

Entscheidbarkeit des Wortproblems für Typ 1-Sprachen

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik

Stand: 25. April 2023

Folien ursprünglich von PD Dr. David Sabel



- ▶ Als Einschub wechseln wir nochmal zu Typ-1-Sprachen und **holen einen Beweis nach**
- ▶ Nächstes Mal geht es wieder mit den regulären Sprachen weiter

Wiederholung:

Definition (Entscheidbarkeit)

Eine Sprache L heißt **entscheidbar**, wenn es einen Algorithmus gibt, der bei Eingabe eines Wortes w in endlicher Zeit feststellt, ob $w \in L$ gilt oder nicht.

Eng verwandt dazu ist:

Definition (Wortproblem für Typ i -Sprachen)

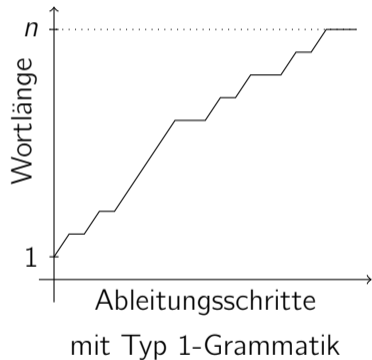
Das **Wortproblem** für Typ i -Sprachen ist die Frage, ob für eine gegebene Typ i -Grammatik $G = (V, \Sigma, P, S)$ und ein Wort $w \in \Sigma^*$ gilt: $w \in L(G)$ oder $w \notin L(G)$.

Satz

Das Wortproblem für Typ 1-Sprachen ist entscheidbar: Es gibt einen Algorithmus, der bei Eingabe von Typ 1-Grammatik G und Wort w nach endlicher Zeit entscheidet, ob $w \in L(G)$ oder $w \notin L(G)$ gilt.

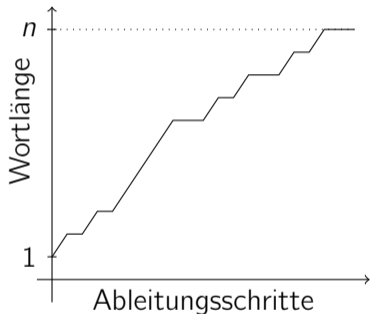
Idee zum Beweis des Satzes

Ableitung eines Wortes der Länge n



Idee zum Beweis des Satzes

Ableitung eines Wortes der Länge n



mit Typ 1-Grammatik

Idee zum Entscheiden des Wortproblems:

- ▶ Betrachte $S \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_m$ mit $|w_m| = n$
- ▶ Da Typ 1-Produktionen nicht verkürzend sind:
 $\forall 1 \leq i \leq m : |w_i| \leq n$
- ▶ Probiere systematisch für alle Satzformen der Länge $\leq n$ durch, ob sie vom Startsymbol aus ableitbar sind
- ▶ Es gibt nur endlich viele Satzformen der Länge $\leq n$ und jede Typ 1-Grammatik leitet nur endlich viele Satzformen der Länge $\leq n$ her, **ohne dabei längere Satzformen zwischendrin herzuleiten**
- ▶ Herleitbare Satzformen der Länge n können **rekursiv** aus den Satzformen der Länge $< n$ berechnet werden

Beweis: Entscheidbarkeit des Wortproblems für Typ 1-Sprachen

Sei $G = (V, \Sigma, P, S)$ eine Typ 1-Grammatik und $w \in \Sigma^*$.

Für $m \in \mathbb{N}$, $n \in \mathbb{N}_{>0}$ sei

L_m^n = Menge aller Satzformen der Länge höchstens n ,
die in höchstens m Schritten von S aus ableitbar sind

$L_m^n := \{w \in (V \cup \Sigma)^* \mid |w| \leq n \text{ und } S \Rightarrow_G^k w \text{ mit } k \leq m\}$

Beweis: Entscheidbarkeit des Wortproblems für Typ 1-Sprachen

Sei $G = (V, \Sigma, P, S)$ eine Typ 1-Grammatik und $w \in \Sigma^*$.

Für $m \in \mathbb{N}$, $n \in \mathbb{N}_{>0}$ sei

L_m^n = Menge aller Satzformen der Länge höchstens n ,
die in höchstens m Schritten von S aus ableitbar sind

$$L_m^n := \{w \in (V \cup \Sigma)^* \mid |w| \leq n \text{ und } S \Rightarrow_G^k w \text{ mit } k \leq m\}$$

Rekursive Berechnung der Mengen (für $n > 0$):

$$L_0^n := \{S\}$$

$$L_m^n := \text{next}(L_{m-1}^n, n) \text{ für } m > 0$$

$$\text{wobei } \text{next}(L, n) := L \cup \{w' \mid w \in L, w \Rightarrow_G w', |w'| \leq n\}$$

Beweis: Entscheidbarkeit des Wortproblems für Typ 1-Sprachen

Sei $G = (V, \Sigma, P, S)$ eine Typ 1-Grammatik und $w \in \Sigma^*$.

