

*Satzformen* sind die Wörter aus  $(N \cup T)^*$ .

Notation:

Wir verwenden oft

- $a, b, c, \dots$  für Terminalsymbole
- $A, B, C, \dots$  für Nonterminale
- $u, v, w, \dots$  für Terminalwörter aus  $T^*$
- $\alpha, \beta, \gamma, \dots$  für Satzformen

# Beispiel

Eine kontextfreie Grammatik:

$$G = (\{E, A\}, \{0, 1, +, -, (, )\}, P, E)$$

mit

$$P = \{E \rightarrow (E + E), E \rightarrow (E - E), E \rightarrow A, A \rightarrow 0, A \rightarrow 1, A \rightarrow AA\}$$

# Ableitungen

Wir definieren die Relation

$$\Rightarrow_G: (N \cup T)^* \rightarrow (N \cup T)^*$$

vermöge

$$\alpha A \beta \Rightarrow \alpha \gamma \beta \text{ falls } A \rightarrow \gamma \in P.$$

$\Rightarrow_G$  heißt Ableitungsrelation.

$\Rightarrow_G^*$  ist die reflexiv-transitive Hülle von  $\Rightarrow_G$

# Beispiel

$$G = (\{E, A\}, \{0, 1, +, -, (, )\}, P, E)$$

mit

$$P = \{E \rightarrow (E + E), E \rightarrow (E - E), E \rightarrow A, A \rightarrow 0, A \rightarrow 1, A \rightarrow AA\}$$

$$E \Rightarrow (E + E) \Rightarrow (E + A) \Rightarrow (A + A) \Rightarrow (AA + A) \Rightarrow (A0 + A) \Rightarrow (A0 + 1) \Rightarrow (10 + 1) \in T^*$$

Wir geben in Zukunft oft nur die Produktionen an.

(Wenn klar ist was  $N$ ,  $T$  und das Startsymbol sind.)

# Sprache einer Grammatik

## Definition

Es sei  $G = (N, T, P, S)$  eine kontextfreie Grammatik.

Wir definieren  $L(G) = \{ w \in T^* \mid S \xRightarrow{*} w \}$ .

$L(G)$  ist die von  $G$  erzeugte Sprache.

Wir nennen  $\alpha \in \beta$  einen *Ableitungsschritt* und

$S \Rightarrow \alpha_1 \Rightarrow \dots \Rightarrow \alpha_{n-1} \Rightarrow w$  eine *Ableitung* von  $w$ .

Eine Ableitung ist eine Linksableitung, falls in jedem Schritt das am weitesten links stehende Nonterminal ersetzt wird.

(Rechtsableitung analog).

# Beispiel

CFG  $G$  mit  $S \rightarrow aSSb$ ,  $S \rightarrow \epsilon$ .

$S \Rightarrow aSSb \Rightarrow aaSSbSb \Rightarrow aaSbSb \Rightarrow aabSb \Rightarrow aabaSSbb \Rightarrow$   
 $aabaSbb \Rightarrow aababb$

ist eine Linksableitung von  $aababb$ .

# Ableitungsbäume

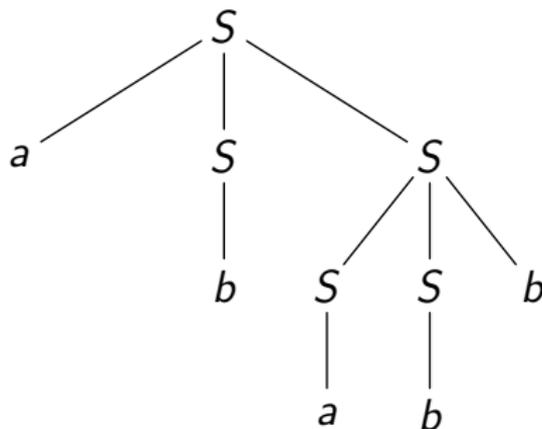
## Definition

Ein Ableitungsbaum ist ein Baum:

- 1 er hat eine Wurzel
- 2 er ist orientiert
- 3 innere Knoten sind Nonterminale
- 4 Blätter sind Terminalsymbole
- 5 Kinder eines inneren Knotens  $A$  sind rechte Seite einer Produktion  $A \rightarrow \alpha$

# Beispiel

$$S \rightarrow aSS \mid SSb \mid a \mid b$$



Zugehörige Linksableitung:

$$S \Rightarrow aSS \Rightarrow abS \Rightarrow abSSb \Rightarrow abaSb \Rightarrow ababb$$

## Theorem

Es sei  $G = (N, T, P, S)$  eine CFG und  $w \in T^*$ .

