

Satz 90

Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA. Der Zeitaufwand des obigen Minimalisierungsalgorithmus ist $O(|Q|^2|\Sigma|)$.

Beweis:

Für jedes $a \in \Sigma$ muss jede Position in der Tabelle nur konstant oft besucht werden. □

6. Einige Ergebnisse zur Entscheidbarkeit

Beispiel 91

Wie wir bereits wissen, ist das Wortproblem für reguläre Grammatiken entscheidbar. Wenn L durch einen deterministischen endlichen Automaten gegeben ist, ist dies sogar in linearer Laufzeit möglich. Allerdings gilt, dass bei der Überführung eines nichtdeterministischen endlichen Automaten in einen deterministischen endlichen Automaten die Komplexität exponentiell zunehmen kann.

Die folgenden Probleme sind für Chomsky-3-Sprachen (also die Klasse der regulären Sprachen) entscheidbar:

Wortproblem: Ist ein Wort w in $L(G)$ (bzw. $L(A)$)?

Das Wortproblem ist für alle Grammatiken mit einem Chomsky-Typ größer 0 entscheidbar. Allerdings wächst die Laufzeit exponentiell mit der Wortlänge n . Für Chomsky-2- und Chomsky-3-Sprachen (d.h. -Grammatiken) gibt es wesentlich effizientere Algorithmen.

Leerheitsproblem: Ist $L(G) = \emptyset$?

Das Leerheitsproblem ist für Grammatiken vom Chomsky-Typ 2 und 3 entscheidbar.

Für andere Typen lassen sich Grammatiken konstruieren, für die nicht mehr entscheidbar ist, ob die Sprache leer ist.

Endlichkeitsproblem: Ist $|L(G)| < \infty$?

Das Endlichkeitsproblem ist für alle regulären Grammatiken lösbar.

Lemma 92

Sei n eine geeignete Pumping-Lemma-Zahl, die zur regulären Sprache L gehört. Dann gilt:

$$|L| = \infty \text{ gdw } (\exists z \in L)[n \leq |z| < 2n] .$$

Beweis:

Wir zeigen zunächst \Leftarrow :

Aus dem Pumping-Lemma folgt: $z = uvw$ für $|z| \geq n$ und $uv^i w \in L$ für alle $i \in \mathbb{N}_0$. Damit erzeugt man unendlich viele Wörter.

Nun wird \Rightarrow gezeigt:

Dass es ein Wort z mit $|z| \geq n$ gibt, ist klar (es gibt ja unendlich viele Wörter). Mit Hilfe des Pumping-Lemmas lässt sich ein solches Wort auf eine Länge $< 2n$ reduzieren. \square

Damit kann das Endlichkeitsproblem auf das Wortproblem zurückgeführt werden.

Schnittproblem: Ist $L(G_1) \cap L(G_2) = \emptyset$?

Das Schnittproblem ist für die Klasse der regulären Grammatiken entscheidbar, nicht aber für die Klasse der Chomsky-2-Grammatiken.

Äquivalenzproblem: Ist $L(G_1) = L(G_2)$?

Das Äquivalenzproblem lässt sich auch wie folgt formulieren:

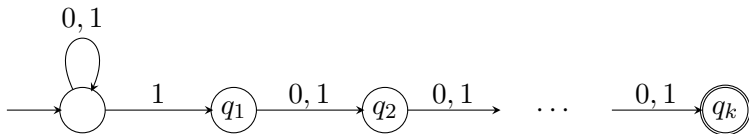
$$L_1 = L_2 \quad \Leftrightarrow \quad (L_1 \cap \overline{L_2}) \cup (L_2 \cap \overline{L_1}) = \emptyset$$

Wichtig für eine effiziente Lösung der Probleme ist, wie die Sprache gegeben ist. Hierzu ein Beispiel:

Beispiel 93

$L = \{w \in \{0, 1\}^*; \text{das } k\text{-letzte Bit von } w \text{ ist gleich } 1\}$

Ein NFA für diese Sprache ist gegeben durch:



Insgesamt hat der NFA $k + 1$ Zustände. Man kann nun diesen NFA in einen deterministischen Automaten umwandeln und stellt fest, dass der entsprechende DFA $\Omega(2^k)$ Zustände hat.