Für $m \in \mathbb{N}$, $n \in \mathbb{N}_{>0}$ sei

L_m^n = Menge aller Satzformen der Länge höchstens n ,
die in höchstens m Schritten von S aus ableitbar sind

$$L_m^n := \{w \in (V \cup \Sigma)^* \mid |w| \leq n \text{ und } S \Rightarrow_G^k w \text{ mit } k \leq m\}$$

Rekursive Berechnung der Mengen (für $n > 0$):

$$L_0^n := \{S\}$$

$$L_m^n := \text{next}(L_{m-1}^n, n) \text{ für } m > 0$$

$$\text{wobei } \text{next}(L, n) := L \cup \{w' \mid w \in L, w \Rightarrow_G w', |w'| \leq n\}$$

Die Berechnung terminiert, da die Mengen endlich sind: $|L_m^n| \leq |\Sigma \cup V|^{n+1}$.

Wir können auch iterativ aus L_{i-1}^n die nachfolgende Menge L_i^n berechnen.

Beweis: Entscheidbarkeit des Wortproblems für Typ 1-Sprachen (2)

Iterative Berechnung:

- ▶ Starte mit $L_0^n = \{S\}$
- ▶ Für $i = 1, 2, 3, \dots$ berechne $L_i^n = \text{next}(L_{i-1}^n, n)$
- ▶ Stoppe dabei, wenn $L_i^n = L_{i-1}^n$
- ▶ Prüfe, ob $w \in L_i^n$ gilt

Korrektheit: Wenn $L_i^n = L_{i-1}^n$, dann gilt $L_{i+k}^n = L_{i-1}^n$ für alle $k \in \mathbb{N}$ und daher enthält L_{i-1}^n genau alle aus S ableitbaren Wörter der Länge n .

Terminierung: Beweis durch Widerspruch.

- ▶ Nimm an, die Berechnung stoppt nicht.
- ▶ Da $L_{i-1}^n \subseteq L_i^n$, muss für alle $i \in \mathbb{N}$ gelten: $L_i^n \supset L_{i-1}^n$.
- ▶ Daher gilt für alle $i \in \mathbb{N}$: $|L_i^n| > |L_{i-1}^n|$.
- ▶ Widerspruch zu $|L_i^n| \leq |V \cup \Sigma|^{n+1}$ für alle $i \in \mathbb{N}$.

Algorithmus 2: Entscheiden des Wortproblems für Typ 1-Grammatiken

Eingabe: Typ 1-Grammatik $G = (V, \Sigma, P, S)$ und ein Wort $w \in \Sigma^*$

Ausgabe: Ja, wenn $w \in L(G)$ und Nein, wenn $w \notin L(G)$

Beginn

$n := |w|;$

$L := \{S\};$

wiederhole

$L_{old} := L;$

$L := next(L_{old}, n);$

bis $(w \in L)$ oder $(L_{old} = L);$

wenn $w \in L$ **dann**

return Ja;

sonst

return Nein;

Ende

Ende

Beispiel

$G = (\{S, B\}, \{a, b, c\}, P, S)$ mit

$P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$

Wir berechnen L_i^6 für alle i :

Beispiel

$G = (\{S, B\}, \{a, b, c\}, P, S)$ mit

$P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$

Wir berechnen L_i^G für alle i :

$$L_0^G = \{S\}$$

Beispiel

$G = (\{S, B\}, \{a, b, c\}, P, S)$ mit

$P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$

Wir berechnen L_i^6 für alle i :

$$L_0^6 = \{S\}$$

$$L_1^6 = \text{next}(L_0^6, 6) = L_0^6 \cup \{w' \mid w \in L_0^6, w \Rightarrow w', |w'| \leq 6\} = \{S, aSBc, abc\}$$

da $S \Rightarrow aSBc$ und $S \Rightarrow abc$

Beispiel

$G = (\{S, B\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$

Wir berechnen L_i^6 für alle i :

$$L_0^6 = \{S\}$$

$$L_1^6 = \text{next}(L_0^6, 6) = L_0^6 \cup \{w' \mid w \in L_0^6, w \Rightarrow w', |w'| \leq 6\} = \{S, aSBc, abc\}$$

da $S \Rightarrow aSBc$ und $S \Rightarrow abc$

$$L_2^6 = \text{next}(L_1^6, 6) = L_1^6 \cup \{w' \mid w \in L_1^6, w \Rightarrow w', |w'| \leq 6\} \\ = \{S, aSBc, abc, abcBc\}$$