Dann gilt  $w \in L(G)$  genau dann, wenn es einen Ableitungsbaum gibt

- mit Wurzel  $S$ ,
- mit Blättern  $w$ .

## Beweis.

(Idee) Allgemeinere Aussage:

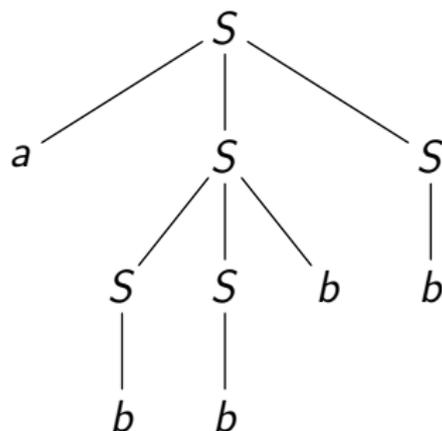
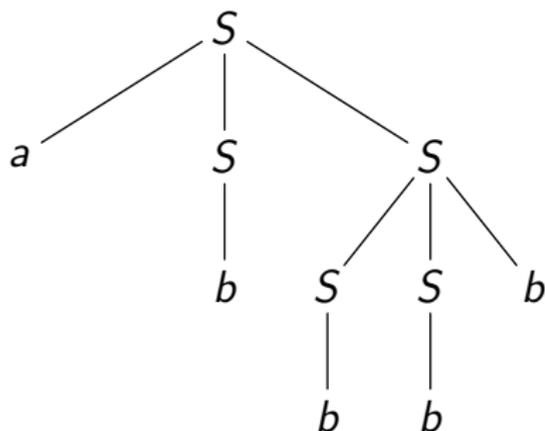
$A \xRightarrow{*} \alpha$  genau dann, wenn Ableitungsbaum mit Wurzel  $A$  und Blättern  $\alpha$ .

Beweis mit Induktion über Länge einer Ableitung. □

Zu jedem Ableitungsbaum gibt es eine *eindeutige* Linksableitung.

# Eindeutigkeit

$$S \rightarrow aSS \mid SSb \mid a \mid b$$

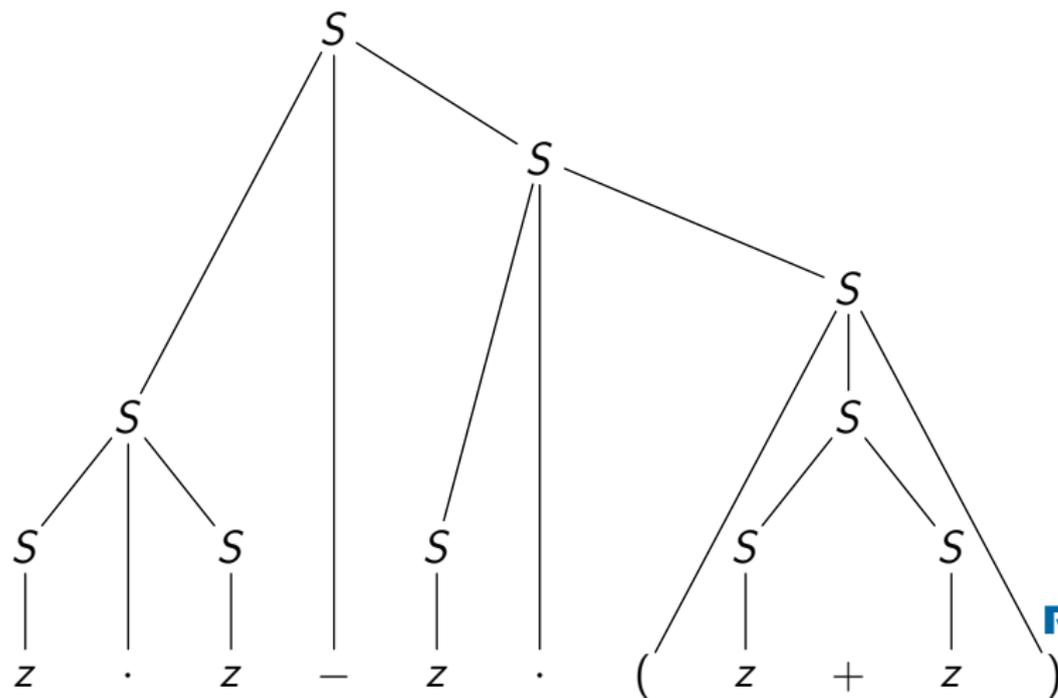


Es gibt zwei verschiedene Linksableitungen.

