TECHNISCHE UNIVERSITÄT DRESDEN

FORMALE SYSTEME

2. Vorlesung: Grammatiken und die Chomsky-Hierarchie

Markus Krötzsch

TU Dresden, 13. Oktober 2016

Wiederholung

- Formale Sprachen sind in Praxis und Theorie sehr wichtig
- Ein Alphabet ist eine nichtleere, endliche Menge von Symbolen
- Ein Wort ist eine endliche Sequenz von Symbolen
- Eine Sprache ist eine Menge von Wörtern
- Es gibt viele Operationen auf Sprachen ($\cap, \cup, \overline{\ }, \circ, {}^*, {}^+, \setminus$)
- Man kann Sprachen auf viele Arten beschreiben
  (von denen wir noch einige genauer kennen lernen werden)

Sprachen beschreiben

Wie kann man Sprachen beschreiben?

Die Operationen aus der vorigen Vorlesung können das schon ganz gut:

Beispiel (Wiederholung): Die Menge aller gültigen Schreibweisen für Dezimalzahlen kann wie folgt definiert werden:

$$\textbf{Dezimalzahl} = \left(\{\epsilon\} \cup \{+, -\}\right) \circ \left(\{0\} \cup (\mathbf{Z} \setminus \{0\}) \circ \mathbf{Z}^*\right) \circ \left(\{\epsilon\} \cup \{.\}\mathbf{Z}^+\right)$$

mit $\mathbf{Z} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Wir haben hier eine unendliche Sprache mit einfachen Grundoperationen aus endlichen Sprachen konstruiert.

Funktioniert das für alle Sprachen?

Wie viele Wörter gibt es?

Satz: Es gibt abzählbar viele Wörter über jedem Alphabet Σ. Jede Sprache ist also entweder endlich oder abzählbar unendlich.

Idee: Zum „Abzählen“ reihen wir die Wörter in Σ^* der Länge nach auf, beginnend mit den kurzen Wörtern.
Zum Beispiel: $\epsilon, \mathtt{a}, \mathtt{b}, \mathtt{aa}, \mathtt{ab}, \mathtt{ba}, \mathtt{bb}, \mathtt{aaa}, \mathtt{aab}, \ldots$

Beweis (die gleiche Idee, aber etwas formaler):

- Laut Definition ist $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots = \bigcup_{i \geq 0} \Sigma^i$.
- Jede der Mengen Σ^i ist endlich (es gibt nur endlich viele Wörter der Länge i).
- Es gibt abzählbar viele Mengen Σ^i (eine für jede natürliche Zahl)
- Eine abzählbare Vereinigung endlicher Mengen ist abzählbar.
- Also ist Σ^* abzählbar. □

Wie viele Sprachen gibt es?

Satz: Es gibt überabzählbar viele Sprachen über jedem beliebigen Alphabet Σ.

Beweis:

- Die Menge aller Wörter Σ^* ist abzählbar unendlich, selbst wenn Σ nur ein Symbol enthält.
- Eine Sprache ist eine beliebige Teilmenge von Σ^*.
- Die Menge aller Sprachen über Σ ist die Menge aller Teilmengen (Potenzmenge) von Σ^*.
- Die Kardinalität der Potenzmenge ist immer größer als die der Menge selbst (Satz von Cantor).
- Also gibt es überabzählbar viele Sprachen. □

Aus Kardinalitäten lernen

Zusammenfassung: Über jedem beliebigen Alphabet Σ gibt es

- abzählbar viele Wörter
- überabzählbar viele Sprachen

Beobachtung: Wir beschreiben Sprachen mit „Wörtern“:

- Natürlichsprachliche Texte sind Wörter
- Computerprogramme sind Wörter
- Mathematische Formeln sind Wörter

Es gibt viel mehr Sprachen als Texte, Programme oder Formeln!

