

# Kontextsensitive Grammatiken

# „Adventskalender“

Typ	Name	Erlaubte Produktionen	Akzeptierende Maschine	Beispiel
3	Regulär	$N \rightarrow wM$ $w \in T^*$	Endlicher Automat	$a^n$
2	Kontextfrei	$N \rightarrow w$ $w \in (N \cup T)^*$	Kellerautomat	$a^n b^n$
1	Kontext-sensitiv	$uNv \rightarrow uvw$ $u, v, w \in (N \cup T)^*$ $S \rightarrow \text{eps}$	Linear gebundener Automat	$a^n b^n c^n$

Skript Worsch: Seite 57

# Eine Java Klasse

---

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        i=0;  
        while (i<5) {  
            System.out.println("Hello World!");  
        }  
    }  
}
```

**Semantisch Korrekt?**

# Eine Java Klasse

---

```
public class HelloWorld {
```

```
    static int i;
```

```
    public static void main(String[] args) {
```

```
        i=0;
```

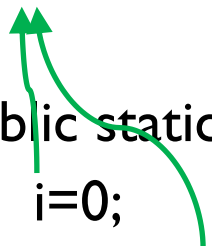
```
        while (i<5) {
```

```
            System.out.println("Hello World!");
```

```
        }
```

```
    }
```

```
}
```



# Eine Java Klasse

---

- ▶ Wenn ich in Java eine Variable **verwenden** möchte, dann muss ich diese vorher **deklarieren**.
- ▶ Diese Deklaration muss nicht unmittelbar vorher erfolgen, sondern irgendwo im **Kontext** der Verwendung.

# Variablendeklaration und Grammatiken

---

- ▶ Grundidee:

Die Produktion  $X \rightarrow w$  zuzulassen, wenn  $X$  vorher deklariert wurde

- ▶ Formalisierung der Idee

Zufügen des vorherigen Kontexts  $u$  zur Produktion

$$uX \rightarrow uw$$

Festlegen dass im Kontext  $u$  die Deklaration von enthalten  $X$  sein muss

$$u \rightarrow \text{irgendwas Deklaration } X \text{ irgendwas}$$

# Kontextsensitive Grammatiken

---

- ▶ Formal darf der Kontext vorne und hinten stehen  
wir erlauben Produktionen der Form  
 $uXv \rightarrow uwv$ ,
- ▶ Umgangssprachlich  
links und rechts des „Geschehens“ wird Kontext  $u$  und  $v$   
gefordert

# Definition 5.1: Typ-1-Grammatik

---

Eine **kontextsensitive Grammatik** oder **Typ-1-Grammatik (TIG)** ist eine Grammatik  $G = (N, T, S, P)$ , bei der alle Produktionen von der folgenden Form sind:

- ▶  $uXv \rightarrow uwv$  mit  $u, v \in (N \cup T)^*$ ,  $w \in (N \cup T)^+$ ,  $X \in N$   
oder
- ▶  $S \rightarrow \varepsilon$ . Falls diese Produktion existiert, kommt aber  $S$  in keiner Produktion auf der rechten Seite vor.

Eine formale Sprache ist vom **Typ I** oder **kontextsensitiv**, wenn es eine Typ-1-Grammatik gibt, die sie erzeugt.



# Beispiel

---

- ▶ Die formale Sprache  $L = \{0^k 1^k 2^k \mid k \in \mathbb{N}\}$  ist nicht kontextfrei
- ▶ Gesucht ist eine kontextsensitive Grammatik, die  $L = \{0^k 1^k 2^k \mid k \in \mathbb{N}\}$  erzeugt.

# Grammatik für $L = \{0^k 1^k 2^k \mid k \in \mathbb{N}\}$

---

