Übung 5: Ogden-Lemma und Kellerautomaten

Theoretische Informatik Sommersemester 2014

Markus Kaiser

22. Mai 2014

Induktive Sprachdefinition

Die induktive Definition zu einer Grammatik G ergibt sich direkt aus ihren Produktionen. Dabei werden kleinere Worte zu größeren Worten zusammengesetzt, die Definition erfolgt bottom-up.

Beispiel

Mit den Produktionen $S \rightarrow 0S1 \mid S11 \mid 1$:

$$\begin{aligned} & & 1 &\in L_G(S) \\ u \in L_G(S) &\implies & 0u1 &\in L_G(S) \\ u \in L_G(S) &\implies & u11 &\in L_G(S) \end{aligned}$$

Also z.B:

$$1 \in L_G(S) \implies 011 \in L_G(S) \implies 01111 \in L_G(S)$$

Satz (Ogden Lemma für kontextfreie Sprachen)

Sei $L \subseteq \Sigma^*$ kontextfrei.
Dann gibt es ein $n > 0$, so dass für jedes $z \in L$ mit $|z| \geq n$ gilt:
Für jede Markierung M von mindestens n Buchstaben in z gibt es eine Zerlegung $z = uvwxy$ mit

- $|vx|_M \geq 1$
- $|vwx|_M \leq n$
- $\forall i \geq 0. uv^i wx^i y \in L$

Beispiel (Markierung)

Sei $w = abaabaaa$ ein Wort. Dann ist

$$w = abaabaaa$$

eine Markierung mit $|w|_M = 3$.

Definition (Greibach-Normalform)

Eine kontextfreie Grammatik ist in Greibach-Normalform (GNF) genau dann wenn alle Produktionen außer $S \to \epsilon$ die Form

$$A \to a\alpha \quad \text{mit} \quad a \in \Sigma, \alpha \in V^*$$

haben.

Satz

Zu jeder CFG G existiert eine CFG G' in Greibach-Normalform mit

$$L(G') = L(G)$$

Satz (Einsetzen von Produktionen)

Enthält eine CFG die Produktionen

$$
\begin{aligned}
A &\to \alpha_1 B \alpha_2 \\
B &\to \beta_1 \mid \cdots \mid \beta_k
\end{aligned}
$$

so ändert sich die erzeugte Sprache nicht, wenn man B in A einsetzt.

$$A \to \alpha_1 \beta_1 \alpha_2 \mid \cdots \mid \alpha_1 \beta_k \alpha_2$$

Beispiel

Die Grammatik

$$
\begin{aligned}
S &\to a \mid aBc \\
B &\to b \mid bS
\end{aligned}
$$

ist äquivalent zur Grammatik

$$S \to a \mid abc \mid abSc$$

Definition (Linksrekursive Produktion)

Man nennt eine Produktion linksrekursiv, wenn sie die Form

$$A \to A\alpha_1 \mid \cdots \mid A\alpha_k \mid \beta_1 \mid \cdots \mid \beta_l \quad \text{mit} \quad \alpha_i, \beta_i \in (V \cup \Sigma)^+$$

hat, wobei die β_i nicht mit A beginnen.

Satz (Ersetzen von linksrekursiven Produktionen)

Sei A eine linksrekursive Produktion einer CFG.
Dann ändert sich die erzeugte Sprache nicht, wenn wir A ersetzen durch

$$A \to \beta_1 \mid \cdots \mid \beta_l \mid \beta_1 B \mid \cdots \mid \beta_l B$$
$$B \to \alpha_1 \mid \cdots \mid \alpha_k \mid \alpha_1 B \mid \cdots \mid \alpha_k B$$

B ist niemals linksrekursiv.

GNF Konstruktion

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

1. Nummeriere Nichtterminale
2. Mache Prduktionen aufsteigend und nicht rekursiv
3. Setze Produktionen absteigend ein

GNF Konstruktion

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

1. Nummeriere Nichtterminale
2. Mache Prduktionen aufsteigend und nicht rekursiv
3. Setze Produktionen absteigend ein

Benenne alle Nichtterminale beliebig um in $A_1, \ldots, A_{|V|}$.

$$S \to Ab, \quad A \to aAS \mid \epsilon$$

wird zu

$$A_1 \to A_2 b$$
$$A_2 \to aA_2A_1$$

GNF Konstruktion

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

1. Nummeriere Nichtterminale
2. Mache Prduktionen aufsteigend und nicht rekursiv
3. Setze Produktionen absteigend ein

Betrachte alle Produktionen $A_l \to \dots$ in aufsteigender Reihenfolge.