Satz: Fast alle Sprachen können in keiner Weise endlich beschrieben werden. Fast alle Sprachen werden durch kein Computerprogramm korrekt erkannt.

Ein starker Satz, aber er gibt uns keine Hinweise, welche Sprachen man wie beschreiben kann. Zum Beispiel gibt es eindeutig beschreibbare Sprachen, für die es kein Computerprogramm gibt.

Grammatiken

Sprachen mit Grammatiken beschreiben

Grammatiken sind die wichtigste Methode zur Beschreibung von Sprachen.

Sie verwenden Ersetzungsregeln um Wörter zu erzeugen.

Beispiel: Die folgende Grammatik erzeugt Dezimalzahlen.

$$\begin{aligned}
\text{Dezimalzahl} &\to \text{Vorzeichen DZahl} \mid \text{DZahl} && (1)\\
\text{Vorzeichen} &\to + \mid - && (2)\\
\text{DZahl} &\to \text{NatZahl} \mid \text{NatZahl . Ziffernreihe} && (3)\\
\text{NatZahl} &\to \text{Ziffer} \mid \text{ZifferAb1 Ziffernreihe} && (4)\\
\text{Ziffernreihe} &\to \text{Ziffer} \mid \text{Ziffer Ziffernreihe} && (5)\\
\text{ZifferAb1} &\to 1 \mid 2 \mid \ldots \mid 9 && (6)\\
\text{Ziffer} &\to 0 \mid 1 \mid 2 \mid \ldots \mid 9 && (7)
\end{aligned}$$

Grammatiken anwenden

$$\begin{aligned}
\text{Dezimalzahl} &\to \text{Vorzeichen DZahl} \mid \text{DZahl} && (1)\\
\text{Vorzeichen} &\to + \mid - && (2)\\
\text{DZahl} &\to \text{NatZahl} \mid \text{NatZahl . Ziffernreihe} && (3)\\
\text{NatZahl} &\to \text{Ziffer} \mid \text{ZifferAb1 Ziffernreihe} && (4)\\
\text{Ziffernreihe} &\to \text{Ziffer} \mid \text{Ziffer Ziffernreihe} && (5)\\
\text{ZifferAb1} &\to 1 \mid 2 \mid \ldots \mid 9 && (6)\\
\text{Ziffer} &\to 0 \mid 1 \mid 2 \mid \ldots \mid 9 && (7)
\end{aligned}$$

Wir können folgende Ersetzungsschritte ausführen:

$$\begin{aligned}
\text{Dezimalzahl} &\overset{(1)}{\Rightarrow} \text{Vorzeichen DZahl} \overset{(2)}{\Rightarrow} -\text{DZahl} \overset{(3)}{\Rightarrow} -\text{NatZahl.Ziffernreihe}\\
&\overset{(4)}{\Rightarrow} -\text{Ziffer.Ziffernreihe} \overset{(7)}{\Rightarrow} -3.\text{Ziffernreihe}\\
&\overset{(5)}{\Rightarrow} -3.\text{Ziffer Ziffernreihe} \overset{(5)}{\Rightarrow} -3.\text{Ziffer Ziffer}\\
&\overset{(7)}{\Rightarrow} -3.1\ \text{Ziffer} \overset{(7)}{\Rightarrow} -3.14
\end{aligned}$$

Definition: Grammatik

Eine Grammatik G ist ein 4-Tupel $G = \langle V, \Sigma, P, S\rangle$ bestehend aus:

- V eine Menge von Variablennamen
- Σ ein Alphabet, disjunkt mit V (d.h., $\Sigma \cap V = \emptyset$)
- P eine Menge von Produktionsregeln der Form $w \to v$ für beliebige Wörter w und v über $\Sigma \cup V$, wobei w mindestens eine Variable enthält (d.h. $w, v \in (\Sigma \cup V)^*$ und $w \notin \Sigma^*$)
- S eine Startvariable aus V (d.h. $S \in V$)

Die Elemente von Σ nennt man auch Terminalsymbole, die Elemente von V analog Nicht-Terminalsymbole.

