

Kapitel 6: Syntaxgesteuerte Übersetzung

Prinzip

Operationen der Übersetzung werden an die Grammatik gebunden
→ Compiler-Generatoren

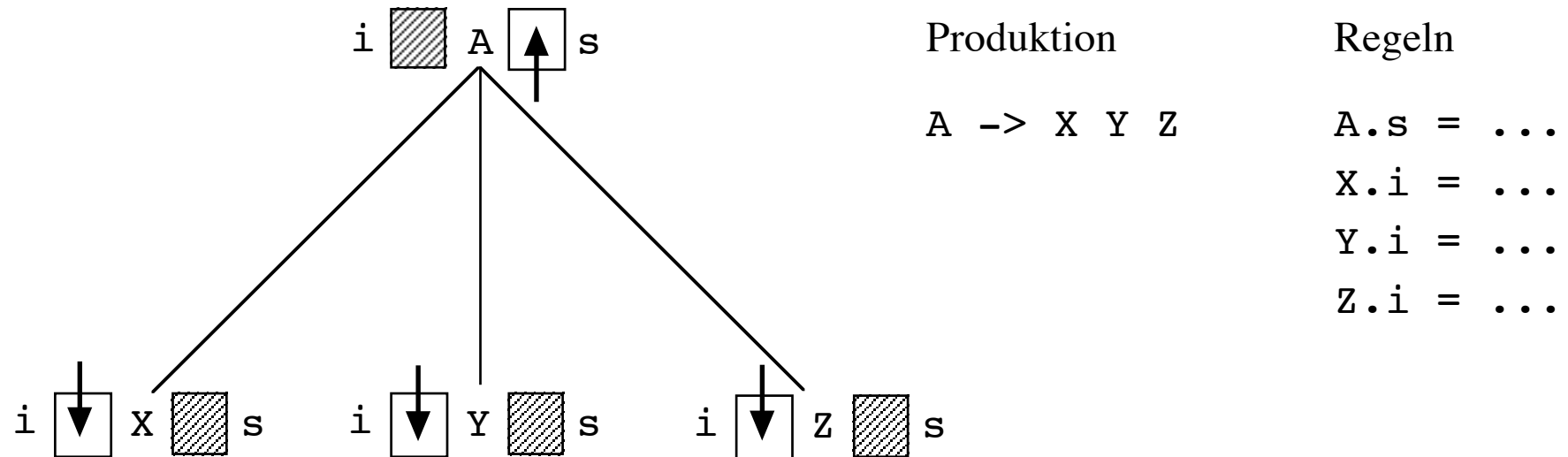
Themen

- Attributierte Grammatik
- L-attributierte Grammatik
- Übersetzungsschema
- Compiler – Generatoren

Attributierte Grammatik

... ist eine kontextfreie Grammatik mit folgenden Erweiterungen

- jedes Grammatik-Symbol kann eine Menge von **Attributen** haben
- jeder Produktion werden **Attribut-Berechnungsregeln** zugeordnet



Es gibt **synthetisierte** (aufsteigende) und **ererbte** (absteigende) Attribute

