Kontextsensitive Grammatiken

„Adventskalender“

Typ	Name	Erlaubte Produktionen	Akzeptierende Maschine	Beispiel
3	Regulär	N → wM $w \in T^*$	Endlicher Automat	a^n
2	Kontextfrei	N → w $w \in (N \cup T)^*$	Kellerautomat	$a^n b^n$
1	Kontext-sensitiv	uNv → uwv $\mathrm{u}, \mathrm{v}, w \in (N \cup T)^*$ S → 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 das im Kontext u die Deklaration von enthalten X sein muss
  $u \rightarrow irgendwas\ DeklarationX\ irgendwas$

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 (T1G) ist eine Grammatik $G = (N, T, S, P)$, bei der alle Produktionen von der folgenden Form sind:

auch erlaubt: eps A eps -> eps BC eps

- $uXv \rightarrow uwv$ 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 1 oder kontextsensitiv, wenn es eine Typ-1-Grammatik gibt, die sie erzeugt.

Beispiel

- Die formale Sprache $L = \{0^k 1^k 2^k | k \in \mathbb{N}\}$ ist nicht kontextfrei

- Gesucht ist eine kontextsensitive Grammatik, die $L = \{0^k 1^k 2^k | k \in \mathbb{N}\}$ erzeugt.

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

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

$$S \rightarrow ASBC \mid ABC$$

- Erzeugt so etwas wie $ABC, AABCBC, AAABCBCBC \ldots$

- Ansatz zum Vertauschen der Bs und Cs:

$$CB \rightarrow BC$$

- Direkt ist das in einer kontextsensitiven Grammatik nicht erlaubt

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

- Ersetzen von $CB \rightarrow BC$?

$$\varepsilon CB \rightarrow \varepsilon HB$$
$$\mathrm{H}B\varepsilon \rightarrow HC\varepsilon$$
$$\varepsilon HC \rightarrow \varepsilon BC$$

- Beispiel:
  $BCBC \Rightarrow BHBC \Rightarrow BHCC \Rightarrow BBCC$

- 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 - aabbcC - aabbcc (aabbcc ist das Ziel, das wir produzieren wollen)

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

- Jetzt können wir so etwas wie
  $ABC, AABBCC, AAABBBCCC, \ldots$
  erzeugen.

- Fehlt noch das Wandeln in Terminale:
  $A \to 0, B \to 1, C \to 2$?

- 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 $L = \{0^k 1^k 2^k | k \in \mathbb{N}\}$

- Ein B darf erst gewandelt werden, wenn vornedran nur Nullen und Einsen stehen.

- Die As stehen „schon immer“ richtig:
  $$A \to 0$$

- Die Bs nur, wenn sie auf 0 oder 1 folgen:
  $$0B \to 01$$
  $$1B \to 11$$

- Wenn sich alle Bs wandeln lassen, stehen auch die Cs richtig:
  $$C \to 2$$

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

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

Rückwärts analysieren => bei gesuchtem Wort anfangen
und von rechts nach links die Regeln anwenden bis S Startsymbol

monotone Grammatik: von links nach rechts Wort wird immer eins länger oder bleibt gleich
Erkennbar mit Turing Maschine, die ein Band hat mit Anfang-/Endmarker und der Länge des gesuchten Wortes; bei Rückwärtsanwendung der Produktionen wird das Wort entweder eins kürzer oder bleibt gleich => passt immer auf das Band mit der Länge des gesuchten Wortes
SIEHE NÄCHSTE 2 SEITEN

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 $S \to \varepsilon$)

- 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)