Die zuvor verwendete Schreibweise $w \to v_1 \mid v_2 \mid \ldots \mid v_n$ ist eine Abkürzung für die Produktionsregeln $w \to v_1, w \to v_2, \ldots, w \to v_n$.

Beispiele: Grammatiken

Beispiel: Eine Grammatik $\langle V, \Sigma, P, \text{Start}\rangle$ sei gegeben durch $V = \{\text{Start}, \text{Summe}, \text{Var}\}$, $\Sigma = \{(,), +, *, \mathtt{x}, \mathtt{y}, \mathtt{z}\}$ und Regeln P:

$$\begin{aligned}
\text{Start} &\to \text{Summe} \mid \text{Summe} * \text{Start}\\
\text{Summe} &\to (\ \text{Var} + \text{Var}\ )\\
\text{Var} &\to \mathtt{x} \mid \mathtt{y} \mid \mathtt{z}.
\end{aligned}$$

Diese Grammatik erzeugt zum Beispiel das Wort $(\mathtt{x} + \mathtt{y}) * (\mathtt{x} + \mathtt{z})$.

Beispiel: Die selbe Sprache wird auch durch die Grammatik $\langle\{\text{Start}, \text{Var}\}, \Sigma, P', \text{Start}\rangle$ definiert, mit Σ wie oben und Regeln P':

$$\begin{aligned}
\text{Start} &\to (\text{Var} + \text{Var})\\
(\text{Var} + \text{Var}) &\to (\text{Var} + \text{Var}) * (\text{Var} + \text{Var})\\
\text{Var} &\to \mathtt{x} \mid \mathtt{y} \mid \mathtt{z}.
\end{aligned}$$

Es ist also erlaubt (1) Terminalsymbole durch Regeln zu verändern und (2) mehr als ein Symbol auf einmal zu ersetzen.

Beispiele: Diskussion

Die Beispiele werfen wichtige Fragen auf:

- Wie kann man herausfinden, ob zwei unterschiedliche Grammatiken die selbe Sprache beschreiben?
- Muss man Grammatiken so allgemein definieren oder genügen auch vereinfachte Formen?
- Kann man Grammatiken (automatisch) vereinfachen, ohne die Sprache zu verändern?

Antworten in dieser Vorlesung ...

Erst einmal sollten wir die „Sprache einer Grammatik“ ordentlich definieren.

Definition: Ableitung

Sei $\langle V, \Sigma, P, S \rangle$ eine Grammatik. Die 1-Schritt-Ableitungsrelation ist eine binäre Relation $\Rightarrow$ zwischen Wörtern aus $(V \cup \Sigma)^*$, so dass $u \Rightarrow v$ genau dann wenn:

$$u = w_1\, x\, w_2 \text{ und } v = w_1\, y\, w_2 \text{ und es gibt eine Regel } x \to y \in P$$

wobei $w_1, w_2, x, y \in (V \cup \Sigma)^*$ beliebige Wörter sind.

Die Ableitungsrelation $\Rightarrow^*$ ist der reflexive, transitive Abschluss von $\Rightarrow$, das heißt $u \Rightarrow^* v$ genau dann wenn:

$$u = w_1 \Rightarrow w_2 \Rightarrow \ldots \Rightarrow w_{n-1} \Rightarrow w_n = v$$

wobei $n \geq 1$ und $w_1, \ldots, w_n \in (V \cup \Sigma)^*$ beliebige Wörter sind. Insbesondere gilt $u \Rightarrow^* u$ für alle $u \in (V \cup \Sigma)^*$ (Fall $n = 1$).

Anmerkung: Der Begriff „Herleitungsrelation“ ist auch gebräuchlich. Wir verwenden „Ableitung“ und „Herleitung“ synonym.