AG – Beispiel

Der Wert von Zahlen in verschiedenen Zahlensystemen soll berechnet werden

Schreibweise <Ziffernfolge>/<Basis>, z.B. 3C5/16

Ziffern 0–9, A–Z

Basis 2–35

Grammatik

$A \rightarrow F / B$

$F \rightarrow F Z$

$F \rightarrow Z$

$Z \rightarrow '0'$

$Z \rightarrow '1'$

•

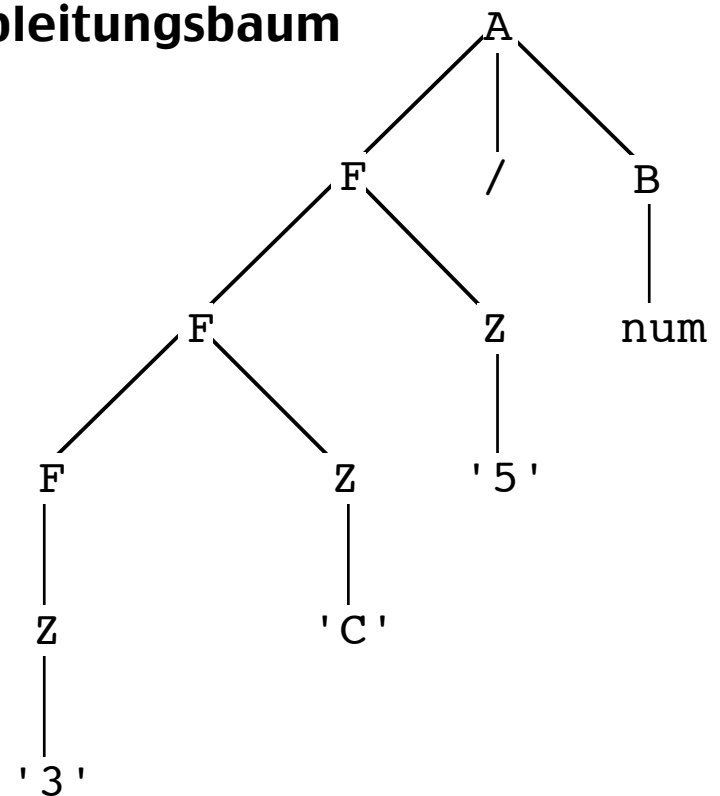
•

$Z \rightarrow 'Y'$

$Z \rightarrow 'Z'$

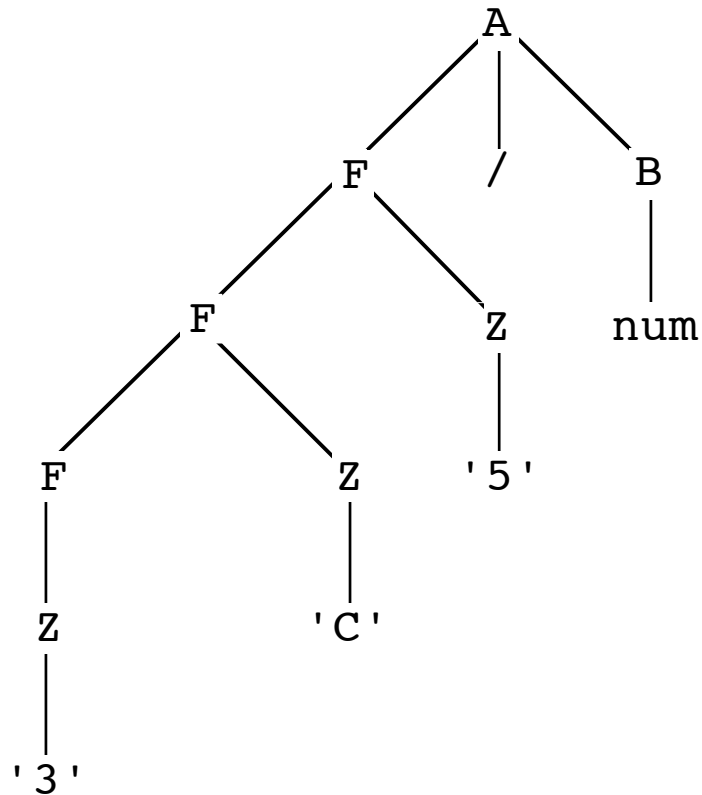
$B \rightarrow \text{num}$

Ableitungsbaum



AG – Beispiel: (1) Anfangswerte

Terminale liefern die Anfangswerte für Attributberechnungen



Produktionen

Regeln

Z \rightarrow '0'

Z.v = 0

Z \rightarrow '1'

Z.v = 1

.

.