- Existieren Produktionen der Form $A_l \to A_r\alpha$ mit $r < l$, dann setze A_r in A_l ein.

$$A_1 \to A_2 \mid a \mid b$$
$$A_2 \to A_1 A_1$$

wird zu

$$A_1 \to A_2 \mid a \mid b$$
$$A_2 \to A_2 A_1 \mid aA_1 \mid bA_1$$

- Entferne danach alle linksrekursiven A_l-Produktionen.

GNF Konstruktion

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

1. Nummeriere Nichtterminale
2. Mache Prduktionen aufsteigend und nicht rekursiv
3. Setze Produktionen absteigend ein

Betrachte alle Produktionen $A_l \rightarrow \dots$ in aufsteigender Reihenfolge.

- Existieren Produktionen der Form $A_l \rightarrow A_r\alpha$ mit $r < l$, dann setze A_r in A_l ein.
- Entferne danach alle linksrekursiven A_l-Produktionen.

$$A_2 \rightarrow A_2A_1 \mid aA_1 \mid bA_1$$

wird zu

$$A_2 \rightarrow aA_1 \mid bA_1 \mid aA_1A_3 \mid bA_1A_3$$
$$A_3 \rightarrow A_1 \mid A_1A_3$$

GNF Konstruktion

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

1. Nummeriere Nichtterminale
2. Mache Prduktionen aufsteigend und nicht rekursiv
3. Setze Produktionen absteigend ein

Betrachte alle Produktionen $A_l \to \dots$ in absteigender Reihenfolge.

- Existieren Produktionen der Form $A_l \to A_r\alpha$ mit $r > l$, dann setze A_r in A_l ein.

$$
\begin{aligned}
A_1 &\to a \mid b \mid A_2 \\
A_2 &\to aA_1 \mid bA_1 \mid aA_1A_3 \mid bA_1A_3 \\
A_3 &\to bA_3 \mid c
\end{aligned}
$$

A_1 wird zu

$$A_1 \to a \mid b \mid aA_1 \mid bA_1 \mid aA_1A_3 \mid bA_1A_3$$

Definition (Kellerautomat)

Ein PDA (Push-Down-Automaton) ist ein Tupel $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ aus einer/einem

- endlichen Menge von Zuständen Q
- endlichen Eingabealphabet Σ
- endlichen Kelleralphabet Γ
- Übergangsfunktion $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to P(Q \times \Gamma^*)$
- Startzustand $q_0 \in Q$
- Kellerinitialisierung $Z_0 \in \Gamma$
- Menge von Endzuständen $F \subseteq Q$

$$q_i \xrightarrow{a, X / \gamma} q_j$$

Definition (Kellerautomat)

Ein PDA (Push-Down-Automaton) ist ein Tupel $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ aus einer/einem

- Übergangsfunktion $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow P(Q \times \Gamma^*)$

Definition (Akzeptanz)

Ein PDA P akzeptiert $w \in \Sigma^*$ mit Endzustand gdw

$$\exists f \in F, \gamma \in \Gamma^*.(q_0, w, Z_0) \rightarrow_P^* (f, \epsilon, \gamma)$$

Ein PDA P akzeptiert $w \in \Sigma^*$ mit leerem Keller gdw

$$\exists q \in Q.(q_0, w, Z_0) \rightarrow_P^* (q, \epsilon, \epsilon)$$

Definition (Kellerautomat)

Ein PDA (Push-Down-Automaton) ist ein Tupel $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ aus einer/einem

- Übergangsfunktion $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow P(Q \times \Gamma^*)$

Beispiel

PDA akzeptierend mit leerem Keller zu $L = \{a^n b^n \mid n \in \mathbb{N}_0\}$.