Anmerkung 2: Manche Autoren schreiben $\vdash$ statt $\Rightarrow$.

Definition: Sprache einer Grammatik

Die von einer Grammatik $G = \langle V, \Sigma, P, S \rangle$ erzeugte Sprache $\mathbf{L}(G)$ besteht aus allen Wörtern über Σ, die man von S ableiten kann:

$$\mathbf{L}(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$$

Vereinfachung: Für einfache Beispiele nehmen wir ab jetzt an, dass alle Großbuchstaben Variablen und alle Kleinbuchstaben Terminalsymbole sind, wobei S die Startvariable ist.

Beispiel: Wir betrachten die Grammatik G_{id} mit den Regeln:

S → BA A → BA | ZA | ϵ B → b Z → z

(Vereinfachte Version der „Bezeichner“, wobei alle Buchstaben als b und alle Ziffern als z abgekürzt werden.)

Dann gilt $\mathbf{L}(G_{\text{id}}) = \{\text{b}\} \circ \{\text{b}, \text{z}\}^*$.

Beweis zum Beispiel (1)

Für die Grammatik G_{id} mit den Regeln

S → BA A → BA | ZA | ϵ B → b Z → z

gilt $\mathbf{L}(G_{\text{id}}) = \{\text{b}\} \circ \{\text{b}, \text{z}\}^*$?
Das müsste man erst einmal beweisen ...

Beweis: Wir zeigen die beiden Richtungen
(1) $\mathbf{L}(G_{\text{id}}) \subseteq \{\text{b}\} \circ \{\text{b}, \text{z}\}^*$ und (2) $\mathbf{L}(G_{\text{id}}) \supseteq \{\text{b}\} \circ \{\text{b}, \text{z}\}^*$ getrennt.

(1) Die Behauptung ist: „jedes Wort in $\mathbf{L}(G_{\text{id}})$ ist in $\{\text{b}\} \circ \{\text{b}, \text{z}\}^*$.“

- $\mathbf{L}(G_{\text{id}}) \subseteq \{\text{b}, \text{z}\}^*$, weil $\{\text{b}, \text{z}\}$ das Alphabet von G_{id} ist
- jedes Wort in $\mathbf{L}(G_{\text{id}})$ beginnt mit b, weil S nur durch BA ersetzt werden kann und B nur durch b
- also ist $\mathbf{L}(G_{\text{id}}) \subseteq \{\text{b}\} \circ \{\text{b}, \text{z}\}^*$

Beweis zum Beispiel (2)

S → BA A → BA | ZA | ϵ B → b Z → z

Beweis (Fortsetzung): Es fehlt noch die Rückrichtung.

(2) Die Behauptung ist: „jedes Wort in $\{b\} \circ \{b, z\}^*$ ist in $\mathbf{L}(G_{id})$."

- Sei $bs_1 \dots s_n$ mit $n \geq 0$ ein beliebiges Wort in $\{b\} \circ \{b, z\}^*$ (also $s_1, \dots, s_n \in \{b, z\}$)
- Wir zeigen $bs_1 \dots s_n \in \mathbf{L}(G_{id})$ indem wir eine Ableitung angeben, d.h. wir zeigen $S \Rightarrow^* bs_1 \dots s_n$
- Die Ableitung beginnt immer so: $S \Rightarrow BA \Rightarrow bA$
- Jetzt müssen wir nur noch $A \Rightarrow^* s_1 \dots s_n$ zeigen.

Beweis zum Beispiel (3)

S → BA A → BA | ZA | ϵ B → b Z → z

Beweis (Fortsetzung): Wir behaupten $A \Rightarrow^* s_1 \dots s_n$ für beliebige $n \geq 0$ und $s_1, \dots, s_n \in \{b, z\}$. Die Ableitung ist in jedem konkreten Fall leicht gefunden, zum Beispiel für $n = 2$ und $s_1 s_2 = zb$ ergibt sich $A \Rightarrow ZA \Rightarrow zA \Rightarrow zBA \Rightarrow zbA \Rightarrow zb$.

