R. Freund und M. Oswald (Hrsg.): 16. Theorietag Automaten und Formale Sprachen, 27.–29.9.2006,
TU Wien, Österreich, Proceedings, pp. 40–45, 2006.

Kontextfreie PC-Grammatiksysteme zum Erzeugen einer Sprache mit Verschränkung

Jürgen Dassow *und* Bianca Truthe

Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik – IWS
Universitätsplatz 2
D-39106 Magdeburg

e-Mail: `{dassow,truthe}@iws.cs.uni-magdeburg.de`

Zusammenfassung

Beim Modellieren natürlicher Sprachen spielen drei Sprachen eine besondere Rolle. Die Aspekte der mehrfachen oder verschränkten Übereinstimmung und der Wiederholung werden durch die Sprachen $K_1 = \{ a^n b^n c^n \mid n \geq 1 \}$, $K_2 = \{ a^n b^m c^n d^m \mid m \geq 1,\ n \geq 1 \}$ bzw. $K_3 = \{ ww \mid w \in \{ a, b \}^+ \}$ widerspiegelt.

In der vorliegenden Arbeit werden drei parallel kommunizierende Grammatiksysteme (kurz PC-Grammatiksysteme) angegeben, die kontextfrei sind und die nicht-kontextfreie Sprache K_2 erzeugen, aber weniger Komponenten haben als bisherige Systeme. In zwei Fällen ist das Ergebnis optimal.

1 Einleitung und Definitionen

Ein parallel kommunizierendes Grammatiksystem (kurz PC-Grammatiksystem) besteht aus einzelnen Grammatiken, von denen jede ihre eigenen Regeln und Satzformen hat, die aber auf dem gleichen Alphabet an einer gemeinsamen Aufgabe arbeiten. Zum Lösen der Aufgabe dürfen die Komponenten (Grammatiken) miteinander kommunizieren. Entsprechend der Monographie [2] wird eine Kommunikation durch eine Anfrage einer Komponente an eine andere angeregt, in der das gesamte Wort der gefragten Komponente angefordert wird. Es wird angenommen, dass die Komponenten des Systems synchron arbeiten. In jeder Zeiteinheit führt jede Komponente genau einen Schritt aus. Wenn keine Komponente fragt, leitet jede Grammatik ihre aktuelle Satzform durch einen Ersetzungsschritt entsprechend ihrer Regeln ab (falls eine Grammatik bereits bei einem Terminalwort angekommen ist, wird dieses Wort beibehalten und die Arbeit der anderen nicht gestört; falls in der Satzform einer Grammatik nur solche Nichtterminale auftreten, für die diese Grammatik keine Ersetzungsregeln hat, blockiert sie die Arbeit des Gesamtsystems). Wenn Komponenten kommunizieren, so werden nach Möglichkeit alle Anfragen gleichzeitig beantwortet. Sollte eine angesprochene Komponente ihrerseits eine Anfrage gestellt haben, so wartet sie mit dem Antworten, bis ihre Anfrage bearbeitet worden ist. Dabei kann es passieren, dass zwei Komponenten auf die jeweils andere warten – was zum Stillstand („Deadlock“) des Systems führt. Im Rücksetzmodus fängt jede Komponente, nachdem sie alle Anfragen beantwortet hat, wieder mit ihrem Startsymbol an; im Fortsetzmodus arbeiten die Komponenten mit ihren bisherigen Wörtern weiter. Die an einer Kommunikation nicht beteiligten Komponenten warten ab; insbesondere führen sie keinen

Ersetzungsschritt aus. Auch jene Komponenten, deren Anfrage in einem Schritt nicht beantwortet wird, ändern an ihrer Satzform nichts.

In der Arbeit [4] wurden einige PC-Grammatiksysteme für die Sprachen $K_1 = \{ a^n b^n c^n \mid n \geq 1 \}$ und $K_3 = \{ ww \,|\, w \in \{ a,b \}^+ \}$ sowie lineare und rechts-lineare PC-Grammatiksysteme für die Sprache $K_2 = \{ a^n b^m c^n d^m \mid m \geq 1,\ n \geq 1 \}$ vorgestellt. In der vorliegenden Arbeit werden kontextfreie PC-Grammatiksysteme angegeben, die die Sprache K_2 erzeugen, aber weniger Komponenten haben als bisherige Systeme.