Da die Komplexität eines Algorithmus von der Größe der Eingabe abhängt, ist dieser Unterschied in der Eingabegröße natürlich wesentlich, denn es gilt:

kurze Eingabe wie beim NFA \Rightarrow **wenig Zeit** für einen effizienten Algorithmus,

lange Eingabe wie beim DFA \Rightarrow **mehr Zeit** für einen effizienten Algorithmus.

7. Das Wortproblem für kontextfreie Sprachen

7.1 Die Chomsky-Normalform

Sei $G = (V, \Sigma, P, S)$ eine kontextfreie Grammatik.

Definition 94

Eine kontextfreie Grammatik G ist in **Chomsky-Normalform**, falls alle Produktionen eine der Formen

$$\begin{array}{ll} A \rightarrow a & A \in V, a \in \Sigma, \\ A \rightarrow BC & A, B, C \in V, \text{ oder} \\ S \rightarrow \epsilon & \end{array}$$

haben.

Algorithmus zur Konstruktion einer (äquivalenten) Grammatik in Chomsky-Normalform

Eingabe: Eine kontextfreie Grammatik $G = (V, \Sigma, P, S)$

- 1 Wir fügen für jedes $a \in \Sigma$ zu V ein neues Nichtterminal Y_a hinzu, ersetzen in allen Produktionen a durch Y_a und fügen $Y_a \rightarrow a$ als neue Produktion zu P hinzu.

/* linearer Zeitaufwand, Größe vervierfacht sich höchstens */

- 2 Wir ersetzen jede Produktion der Form

$$A \rightarrow B_1 B_2 \cdots B_r \quad (r \geq 3)$$

durch

$$A \rightarrow B_1 C_2, C_2 \rightarrow B_2 C_3, \dots, C_{r-1} \rightarrow B_{r-1} B_r,$$

wobei C_2, \dots, C_{r-1} neue Nichtterminale sind.

/* linearer Zeitaufwand, Größe vervierfacht sich höchstens */

- 3 Für alle $C, D \in V$, $C \neq D$, mit

$$C \rightarrow^+ D,$$

füge für jede Produktion der Form

$$A \rightarrow BC \in P \text{ bzw. } A \rightarrow CB \in P$$

die Produktion

$$A \rightarrow BD \text{ bzw. } A \rightarrow DB$$

zu P hinzu.

/* quadratischer Aufwand **pro** A */

- 4 Für alle $\alpha \in V^2 \cup \Sigma$, für die $S \rightarrow^* \alpha$, füge $S \rightarrow \alpha$ zu P hinzu.
- 5 Streiche alle Produktionen der Form $A \rightarrow B$ aus P .

Zusammenfassend können wir festhalten:

Satz 95

Aus einer kontextfreien Grammatik $G = (V, \Sigma, P, S)$ der Größe $s(G)$ kann in Zeit $O(|V|^2 \cdot s(G))$ eine äquivalente kontextfreie Grammatik in Chomsky-Normalform der Größe $O(|V|^2 \cdot s(G))$ erzeugt werden.

7.2 Der Cocke-Younger-Kasami-Algorithmus

Der CYK-Algorithmus (oft auch Cocke-Kasami-Younger, CKY) entscheidet das **Wortproblem** für kontextfreie Sprachen, falls die Sprache in Form einer Grammatik in Chomsky-Normalform gegeben ist.

Eingabe: Grammatik $G = (V, \Sigma, P, S)$ in Chomsky-Normalform, $w = w_1 \dots w_n \in \Sigma^*$ mit der Länge n . O.B.d.A. $n > 0$.

Definition

$$V_{ij} := \{A \in V; A \rightarrow^* w_i \dots w_j\}.$$

Es ist klar, dass $w \in L(G) \Leftrightarrow S \in V_{1n}$.

Der CYK-Algorithmus berechnet alle V_{ij} rekursiv nach wachsendem $j - i$. Den Anfang machen die

$$V_{ii} := \{A \in V; A \rightarrow w_i \in P\},$$

der rekursive Aufbau erfolgt nach der Regel

$$V_{ij} = \bigcup_{i \leq k < j} \{A \in V; (A \rightarrow BC) \in P \wedge B \in V_{ik} \wedge C \in V_{k+1,j}\} \quad \text{für } i < j.$$

Die Korrektheit dieses Aufbaus ist klar, wenn die Grammatik in Chomsky-Normalform vorliegt.