Aber es gibt unendlich viele Fälle!
$\rightsquigarrow$ Wir zeigen, wie man die Ableitung rekursiv konstruieren kann.

- Für $n = 0$ ist $s_1 \dots s_n = \epsilon$ und $A \Rightarrow \epsilon$ gilt direkt
- Für $n > 0$ und $s_1 = b$ verwenden wir $A \Rightarrow BA \Rightarrow bA$ und von da ab (rekursiv) die Ableitung $A \Rightarrow^* s_2 \dots s_n$.
- Für $n > 0$ und $s_1 = z$ verwenden wir $A \Rightarrow ZA \Rightarrow zA$ und von da ab (rekursiv) die Ableitung $A \Rightarrow^* s_2 \dots s_n$.

Die Rekursion terminiert, da wir mit jedem Rekursionschritt kürzere Wörter ableiten.

Beweis zum Beispiel (4)

S → BA A → BA | ZA | ϵ B → b Z → z

Beweis (Fortsetzung): Der „rekursiv konstruierte Beweis" ist natürlich eine vollständige Induktion:

Induktionsbehauptung: $A \Rightarrow^* s_1 \dots s_n$ für alle $n \geq 0$ und $s_1, \dots, s_n \in \{b, z\}$
Induktionsanfang: für $n = 0$ gilt $A \Rightarrow^* \epsilon$
Induktionshypothese: $A \Rightarrow^* s_1 \dots s_n$ gilt für n und alle $s_1, \dots, s_n \in \{b, z\}$
Induktionschritt: $A \Rightarrow^* s_1 \dots s_{n+1}$ gilt für alle $s_1, \dots, s_n \in \{b, z\}$, weil:

- wenn $s_1 = b$ dann $A \Rightarrow BA \Rightarrow bA$
- wenn $s_1 = z$ dann $A \Rightarrow ZA \Rightarrow zA$
- Behauptung folgt, da $A \Rightarrow^* s_2 \dots s_{n+1}$ (Induktionshypothese)

Zusammenfassung Beweis Teil (2):
Für beliebige Wörter w gilt: wenn $w \in \{b\} \circ \{b, z\}^*$ dann $w \in \mathbf{L}(G_{id})$.
Anders gesagt: $\{b\} \circ \{b, z\}^* \subseteq \mathbf{L}(G_{id})$. □

Die Chomsky-Hierarchie

Sprachen einteilen nach Grammatiken

Die meisten Sprachen können nicht mit Grammatiken beschrieben werden (abzählbar viele Grammatiken vs. überabzählbar viele Sprachen)

- Die meisten praktisch relevanten Sprachen haben aber eine Grammatik
- Manche Sprachen kann man sogar mit einfacheren Sonderformen von Grammatiken beschreiben

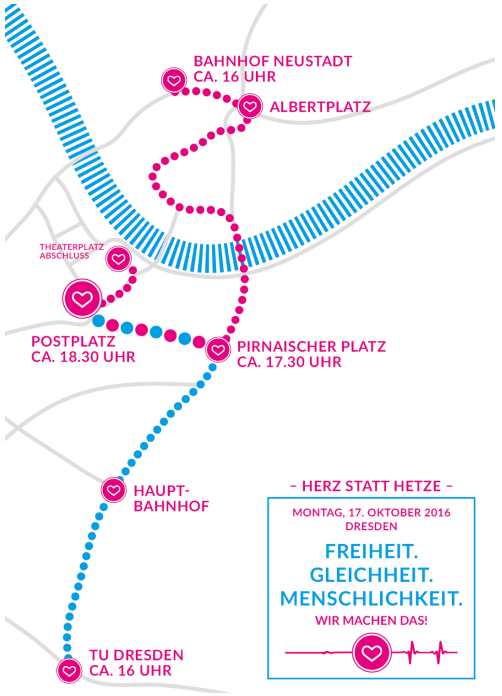
Ministerio de Cultura de la Nación 2015, CC-BY-SA 2.0

Man kann Sprachen nach Komplexität ihrer Grammatiken unterteilen.