Z \rightarrow 'Y'

Z.v = 34

Z \rightarrow 'Z'

Z.v = 35

B \rightarrow num

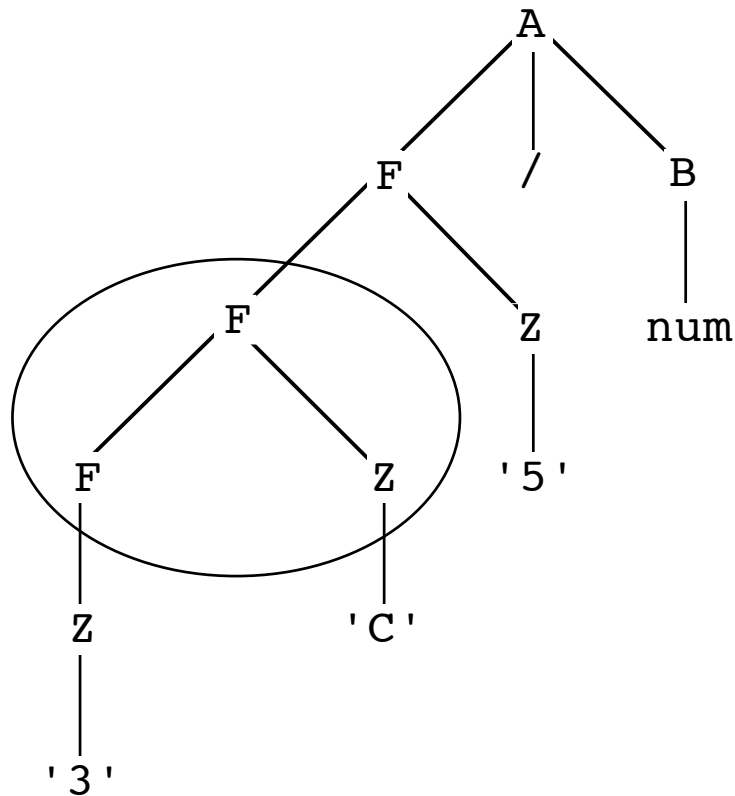
B.v = num.x

F \rightarrow Z

F.v = Z.v

AG – Beispiel: (2) Berechnungen

Berechnungen erfolgen i.a. aufsteigend zur Wurzel hin (strukturelle Semantik)