da $aSBc \Rightarrow aaSBcBc$ (zu lang) und $aSBc \Rightarrow abcBc$

Beispiel

$G = (\{S, B\}, \{a, b, c\}, P, S)$ mit

$P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$

Wir berechnen L_i^6 für alle i :

$$L_0^6 = \{S\}$$

$$L_1^6 = \text{next}(L_0^6, 6) = L_0^6 \cup \{w' \mid w \in L_0^6, w \Rightarrow w', |w'| \leq 6\} = \{S, aSBc, abc\}$$

da $S \Rightarrow aSBc$ und $S \Rightarrow abc$

$$\begin{aligned} L_2^6 &= \text{next}(L_1^6, 6) = L_1^6 \cup \{w' \mid w \in L_1^6, w \Rightarrow w', |w'| \leq 6\} \\ &= \{S, aSBc, abc, aabcBc\} \end{aligned}$$

da $aSBc \Rightarrow aaSBcBc$ (zu lang) und $aSBc \Rightarrow aabcBc$

$$\begin{aligned} L_3^6 &= \text{next}(L_2^6, 6) = L_2^6 \cup \{w' \mid w \in L_2^6, w \Rightarrow w', |w'| \leq 6\} \\ &= \{S, aSBc, abc, aabcBc, aabBcc\} \end{aligned}$$

da $aabcBc \Rightarrow aabBcc$

Beispiel (2)

$G = (\{S, B\}, \{a, b, c\}, P, S)$ mit

$P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$

...

$L_3^6 = \{S, aSBc, abc, aabcBc, aabBcc\}$

Beispiel (2)

$G = (\{S, B\}, \{a, b, c\}, P, S)$ mit

$P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$

...

$L_3^6 = \{S, aSBc, abc, aabcBc, aabBcc\}$

$L_4^6 = \text{next}(L_3^6, 6) = L_3^6 \cup \{w' \mid w \in L_3^6, w \Rightarrow w', |w'| \leq 6\}$
 $= \{S, aSBc, abc, aabcBc, aabBcc, aabbcc\}$

da $aabBcc \Rightarrow aabbcc$

Beispiel (2)

$G = (\{S, B\}, \{a, b, c\}, P, S)$ mit

$P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$

...

$L_3^6 = \{S, aSBc, abc, aabcBc, aabBcc\}$

$L_4^6 = \text{next}(L_3^6, 6) = L_3^6 \cup \{w' \mid w \in L_3^6, w \Rightarrow w', |w'| \leq 6\}$
 $= \{S, aSBc, abc, aabcBc, aabBcc, aabbcc\}$

da $aabBcc \Rightarrow aabbcc$

$L_5^6 = \text{next}(L_4^6, 6) = L_4^6 \cup \{w' \mid w \in L_4^6, w \Rightarrow w', |w'| \leq 6\}$
 $= \{S, aSBc, abc, aabcBc, aabBcc, aabbcc\} = L_4^6$

Beispiel (2)

$G = (\{S, B\}, \{a, b, c\}, P, S)$ mit

$P = \{S \rightarrow aSBc, S \rightarrow abc, cB \rightarrow Bc, bB \rightarrow bb\}$

...

$L_3^6 = \{S, aSBc, abc, aabcBc, aabBcc\}$

$L_4^6 = \text{next}(L_3^6, 6) = L_3^6 \cup \{w' \mid w \in L_3^6, w \Rightarrow w', |w'| \leq 6\}$
 $= \{S, aSBc, abc, aabcBc, aabBcc, aabbcc\}$

da $aabBcc \Rightarrow aabbcc$

$L_5^6 = \text{next}(L_4^6, 6) = L_4^6 \cup \{w' \mid w \in L_4^6, w \Rightarrow w', |w'| \leq 6\}$
 $= \{S, aSBc, abc, aabcBc, aabBcc, aabbcc\} = L_4^6$

d.h. $L_i^6 = \{S, aSBc, abc, aabcBc, aabBcc, aabbcc\}$ für alle $i \geq 4$

Wortproblem für Typ 2- und Typ 3-Sprachen

Korollar

Das Wortproblem für Typ 2- und Typ 3-Sprachen ist entscheidbar.

Wortproblem für Typ 2- und Typ 3-Sprachen

Korollar

Das Wortproblem für Typ 2- und Typ 3-Sprachen ist entscheidbar.

Bemerkungen:

- ▶ Der Algorithmus für Typ 1-Sprachen hat exponentielle Laufzeitkomplexität
- ▶ Das Wortproblem für Typ 2- und das Wortproblem für Typ 3-Sprachen sind in polynomieller Zeit lösbar
- ▶ Das Wortproblem für Typ 0-Sprachen ist unentscheidbar (aber rekursiv aufzählbar)