Noam Chomsky, 2015

Ministerio de Cultura de la Nación 2015, CC-BY-SA 2.0

„This world is full of suffering, distress, violence and catastrophes.
Students must decide: does something concern you or not?
I say: look around, analyze the problems, ask yourself what you can do and set out on the work.“ [ZEIT Campus, 2011]



Die Chomsky-Hierarchie

Die Chomsky-Hierarchie unterteilt Grammatiken in vier Stufen:

- **Typ 0**: beliebige Grammatiken
- **Typ 1**[a]: kontextsensitive Grammatiken:
  Alle Regeln $w \to v$ erfüllen die Bedingung $|w| \leq |v|$.
- **Typ 2**: kontextfreie Grammatiken:
  Alle Regeln haben die Form $A \to v$ für eine Variable A.
- **Typ 3**: reguläre Grammatiken:
  Alle Regeln haben eine der Formen

$$A \to cB \qquad A \to c \qquad A \to \epsilon$$

  wobei A und B Variablen sind und c ein Terminalsymbol ist.

[a] Erste Version. Wir werden Typ 1 später noch leicht erweitern.

Chomsky-Hierarchie: Beispiele

Beispiel: Eine Typ-3-Grammatik (regulär):

$$S \to bA \qquad A \to zA \qquad A \to z$$

Beispiel: Eine Typ-2-Grammatik (kontextfrei, nicht regulär):

$$S \to bBz \qquad B \to BzB \qquad B \to \epsilon$$

Beispiel: Eine Typ-1-Grammatik (kontextsensitiv, nicht kontextfrei):

$$S \to bC \qquad bC \to Sz \qquad Sz \to bz$$

Beispiel: Eine Typ-0-Grammatik (nicht kontextsensitiv):

$$S \to bzDzDz \qquad zD \to zzD \qquad zzzDz \to z$$

Alle vier Grammatiken erzeugen die gleiche Sprache $\{b\} \circ \{z\}^+$.

Kontextfrei und kontextsensitiv

Der Namensgebung liegt die folgende Idee zu Grunde:

Typische kontextsensitive Regel:

$$aBc \to avc$$

„B kann im Kontext von a und c durch v ersetzt werden.“

Typische kontextfreie Regel:

$$B \to v$$

„B kann in jedem Kontext durch v ersetzt werden.“

Allerdings können kontextsensitive Regeln auch andere Formen haben.

Typen von Sprachen

Die Chomsky-Hierarchie kann auf Sprachen angewendet werden:

Eine Sprache **L** ist vom Typ i wenn sie durch eine Grammatik G von Typ i beschrieben wird, d.h. wenn $\mathbf{L}(G) = \mathbf{L}$. Es gibt demnach reguläre, kontextfreie und kontextsensitive Sprachen.

Die Beispiele haben gezeigt: man kann die gleiche Sprache oft mit Grammatiken verschiedenen Typs beschreiben.

Aber wir werden sehen: für viele Sprachen gibt es einen maximalen Typ von Grammatik, z.B. gibt es kontextfreie Sprachen, die nicht regulär sind.

Beziehungen zwischen Chomskys Typen

Bilden die Typen wirklich eine Hierarchie

„Typ 3 $\subseteq$ Typ 2 $\subseteq$ Typ 1 $\subseteq$ Typ 0“?

- Jede Grammatik von Typ 1, 2 oder 3 ist auch von Typ 0 (per Definition)
- Jede reguläre Grammatik ist kontextfrei (die Typ-2-Regeln $A \to cB$, $A \to c$ und $A \to \epsilon$ sind auch von Typ 1)
- Aber nicht jede kontextfreie Grammatik ist kontextsensitiv (Regeln der Form $A \to \epsilon$ sind nicht erlaubt)

Allgemein können Typ-1-Sprachen in unserer Definition nie das leere Wort enthalten!

