/1 u. 5 lag die Äquivalenz auf der Hand. Bei den Grammatiken G und G' von Beispiel 2.32/6 muß man für den Nachweis der Äquivalenz schon mehr Mühe aufwenden. Für den allgemeinen Fall gilt der

a) Zeige: Die Grammatiken $\mathbf{G_1}$ und $\mathbf{G_2}$ sind, entgegen der Behauptung auf Seite 120 oben, nicht äquivalent, denn das Wort **(a + b) * c** gehört zwar zur Sprache $\mathbf{L(G_1)}$, nicht aber zur Sprache $\mathbf{L(G_2)}$.

b) Verifiziere: Wenn man die Produktionen **P** der Grammatik $\mathbf{G_2}$ um die Regel

   (6) $\quad S \to S * S$

   ergänzt, läßt sich das Wort **(a + b) * c** auch in der Grammatik $\mathbf{G_2}$ auf das Startsymbol **S** reduzieren.

36. Beispiel einer **Grammatik vom Typ 1 (kontextsensitive Grammatik)**

Rückblick:
*Eine Grammatik **G** und die zugehörige Sprache **L(G)** heißen **kontextfrei** oder **vom Typ 2** genau dann, wenn die linke Seite jeder Produktionsregel aus genau einem Nonterminal-Symbol und die rechte Seite aus einer beliebigen Aneinanderreihung von Terminal- und Nonterminal-Symbolen besteht.*

Beachte:
*Die Syntaxanalyse („bottom-up") eines zur **kontextfreien Sprache L(G)** gehörenden Wortes **w** (z. B. arithmetischer Term, Quelltext einer Programmiersprache) läßt sich als Syntaxbaum realisieren, dessen Wurzel das Startsymbol **S** ist, dessen innere Knoten aus Nonterminals und dessen Endknoten („Blätter") aus Terminals bestehen.*

Folgende Grammatik **G** sei durch das Quadrupel **(T; N; S; P)** gegeben:

T := {a, b, c}

N:= {B, C, **S**}   mit   **S** = Startsymbol

Produktionen **P**:

(1) S $\rightarrow$ aSBC | aBC

(2) CB $\rightarrow$ BC

(3) aB $\rightarrow$ ab

(4) bB $\rightarrow$ bb

(5) bC $\rightarrow$ bc

(6) cC $\rightarrow$ cc

Zeige anhand von Beispielen (Linksableitungen für die Wörter abc, aabbcc, aaabbbccc):

Die zur Grammatik **G** gehörende Sprache ist

L(G) $= \{ w \mid w = a^n b^n c^n,\ n \in \mathbb{N} \}$
$= \{ abc, aabbcc, aaabbbccc, aaaabbbbcccc, \ldots \}$.

Die Grammatik **G** ist nicht kontextfrei; vielmehr verlangen die Regeln (2) bis (6), daß die Nonterminals auf der linken Seite nur dann ersetzt werden können, wenn sie in einem bestimmten Kontext mit anderen Zeichen (Terminals oder Nonterminals) stehen. Die Ersetzungsregeln (2) bis (6) sind folglich kontextsensitiv, und die zughörige Grammatik **G** heißt **kontextsensitiv** oder vom **Typ 1**.

Hinweis:
*Zu dieser Sprache **L(G)** läßt sich keine kontextfreie Grammatik (Grammatik vom Typ 2) angeben.*