# Beispiel 1

$$S \rightarrow S + S \mid S - S \mid S \cdot S \mid S / S \mid (S) \mid z$$

Betrachte:  $z \cdot z - z \cdot (z + z)$



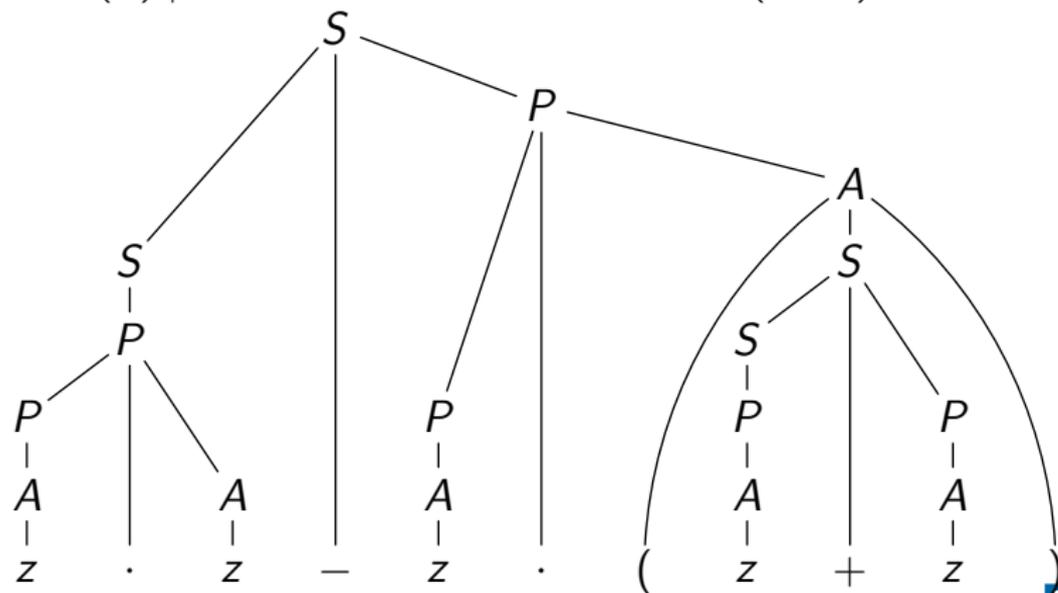
## Beispiel 2

$$S \rightarrow S + P \mid S - P \mid P$$

$$P \rightarrow P \cdot A \mid P / A \mid A$$

$$A \rightarrow (S) \mid z$$

Betrachte:  $z \cdot z - z \cdot (z + z)$



Welchen Vorteil hat diese Grammatik?

## Beispiel 3

$$S \rightarrow P + S \mid P - S \mid P$$

$$P \rightarrow A \cdot P \mid A/P \mid A$$

$$A \rightarrow (S) \mid z$$

Betrachte:  $z \cdot z - z \cdot (z + z)$

Welchen Vorteil hat diese Grammatik?

```
#include <stdlib.h>
#include <stdio.h>
```

```
char * s;
void A(), P(), Z();
```

```
void S() {
    P();
    if(*s == '+') s++, S();
    if(*s == '-') s++, S();
}
```

```
void P() {
    A();
    if(*s == '*') s++, P();
    if(*s == '/') s++, P();
}
```

```
void A() {
    if(*s == '(') {
        ++s;
        S();
    }
    if(*s == ')') ++s;
    else exit(1);
}
else Z();
}
```

```
int main(int argc, char * argv[])
{
    if(argc != 2) exit(1);
    s = argv[1];
    S();
    if(*s != 0) exit(1);
    printf("okay!\n");
    return 0;
}
```