Typ 1 mit ϵ-Regeln

Wir erweitern unsere Definition wie folgt:

Eine Grammatik $G = \langle V, \Sigma, P, S\rangle$ ist von Typ 1 (kontextsensitiv), wenn eine der folgenden Bedingungen gilt:

(1) Alle Regeln $w \to v$ erfüllen die Bedingung $|w| \leq |v|$ (ursprüngliche Definition)

(2) Es gibt eine Regel $S \to \epsilon$ und alle anderen Regeln $w \to v$ erfüllen zwei Bedingungen:
   (a) $|w| \leq |v|$ (insbesondere ist also $v \neq \epsilon$)
   (b) S kommt nicht in v vor

   (Sonderfall mit ϵ-Regeln)

Beispiel: Die reguläre Grammatik G mit den Regeln

$$S \to \epsilon \mid A \qquad A \to \mathtt{nyan}A \qquad A \to \mathtt{nyan}$$

ist kontextsensitiv nach der erweiterten Definition. $\mathbf{L}(G) = \{\mathtt{nyan}\}^*$.

Kontextfrei vs. kontextsensitiv mit ϵ

Manche Typ-2-Grammatiken sind weiterhin nicht von Typ 1:

Beispiel: Die kontextfreie (und reguläre) Grammatik

$$S \to bA \qquad A \to zA \qquad A \to \epsilon$$

ist nicht kontextsensitiv, nicht einmal mit ϵ-Erweiterung.

Eine kontextfreie Grammatik $G = \langle V, \Sigma, P, S \rangle$ ist ϵ-frei, wenn eine der folgenden Bedingungen gilt:

(1) Es gibt keine Regel der Form $A \to \epsilon$

(2) Es gibt eine Regel $S \to \epsilon$ und bei allen anderen Regeln $A \to v$ kommt S nicht in v vor.

Offensichtlich: jede ϵ-freie kontextfreie Grammatik ist kontextsensitiv
Weniger offensichtlich: ϵ-freie kontextfreie Grammatiken erzeugen die gleichen Sprachen wie kontextfreie Grammatiken allgemein

Erzeugung ϵ-freier Grammatiken

Der folgende Algorithmus kann ϵ-Regeln eliminieren.

Eingabe: kontextfreie Grammatik (CFG) $G = \langle V, \Sigma, P, S \rangle$
Ausgabe: ϵ-freie CFG $G' = \langle V', \Sigma, P', S' \rangle$ mit $\mathbf{L}(G') = \mathbf{L}(G)$

- Initialisiere $P' := P$ und $V' := V$
- Berechne $V_\epsilon = \{A \in V \mid A \Rightarrow^* \epsilon\}$ (Details dazu später)
- Entferne alle ϵ-Regeln aus P'
- Solange es in P' eine Regel $B \to xAy$ gibt, mit

  $$A \in V_\epsilon \qquad |x| + |y| \geq 1 \qquad B \to xy \notin P'$$

  wähle eine solche Regel und setze $P' := P' \cup \{B \to xy\}$
- Falls $S \in V_\epsilon$ dann definiere ein neues Startsymbol $S' \notin V$, setze $V' := V' \cup \{S'\}$ und $P' := P' \cup \{S' \to S, S' \to \epsilon\}$. Falls $S \notin V_\epsilon$, dann verwenden wir einfach $S' := S$ als Startsymbol.