Die hier verwendeten Definitionen und Notationen lehnen sich an jene aus der Monographie [2] an. Es sei $n \geq 1$ eine natürliche Zahl. Ein PC-Grammatiksystem mit n Komponenten ist ein $(n+3)$-Tupel

$$\Gamma^n = (N, T, K, (P_1, S_1), (P_2, S_2), \ldots, (P_n, S_n))$$

mit

- einer endlichen Menge N (der Menge der Nichtterminale),
- einer endlichen Menge T (der Menge der Terminale) mit $T \cap N = \emptyset$,
- einer Menge K von n Anfragesymbolen ($K \cap (N \cup T) = \emptyset$) und
- n Paaren (P_i, S_i) (den Komponenten des Systems), wobei $S_i \in N$ gilt und $G_i = (N \cup K, T, P_i, S_i)$ eine Chomsky-Grammatik ist.

Ein PC-Grammatiksystem heißt zentralistisch, falls Anfragesymbole nur in den Regeln der ersten Komponente auftreten, sonst nicht-zentralistisch. Die erste Komponente wird auch mit „Meister“ und die anderen Komponenten werden mit „Mitarbeiter“ bezeichnet. Ein PC-Grammatiksystem heißt kontextfrei (*CF*), wenn die Regeln in den Komponenten kontextfrei sind.

Eine Konfiguration eines PC-Grammatiksystems mit n Komponenten ist ein n-Tupel $(w_1, \ldots, w_n)$ mit $w_i \in (N \cup T \cup K)^*$. Eine Konfiguration $(x_1, x_2, \ldots, x_n)$ wird durch ein PC-Grammatiksystem $\Gamma^n = (N, T, K, (P_1, S_1), \ldots, (P_n, S_n))$ in eine Konfiguration $(y_1, y_2, \ldots, y_n)$ im Rücksetz- oder Fortsetzmodus überführt (geschrieben $(x_1, x_2, \ldots, x_n) \Longrightarrow (y_1, y_2, \ldots, y_n)$ mit Index R für „returning“ oder N für „non-returning“), falls einer der beiden folgenden Fälle zutrifft:

1. Keines der Wörter x_i, $i = 1, \ldots, n$, enthält ein Anfragesymbol.
   Für jedes $i = 1, \ldots, n$ gilt: $x_i \Longrightarrow_{G_i} y_i$ oder $x_i \in T^*$ und $y_i = x_i$.

2. Eines der Wörter x_i, $i = 1, \ldots, n$, enthält ein Anfragesymbol.
   Für jeden Index $i = 1, \ldots, n$ gilt: Falls das Wort x_i ein Anfragesymbol enthält, gibt es eine natürliche Zahl $m_i \geq 1$, Wörter $z_{i,1}, \ldots, z_{i,m_i+1} \in (N \cup T)^*$ und natürliche Zahlen $t_{i,1}, \ldots, t_{i,m_i} \in \{ 1, \ldots, n \}$ so, dass $x_i = z_{i,1} Q_{t_{i,1}} z_{i,2} Q_{t_{i,2}} \cdots z_{i,m_i} Q_{t_{i,m_i}} z_{i,m_i+1}$ ist.
   Sollte eines der Wörter $x_{t_{i,j}}$ ($j \in \{ 1, \ldots, m_i \}$) ein Anfragesymbol enthalten, dann gilt $y_i = x_i$, sonst $y_i = z_{i,1} x_{t_{i,1}} z_{i,2} x_{t_{i,2}} \cdots z_{i,t_i} x_{t_{i,m_i}} z_{i,m_i+1}$.
   Falls das Wort x_i kein Anfragesymbol enthält, dann gilt im Fortsetzmodus $y_i = x_i$ und im Rücksetzmodus gibt es zwei mögliche Fälle:

   - Es gibt ein Wort x_k mit dem Anfragesymbol Q_i so, dass $|x_{t_{k,j}}|_K = 0$ für alle $j = 1, \ldots, m_k$ gilt. Dann ist $y_i = S_i$ (falls die i-te Komponente von der k-ten Komponente gefragt wird und keine der von der k-ten Komponente gefragten Komponenten selbst fragt, wird die i-te Komponente nach dem Beantworten auf das Startsymbol zurückgesetzt).

