5. Die syntaktische Analyse

- mittels sog. Parser
- Input: Folge von Token (Symbolen), geliefert vom Scanner/Sieber
- Aufgabe: Teilfolgen zusammenfassen zu größeren syntaktischen Einheiten
- Ausdrücke, Anweisungen(-folgen), Deklarationen, Spezifikationen
- Qutput: geeignete Darstellung der syntaktischen Struktur des Programms (i.d.R. Syntaxbaum/Parse-Tree)
- wichtige „Nebenaufgabe“: Erkennen von Syntaxfehlern

5. Die syntaktische Analyse

- Beschreibung der syntaktischen Struktur einer Programmiersprache: durch kontextfreie Grammatik
- Grammatik ist Input für Parsergeneratoren
- zu jeder kontextfreien Grammatik Kellerautomat konstruierbar, der Sprache akzeptiert
- 2 Klassen von Syntaxanalyseverfahren:
  - Top-Down-Analysatoren: Analyse beginnt mit Startsymbol der Grammatik, Konstruktion des Syntaxbaums mit der Wurzel
  - Bottom-up-Parser: beginnen mit Programm als Eingabewort, suchen für immer längere Anfangsstücke syntaktische Struktur, bis Produktion der Grammatik entdeckt wurde

5. Die syntaktische Analyse

Wiederholung/Definition:

kontextfreie Grammatik (kfG) ist Viertupel
G = (Vn, Vt, P, S) mit
Vn, Vt disjunkte Alphabete,
Vn Menge der Nichtterminalsymbole,
Vt Menge der Terminalsymbole,
P aus Vn x (Vn ∪ Vt)* Menge der Produktionsregeln
S ϵ Vn Startsymbol

Terminalsymbole: Symbole, die im Programmcode wirklich auftreten
Nichtterminalsymbole: Stellvertreter für Mengen von Wörter,
die sie produzieren (ableiten)

5. Die syntaktische Analyse

Beispiel:

G1 = ({E,T,F}, {+,*,(,),id}, {E→E+T|T, T →T*F|F, F →(E)|id}, E)

Produktionsregel: Ersetzen der linken Seite durch rechte Seite

„produziert“ neue Wörter

E ⇨ E+T ⇨ T+T ⇨ T*F +T ⇨ T * id +T ⇨ F * id +T ⇨ F * id +F

⇨ id * id + F ⇨ id * id + id,

also E ⇨* id * id + id

5. Die syntaktische Analyse

Wiederholung/Definition:

Sei G = (Vn, Vt, P, S) eine kfG. Die von G definierte (erzeugte) Sprache ist **L(G)** = {u ϵ Vt* | S ⇨* u}. Ein Wort x ϵ L(G) heißt **Satz** von G.

Beispiel: id*id+id ϵ L(G1)

Beispiel: G2 = ({E},{+,*,(,),id}, {E→E+E | E*E | (E) | id}, E}

Aufgabe: Zeigen Sie: id*id+id ϵ L(G2)

5. Die syntaktische Analyse

Wiederholung/Definition Syntaxbaum:

Sei G = (Vn, Vt, P, S) eine kfG. Sei B ein geordneter Baum. Seine Blätter seien markiert mit Symbolen aus Vt U {ε}, die innerern Knoten mit Symbolen aus Vn.

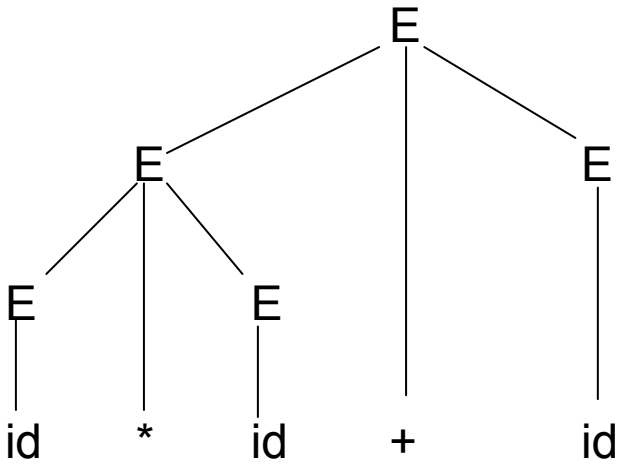
B heißt **Syntaxbaum** für x ϵ Vt* und X ϵ Vn , wenn gilt:

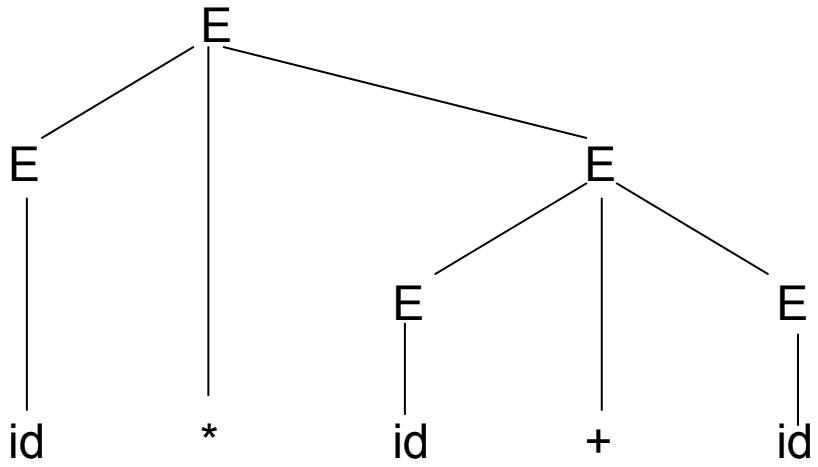
1. Die Wurzel ist markiert mit X.
2. Das Blattwort von B (=Konkatenation der Markierungen der Blätter von links nach rechts) ist x.
3. Ist n ein bel. innerer Knoten, markiert mit Nichtterminal A, so sind entweder seine Kinder von links nach rechts markiert mit N1,N2, ..,Nk und A→N1N2..Nk ist Produktion in P,

   oder sein einziges Kind ist markiert mit ε und A → ε ist eine Produktion in P.

