

## Beispiel (Anwendung von Satz 95)

### Korollar 96

*Die Sprache*

$$L = \{0^i 1^j 2^k; i = j \text{ oder } j = k\} \subset \{0, 1, 2\}^*$$

*ist kontextfrei, aber nicht deterministisch kontextfrei ( $\in \text{CFL} \setminus \text{DCFL}$ ).*

**Beweis:**

*L wird erzeugt z.B. von der CFG mit den Produktionen*

$$S \rightarrow A \mid AB \mid B \mid C \mid CD \mid D \mid \epsilon$$

$$A \rightarrow 01 \mid 0A1$$

$$C \rightarrow 0 \mid 0C$$

$$B \rightarrow 2 \mid 2B$$

$$D \rightarrow 12 \mid 1D2$$

## Beispiel (Anwendung von Satz 95)

### Korollar 96

*Die Sprache*

$$L = \{0^i 1^j 2^k; i = j \text{ oder } j = k\} \subset \{0, 1, 2\}^*$$

*ist kontextfrei, aber nicht deterministisch kontextfrei ( $\in \text{CFL} \setminus \text{DCFL}$ ).*

**Beweis:**

*Wäre  $L \in \text{DCFL}$ , dann auch*

$$L' := \bar{L} \cap 0^* 1^* 2^* = \{0^i 1^j 2^k; i \neq j \text{ und } j \neq k\}.$$

*Mit Hilfe von Ogden's Lemma sieht man aber leicht, dass dies nicht der Fall ist.*

## 4.10 $LR(k)$ -Grammatiken

### Beispiel 97 (Grammatik für Arithmetische Ausdrücke)

Regeln:

$$S \rightarrow A$$

$$A \rightarrow E \mid A + E \mid A - E$$

$$E \rightarrow P \mid E * P \mid E / P$$

$$P \rightarrow (A) \mid a$$

Wir betrachten für das Parsen einen **bottom-up**-Ansatz, wobei die Reduktionen von links nach rechts angewendet werden.

## Beispiel 97 (Grammatik für Arithmetische Ausdrücke)

Dabei können sich bei naivem Vorgehen allerdings Sackgassen ergeben, die dann aufgrund des Backtracking zu einem ineffizienten Algorithmus führen:

Regeln:

$$S \rightarrow A$$

$$A \rightarrow E \mid A + E \mid A - E$$

$$E \rightarrow P \mid E * P \mid E / P$$

$$P \rightarrow (A) \mid a$$

Ableitung:

$$a \quad + \quad a \quad * \quad a$$

$$P \quad + \quad a \quad * \quad a$$

$$E \quad + \quad a \quad * \quad a$$

$$A \quad + \quad a \quad * \quad a$$

$$S \quad + \quad a \quad * \quad a$$

Sackgasse!

## Beispiel 97 (Grammatik für Arithmetische Ausdrücke)

... oder auch

Regeln:

$$S \rightarrow A$$

$$A \rightarrow E \mid A + E \mid A - E$$

$$E \rightarrow P \mid E * P \mid E / P$$

$$P \rightarrow (A) \mid a$$

Ableitung:

$$a \quad + \quad a \quad * \quad a$$

$$P \quad + \quad a \quad * \quad a$$

$$E \quad + \quad a \quad * \quad a$$

$$A \quad + \quad a \quad * \quad a$$

$$A \quad + \quad P \quad * \quad a$$

$$A \quad + \quad E \quad * \quad a$$

$$A \quad * \quad a$$

$$A \quad * \quad P$$

$$A \quad * \quad E$$

Sackgasse!

## Beispiel 97 (Grammatik für Arithmetische Ausdrücke)

Zur Behebung des Problems führen wir für jede Ableitungsregel (besser hier: **Reduktionsregel**) einen **Lookahead** (der Länge  $k$ ) ein (in unserem Beispiel  $k = 1$ ) und legen fest, dass eine Ableitungsregel nur dann angewendet werden darf, wenn die nächsten  $k$  Zeichen mit den erlaubten Lookaheads übereinstimmen.

## Beispiel 97 (Grammatik für Arithmetische Ausdrücke)

Produktion	Lookaheads (der Länge 1)
$S \rightarrow A$	$\epsilon$
$A \rightarrow E$	$+, -, ), \epsilon$
$A \rightarrow A + E$	$+, -, ), \epsilon$
$A \rightarrow A - E$	$+, -, ), \epsilon$
$E \rightarrow P$	beliebig
$E \rightarrow E * P$	beliebig
$E \rightarrow E / P$	beliebig
$P \rightarrow (A)$	beliebig
$P \rightarrow a$	beliebig

## Beispiel 97 (Grammatik für Arithmetische Ausdrücke)

Damit ergibt sich

Produktion	Lookaheads
$S \rightarrow A$	$\epsilon$
$A \rightarrow E$	$+, -, ), \epsilon$
$A \rightarrow A + E$	$+, -, ), \epsilon$
$A \rightarrow A - E$	$+, -, ), \epsilon$
$E \rightarrow P$	beliebig
$E \rightarrow E * P$	beliebig
$E \rightarrow E / P$	beliebig
$P \rightarrow (A)$	beliebig
$P \rightarrow a$	beliebig

Ableitung:

$$\begin{array}{cccccc}
 a & + & a & * & a \\
 P & + & a & * & a \\
 E & + & a & * & a \\
 A & + & a & * & a \\
 A & + & P & * & a \\
 A & + & E & * & a \\
 A & + & E & * & P \\
 A & + & & E & \\
 & & A & & \\
 & & S & & 
 \end{array}$$



## Definition 98

Eine kontextfreie Grammatik ist eine  $LR(k)$ -Grammatik, wenn man durch Lookaheads der Länge  $k$  erreichen kann, dass bei einer Reduktion von links nach rechts in jedem Schritt höchstens eine Produktion/Reduktion anwendbar ist.