Produktionen

$F_1 \rightarrow F_2 Z$

$A \rightarrow F / B$

Regeln

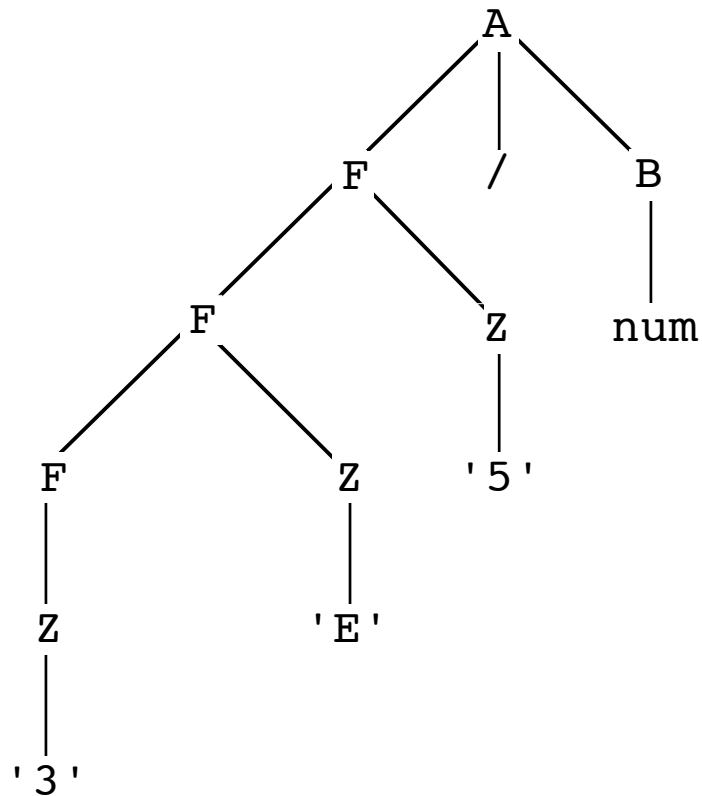
$F_1.v = F_2.v * b + Z.v$

$A.v = F.v$

In den Berechnungsregeln einer AG gibt es keine globalen Variablen

AG – Beispiel: (3) Kontextinformation

Attributwerte können in benachbarte linke/rechte Teilbäume übertragen und von oben nach unten weitergegeben werden (ererbte Attribute)



Produktionen

$A \rightarrow F / B$

$F_1 \rightarrow F_2 Z$

Regeln

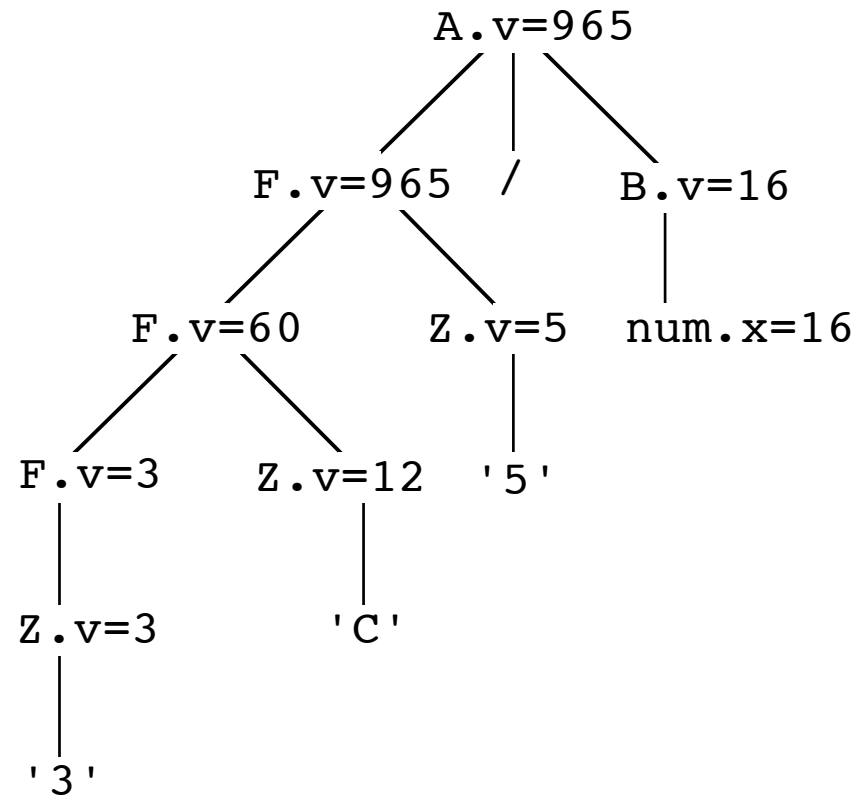
$F.b = B.v$

$F_2.b = F_1.b$

AG – Beispiel: (4) Attributierte Grammatik

Produktionen	Regeln
$A \rightarrow F / B$	$A.v = F.v$ $F.b = B.v$
$F_1 \rightarrow F_2 Z$	$F_1.v = F_2.v * F_1.b + Z.v$ $F_2.b = F_1.b$
$F \rightarrow Z$	$F.v = Z.v$
$Z \rightarrow '0'$	$Z.v = 0$
$Z \rightarrow '1'$	$Z.v = 1$
.	
.	
$Z \rightarrow 'Y'$	$Z.v = 34$
$Z \rightarrow 'Z'$	$Z.v = 35$
$B \rightarrow \text{num}$	$B.v = \text{num.x}$