   - Es gibt kein solches Wort x_k. Dann ist $y_i = x_i$ (wenn die i-te Komponenten nicht gefragt wird oder die fragende Komponente nicht von allen gefragten eine Antwort erhält, ändert sich ihr Wort nicht).

Mit $\Longrightarrow^*$ wird der reflexive und transitive Abschluss von $\Longrightarrow$ notiert.

Die Konfiguration $(S_1,\dots,S_n)$ heißt Anfangskonfiguration; jede Konfiguration $(w_1,\dots,w_n)$ mit $w_1 \in T^*$ heißt Endkonfiguration des Systems Γ^n, falls sie von der Anfangskonfiguration aus erreichbar ist, also $(S_1,\dots,S_n) \Longrightarrow^* (w_1,\dots,w_n)$ gilt.

Wenn eine Komponente das Ableiten nicht (mehr) beeinflusst, ihr Inhalt also keine Rolle spielt, dann wird sie durch einen Stern $*$ gekennzeichnet. Eine Komponente hat Einfluss, falls sie gefragt wird oder das System blockiert und zusätzlich im Rücksetzmodus, falls sie eine noch benötigte Komponente fragt (also deren Inhalt sich gegenüber dem Nicht-Gefragt-Werden ändert).

Die von einem PC-Grammatiksystem Γ^n im Rücksetzmodus (oder Fortsetzmodus) erzeugte Sprache $L_{\mathsf{R}}(\Gamma^n)$ (bzw. $L_{\mathsf{N}}(\Gamma^n)$) ist die Menge aller Wörter $w \in T^*$, zu denen es Wörter $w_2,\dots,w_n$ so gibt, dass $(w,w_2,\dots,w_n)$ eine Endkonfiguration von Γ^n ist:

$$L_x(\Gamma^n) = \{\, w \in T^* \mid (S_1,\dots,S_n) \Longrightarrow_x^* (w,w_2,\dots,w_n) \,\} \text{ mit } x \in \{\, \mathsf{R}, \mathsf{N} \,\}.$$

Es bezeichnen *PC$_n$CF* (*CPC$_n$CF*) die Menge aller (zentralistischen) PC-Grammatiksysteme mit höchstens n kontextfreien Komponenten.

2 PC-Grammatiksysteme für die Sprache K_2

In [3] wurden ein kontextfreies PC-Grammatiksystem mit drei Komponenten und ein kontextfreies PC-Grammatiksystem mit zehn Komponenten angegeben, die im Rücksetz- bzw. Fortsetzmodus die Sprache $K_2 = \{\, a^n b^m c^n d^m \mid m \geq 1,\ n \geq 1 \,\}$ erzeugen. Dort wurde offen gelassen, ob es ein zentralistisches PC-Grammatiksystem gibt, das die Sprache K_2 erzeugt. In [1] wurde ein kontextfreies, zentralistisches PC-Grammatiksystem mit vier Komponenten angegeben, das im Rücksetzmodus diese Sprache erzeugt, und eines mit fünf Komponenten, das im Fortsetzmodus diese Sprache generiert.

Somit sind hinsichtlich der Komponentenanzahl die besten PC-Grammatiksysteme, die die Sprache K_2 erzeugen, bisher

- ein *CPC$_5$CF*-System im Fortsetzmodus ([1]),
- ein *CPC$_4$CF*-System im Rücksetzmodus ([1]) und
- ein *PC$_3$CF*-System im Rücksetzmodus ([3]).

In diesem Abschnitt werden

- ein *PC$_2$CF*-System für den Rücksetzmodus,
- ein *CPC$_3$CF*-System für den Rücksetzmodus und
- ein *CPC$_2$CF*-System für den Fortsetzmodus

angegeben, womit die bisherigen Ergebnisse bezogen auf die Anzahl der benötigten Komponenten verbessert werden.