(CFG steht für context-free grammar und wird oft als Abkürzung verwendet)

Berechnung von V_ϵ

Die Variablen $A \in V$ mit $A \Rightarrow^* \epsilon$ kann man leicht rekursiv finden:

Eingabe: kontextfreie Grammatik $G = \langle V, \Sigma, P, S \rangle$
Ausgabe: V_ϵ für diese Grammatik

- Intialisiere $V_\epsilon := \{A \mid A \to \epsilon \in P\}$
- Solange es eine Regel $B \to A_1 \dots A_n \in P$ gibt, so dass gilt

  $$B \notin V_\epsilon \qquad \text{und} \qquad A_i \in V_\epsilon \text{ für alle } i \in \{1, \dots, n\}$$

  wähle eine solche Regel und setze $V_\epsilon := V_\epsilon \cup \{B\}$.

Beispiel: $S \to aB \mid BAA$ $B \to a \mid AC$ $A \to bb \mid \epsilon$
$C \to DS \mid \epsilon$ $D \to aAS$
Initialisierung: $V_\epsilon = \{A, C\}$
Schritt 1: $V_\epsilon = \{A, C, B\}$
Schritt 2: $V_\epsilon = \{A, C, B, S\}$ – Terminierung

Beispiel: Eliminierung von ϵ-Regeln

$$S \to bBz \qquad B \to BzB \qquad B \to \epsilon$$

$V_\epsilon = \{B\}$

$P' = \{$ $S \to bBz$, $B \to BzB$, ~~$B \to \epsilon$~~, $S \to bz$, $B \to Bz$, $B \to zB$, $B \to z$ $\}$

- Initialisiere $P' := P$
- Entferne alle ϵ-Regeln aus P'
- Solange es in P' eine Regel $B \to xAy$ gibt, mit

  $A \in V_\epsilon$ $|x| + |y| \geq 1$ $B \to xy \notin P'$

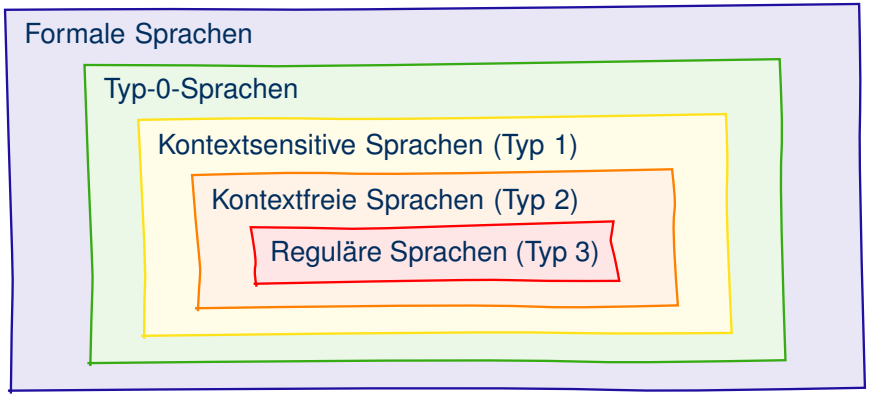
  wähle eine solche Regel und setze $P' := P' \cup \{B \to xy\}$

Zusammenfassung der Ergebnisse

Wir können jede CFG in eine ϵ-freie CFG umschreiben. Daher gilt:

Satz: Jede kontextfreie Sprache ist kontextsensitiv im erweiterten Sinn (mit ϵ-Sonderfall).

Damit gelten alle Inklusionen der Hierarchie. Grafisch dargestellt:



Zusammenfassung und Ausblick

Nicht alle Sprachen sind mathematisch beschreibbar oder durch Computer entscheidbar

Induktion (= rekursive Konstruktion von Beweisen) ist oft nötig, um Behauptungen über unendliche Sprachen zu beweisen

Grammatiken können (manche) Sprachen beschreiben. Die Chomsky-Hierarchie teilt Grammatiken in vier Typen ein.

Offene Fragen:

- Was gewinnt man durch die Beschränkung auf spezielle Typen?
- Sind die Typen wirklich unterschiedlich?
- Wie kommt Berechnung ins Spiel?