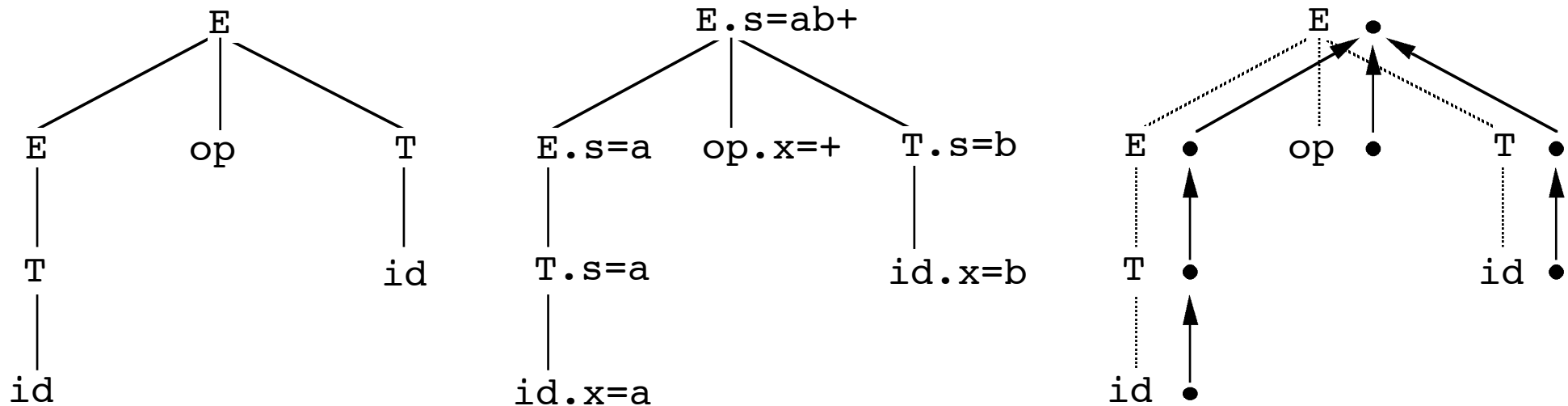
Attributierter Ableitungsbaum



Attributierte LR-Grammatik für Ausdrücke (Infix=>Postfix)

Produktion	Regeln
$E \rightarrow E_1 \text{ op } T$	$E.s = \text{conc}(E_1.s, T.s, \text{op}.x)$
$E \rightarrow T$	$E.s = T.s$
$T \rightarrow (E)$	$T.s = E.s$
$T \rightarrow \text{id}$	$T.s = \text{id}.x$