## Korollar 99

*Jede kontextfreie Sprache, für die es eine  $LR(k)$ -Grammatik gibt, ist deterministisch kontextfrei.*

**Bemerkung:**

Es gibt eine (im allgemeinen nicht effiziente) Konstruktion, um aus einer  $LR(k)$ -Grammatik,  $k > 1$ , eine äquivalente  $LR(1)$ -Grammatik zu machen.

**Korollar 100**

*Die folgenden Klassen von Sprachen sind gleich:*

- *die Klasse DCFL,*
- *die Klasse der  $LR(1)$ -Sprachen.*

## 4.11 $LL(k)$ -Grammatiken

### Beispiel 101 (Noch eine Grammatik)

Regeln:

$$S \rightarrow A$$

$$A \rightarrow E \mid E + A$$

$$E \rightarrow P \mid P * E$$

$$P \rightarrow (A) \mid a$$

$$S \rightarrow A$$

$$A \rightarrow EA'$$

$$A' \rightarrow +A \mid \epsilon$$

$$E \rightarrow PE'$$

$$E' \rightarrow *E \mid \epsilon$$

$$P \rightarrow (A) \mid a$$

Wir betrachten nun für das **Parsen** einen **top-down**-Ansatz, wobei die Produktionen in Form einer Linksableitung angewendet werden.

## Beispiel 101 (Noch eine Grammatik)

Regeln:

$$S \rightarrow A$$

$$A \rightarrow E \mid E + A$$

$$E \rightarrow P \mid P * E$$

$$P \rightarrow (A) \mid a$$

Ableitung:

$S$

$A$

$E$

$P$

$a \quad + \quad a \quad * \quad a$

## Beispiel 101 (Noch eine Grammatik)

Wir bestimmen nun für jede Produktion  $A \rightarrow \alpha$  ihre **Auswahlmenge**, das heißt die Menge aller terminalen Präfixe der Länge  $\leq k$  der von  $\alpha$  ableitbaren Zeichenreihen. Es ergibt sich (der Einfachheit lassen wir  $\epsilon$ -Produktionen zu):

$S$	$\rightarrow A$	$\{a, ()$
$A$	$\rightarrow EA'$	$\{a, ()$
$A'$	$\rightarrow +A$	$\{+\}$
$A'$	$\rightarrow \epsilon$	$\{), \epsilon\}$
$E$	$\rightarrow PE'$	$\{a, ()$
$E'$	$\rightarrow *E$	$\{*\}$
$E'$	$\rightarrow \epsilon$	$\{+, ), \epsilon\}$
$P$	$\rightarrow (A)$	$\{($
$P$	$\rightarrow a$	$\{a\}$

## Beispiel 101 (Noch eine Grammatik)

Damit ergibt sich

$S$	$\rightarrow A$	$\{a, (\}$
$A$	$\rightarrow EA'$	$\{a, (\}$
$A'$	$\rightarrow +A$	$\{+\}$
$A'$	$\rightarrow \epsilon$	$\{), \epsilon\}$
$E$	$\rightarrow PE'$	$\{a, (\}$
$E'$	$\rightarrow *E$	$\{*\}$
$E'$	$\rightarrow \epsilon$	$\{+, ), \epsilon\}$
$P$	$\rightarrow (A)$	$\{(\}$
$P$	$\rightarrow a$	$\{a\}$

Ableitung:

$$\begin{aligned} &S \\ &EA' \\ &aE'A' \\ &\vdots \\ &a + PE'A' \\ &\vdots \\ &a + a * PE'A' \\ &\vdots \\ &a + a * a \end{aligned}$$

## Definition 102

Eine kontextfreie Grammatik ist eine  $LL(k)$ -Grammatik, wenn man durch Lookaheads der Länge  $k$  erreichen kann, dass bei einer top-down Linksableitung in jedem Schritt höchstens eine Produktion anwendbar ist. Eine Sprache ist  $LL(k)$ , wenn es dafür eine  $LL(k)$ -Grammatik gibt.

## Korollar 103

*Jede kontextfreie Sprache, für die es eine  $LL(k)$ -Grammatik gibt, ist deterministisch kontextfrei.*

## Bemerkung

Man kann zeigen, dass die Klasse der  $LL(k)$ -Sprachen eine echte Teilklasse der Klasse der  $LL(k+1)$ -Sprachen ist.

## Bemerkungen:

- 1 Parser für  $LL(k)$ -Grammatiken entsprechen der Methode des **rekursiven Abstiegs** (**recursive descent**).
- 2  $LL(k)$  ist eine strikte Teilklasse von  $LR(k)$ .
- 3 Es gibt  $L \in \text{DCFL}$ , so dass  $L \notin LL(k)$  für alle  $k$ .