5. Die syntaktische Analyse

Beispiel: Syntaxbäume für id*id+id bzgl. G2





Satz x ϵ L(G) **mehrdeutig**: es ex. mehr als ein Strukturbaum für x

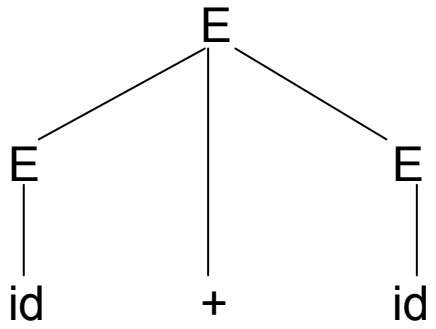
kfG G **mehrdeutig**: G enthält mind. einen mehrdeutigen Satz (Bsp: G2)

G **eindeutig**: G ist nicht mehrdeutig

Aufgabe: Ist G1 eindeutig oder mehrdeutig?

5. Die syntaktische Analyse

Beispiel: Syntaxbaum für id+id bzgl. G2



2 Ableitungen:

E ⇨ E + E ⇨ id + E ⇨ id + id (Linksableitung)

E ⇨ E + E ⇨ E + id ⇨ id + id (Rechtsableitung)

Linksableitung: es wird zuerst immer das am weitesten links stehende Nichtterminal ersetzt

Rechtableitung: es wird zuerst immer das am weitesten rechts stehende Nichtterminal ersetzt

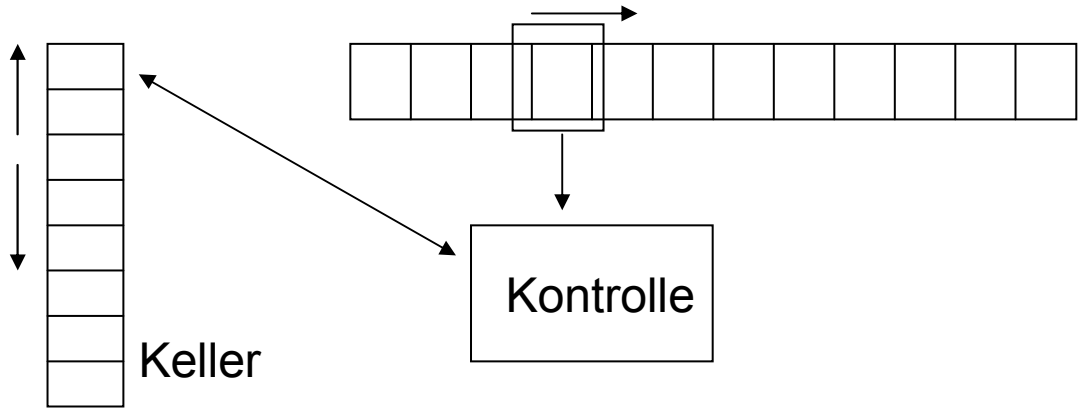
5. Die syntaktische Analyse

kfG zur Syntaxbeschreibung einer PS:

muss eindeutig sein

→ zu syntaktisch korrektem Programm (= Satz der Grammatik) ex. genau ein Syntaxbaum, genau eine Links- und genau eine Rechtsableitung

Akzeptor der kfGrammatiken: Kellerautomat



5. Die syntaktische Analyse

Wiederholung/Definition:

Kellerautomat (KA) ist 5-Tupel $P=(V,Q, \Delta, q0,F)$ wobei

V	Eingabealphabet
Q	endl. Menge von Zuständen
$q0 \in Q$	Anfangszustand
F aus Q	Menge an Endzuständen
Δ	Übergangsrelation $Q^+ \times (V \cup \{\varepsilon\}) \rightarrow Q^*$

Deterministischer KA: zu jeder Konfiguration gibt es höchstens einen Übergang, für ein akzeptiertes Wort genau eine Folge von Konfigurationen.

5. Die syntaktische Analyse

- zu jeder kfG kann (nichtdeterministischer) KA konstruiert werden, der die von kfG definierte Sprache akzeptiert
- benutzt man KA für Compiler: zusätzlich interessiert syntaktische Struktur
- $\rightarrow$ KA mit Ausgabe: P = (V, Q, O, Δ, q0,F) wobei O endl. Ausgabealphabet
- KA mit Ausgabe als Parser: als Ausgabealphabet die Produktionen der KfG (bzw. Stellvertreternummern)

5. Die syntaktische Analyse

Beispiel:

G1 = ({E,T,F}, {+,*,(,),id}, {E→E+T|T, T →T*F|F, F →(E)|id}, E)

Versuch:

Herleitung von id + id * id

Start: E

Problem: Welche der Regeln E→E+T bzw. E → T soll genommen werden?

LL(1)-Kriterium: auf nächstes Eingabesymbol schauen -> id

Problem: id kann Anfangssymbol eines aus T und aus E+T hergeleiteten Satzes sein!

Fazit: G1 ist keine LL(1)-Grammatik!