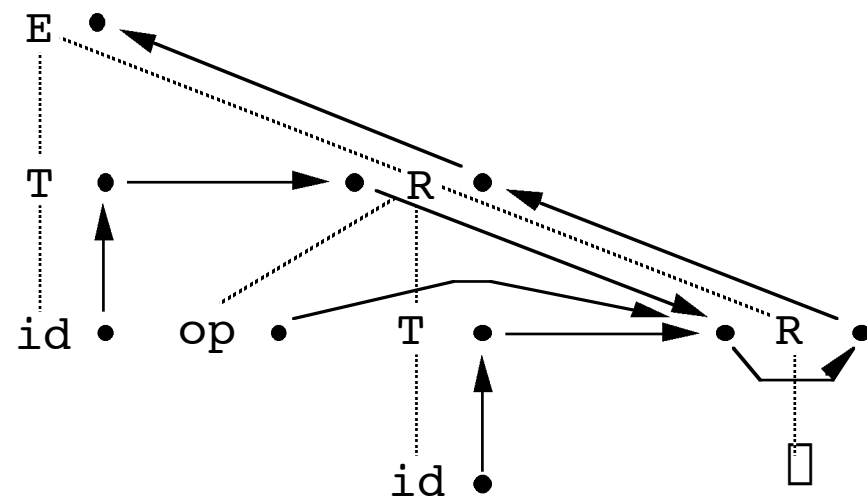
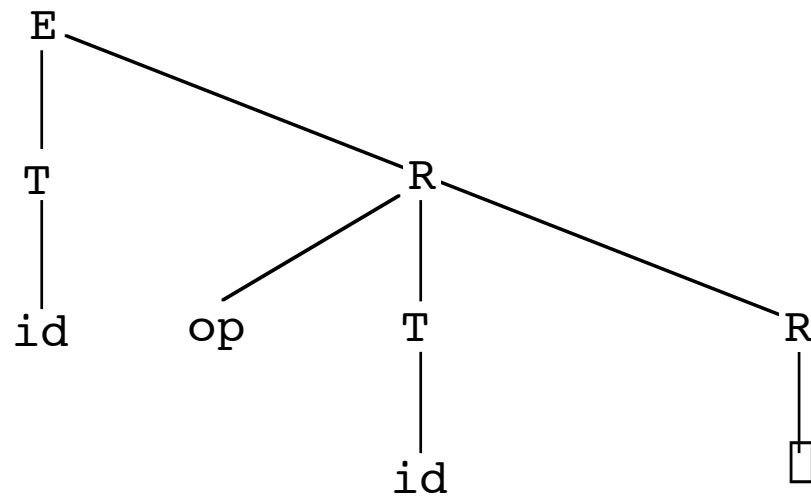
id op id



Eine AG, die nur synthetisierte Attribute hat, heißt **S-attribuiert**

Attributierte LL-Grammatik für Ausdrücke (Infix=>Postfix)

Produktion	Regeln
$E \rightarrow T R$	$R.i = T.s$ $E.s = R.s$
$R \rightarrow op T R_1$	$R_1.i = conc(R.i, T.s, op.x)$ $R.s = R_1.s$
$R \rightarrow \square$	$R.s = R.i$
$T \rightarrow (E)$	$T.s = E.s$
$T \rightarrow id$	$T.s = id.x$



L-attributierte Grammatik

Eine AG ist L-attribuiert (LAG), wenn alle Attribute in einem links-rechts-Tiefendurchlauf des Ableitungsbaums berechnet werden können

LAG-Bedingung:

In jeder Produktion $A \rightarrow \dots X \dots$ dürfen die ererbten Attribute von X nur von Attributen jener Symbole abhängen, die links von X stehen

Jede S-attributierte Grammatik ist L-attribuiert

Übersetzungsschema

- Bei einer LAG kann man die Regeln in die Produktionen $A \rightarrow \dots X \dots$ einsetzen
- Berechnung ererbter Attribute von X werden links von X geschrieben
 - Berechnung synthetisierter Attribute von A werden am Ende plziert

Produktion	Regeln
$E \rightarrow T R$	$R.i = T.s$ $E.s = R.s$
...	

Übersetzungsschema
$E \rightarrow T \{R.i = T.s\} R \{E.s = R.s\}$
...

Die eingesetzten Regeln nennt man auch (semantische) Aktionen

Einpass-Compiler-Generatoren

TOP DOWN

Übersetzungsschema als Vorlage: Syntax-Prozeduren, Attribute als Parameter

BOTTOM UP

Parallel zum Symbol-Stack ein Attribut-Stack, Aktionen nur bei der Reduktion

Problem: Aktionen für ererbte Attribute müssen transformiert werden

$$E \rightarrow T \{R.i = T.s\} R \{E.s = R.s\}$$
$$\square$$
$$E \rightarrow T M R \{E.s = R.s\}$$
$$M \rightarrow \square \{R.i = T.s\}$$

Einfügen von \square -Produktionen kann die LR-Eigenschaft der Grammatik zerstören!

Allgemeine Lösung: Mehrpass-Übersetzung

a) Ableitungsbaum erzeugen

b) Attribute auswerten

Beispiel: Generator Ox