- ▶ Produktion von gleich vielen Terminalen 0,1,2 über Nichtterminale  $A, B, C$ :

$$S \rightarrow ASBC \mid ABC$$

- ▶ Erzeugt so etwas wie  $ABC, AABCBC, AAABCBCBC \dots$

- ▶ Ansatz zum Vertauschen der  $B$ s und  $C$ s:

$$CB \rightarrow BC$$

- ▶ Direkt ist das in einer kontextsensitiven Grammatik nicht erlaubt

# Grammatik für $L = \{0^k 1^k 2^k \mid k \in \mathbb{N}\}$

---

- ▶ Ersetzen von  $CB \rightarrow BC$ ?

$$\varepsilon CB \rightarrow \varepsilon HB$$

$$HB\varepsilon \rightarrow HC\varepsilon$$

$$\varepsilon HC \rightarrow \varepsilon BC$$

- ▶ Beispiel:

$$B\textcolor{green}{C}BC \Rightarrow BHBC \Rightarrow BHCC \Rightarrow BBCC$$

- ▶ Produktionen:  $CB \rightarrow HB, HB \rightarrow HC, HC \rightarrow BC$

# Allgemein für $XY \rightarrow YX$

---

## ► Ersetzen von $XY \rightarrow YX$ ?

$XY \rightarrow HY$

$HY \rightarrow HX$

$HX \rightarrow YX$

BSP: S - ASX - AAXX - aaXX - aaBCBC - aabCBC - aabHCC - aabBCC - aabbCC - aabbC - aabbcc (aabbcc ist das Ziel, das wir produzieren wollen)

# Grammatik für $L = \{0^k 1^k 2^k \mid k \in \mathbb{N}\}$

---

- ▶ Jetzt können wir so etwas wie  $ABC, AAB BCC, AAABBBCCC, \dots$  erzeugen.
- ▶ Fehlt noch das Wandeln in Terminale:  
 $A \rightarrow 0, B \rightarrow 1, C \rightarrow 2$  ?
- ▶ Achtung:  
 $\underline{S} \Rightarrow A\underline{S}ABC \Rightarrow AABCBC \Rightarrow^* 001212 \notin L$
- ▶ Ein  $B$  darf erst gewandelt werden, wenn vornedran nur Nullen oder Einsen stehen.

# Grammatik für $L = \{0^k 1^k 2^k \mid k \in \mathbb{N}\}$

---

- ▶ Ein  $B$  darf erst gewandelt werden, wenn vornedran nur Nullen und Einsen stehen.
- ▶ Die  $A$ s stehen „schon immer“ richtig:  
 $A \rightarrow 0$
- ▶ Die  $B$ s nur, wenn sie auf 0 oder 1 folgen:  
 $0B \rightarrow 01$   
 $1B \rightarrow 11$
- ▶ Wenn sich alle  $B$ s wandeln lassen, stehen auch die  $C$ s richtig:  
 $C \rightarrow 2$

# Grammatik für $L = \{0^k 1^k 2^k \mid k \in \mathbb{N}\}$

---

$G = \{\{S, A, B, C, H, I\}, \{0, 1, 2\}, S, P\}$  mit

$$P = \left\{ \begin{array}{l} S \rightarrow ASBC \mid ABC, \\ CB \rightarrow HB, HB \rightarrow HC, HC \rightarrow BC, \\ A \rightarrow 0, 0B \rightarrow 01, 1B \rightarrow 11, C \rightarrow 2 \end{array} \right\}$$

# Automat für kontextsensitive Grammatiken

---

## Linear beschränkter Automat

- ▶ Turingmaschine
- ▶ Bandlänge begrenzt auf Länge des zu erkennenden Wortes + Anfangsmarker + Endmarker



# Längenbegrenzung bei der Analyse

---

- ▶ Produktionen der kontextsensitiven Grammatiken sind nicht-verkürzend (Ausnahme  $S \rightarrow \varepsilon$ )
- ▶ Durch „Ausprobieren“ kann man mit einer endlichen Zahl von Schritten entscheiden ob ein Wort aus der Grammatik erzeugt werden kann (ähnlich wie bei der Chomsky Normalform)