Es gelte

$$\Gamma_1 = (\{\, S_1, S_2, S_1', T, B, D, B', D' \,\}, \{\, a, b, c, d \,\}, \{\, Q_1, Q_2 \,\}, (P_1, S_1), (P_2, S_2))$$

mit

$$\begin{aligned} P_1 &= \{\, S_1 \to aTcD,\, T \to aTc,\, T \to B,\, B \to bB',\, D \to dD',\, B \to b,\, D \to d,\, S_1 \to S_1',\, S_1' \to Q_2 \,\}, \\ P_2 &= \{\, S_2 \to S_2,\, S_2 \to Q_1,\, B' \to B,\, D' \to D \,\}. \end{aligned}$$

Das PC-Grammatiksystem Γ_1 arbeitet im Rücksetzmodus wie folgt. Die erste Komponente erzeugt die as und cs sowie je genau ein b und ein d und möglicherweise jeweils einen Platzhalter für weitere bs und ds. Die zweite Komponente wartet zunächst ab und fragt schließlich die erste Komponente. Wenn dies zu früh geschieht, erhält sie ein „unbekanntes“ Nichtterminal, kann also höchstens ein Nichtterminal ersetzen und blockiert dann, da die erste Komponente noch nicht soweit ist, um sich das Wort wieder zu holen. Falls die zweite Komponente nicht rechtzeitig fragt, blockiert die erste. Das System blockiert auch dann, wenn die erste Komponente zum Fragezeitpunkt nur ein Nichtterminal (B' oder D') enthält. Das kann zwar in der zweiten Komponente abgeleitet werden, aber die zweite Komponente blockiert nach einem Schritt, da keine Regel mehr anwendbar ist und die erste Komponente noch nicht fragen kann. Also müssen, wenn die zweite Komponente fragt, je genau ein B' und D' in der ersten Komponente vorkommen. Diese beiden Nichtterminale werden in der zweiten Komponente abgeleitet. Danach holt sich die erste Komponente dieses Wort, leitet die beiden Nichtterminale ab (erzeugt dabei jeweils ein Terminal) und muss von der zweiten Komponente wieder zur rechten Zeit (nach zwei Schritten) gefragt werden. Durch dieses Verfahren ist sichergestellt, dass die bs und ds gleich häufig erzeugt werden. Dieser Vorgang wiederholt sich, bis die erste Komponente Terminale erzeugt.

Es sei Γ_2 folgendes kontextfreie, zentralistische PC-Grammatiksystem:

$$\Gamma_2 = (N, \{\, a, b, c, d \,\}, \{\, Q_1, Q_2, Q_3 \,\}, (P_1, S_1), (P_2, S_2), (P_3, S_3))$$

mit

$$\begin{aligned}
N &= \{\, S_1, S_2, S_3, S_2', S_3', Z_1, Z_2, Z_3, X, X', Y, Y', T, T', U, U', B, D, \bar{B}, \bar{D} \,\}, \\
P_1 &= \{\, S_1 \to Z_1,\, Z_1 \to Z_2,\, Z_2 \to Z_3,\, Z_3 \to Z_2,\, Z_3 \to Q_2,\, B \to Q_2,\, D \to Q_3,\, \bar{B} \to b,\, \bar{D} \to d \,\}, \\
P_2 &= \{\, S_2 \to S_2',\, S_2' \to bB,\, S_2 \to \bar{B},\, S_2 \to X,\, X \to X',\, X' \to aUc\bar{D},\, U \to \bar{B},\, U \to U',\, U' \to aUc \,\} \\
&\quad \cup \{\, S_2 \to Y,\, Y \to Y',\, Y' \to aTcD,\, T \to B,\, T \to T',\, T' \to aTc \,\}, \\
P_3 &= \{\, S_3 \to X,\, S_3 \to S_3',\, X \to S_3',\, S_3' \to X,\, S_3' \to dD,\, S_3 \to Y,\, Y \to Y',\, Y' \to \bar{D} \,\}.
\end{aligned}$$

Im Folgenden wird die Arbeitsweise des *CPC*$_3$*CF*-Systems Γ_2 beschrieben.

Die zweite Komponente erzeugt die as und cs (gleichzeitig) oder die bs. Die ds werden von der dritten Komponente geliefert (mit Ausnahme des letzten, welches in der ersten Komponente entsteht). Die erste Komponente sorgt dafür, dass zum einen zuerst die as und cs entstehen und zum anderen danach gleich viele bs wie ds erzeugt werden.

Da die erste Komponente mindestens vier Schritte bis zum ersten Fragen ausführt, muss sich die zweite Komponente zu Beginn für die Regel $S_2 \to X$ oder $S_2 \to Y$ entscheiden, da sie sonst nach spätestens zwei Schritten blockiert. Die erste Komponente fragt die zweite nach $4+2k$ Schritten ($k \geq 0$). Nach $4+2k$ Schritten beinhaltet die zweite Komponente das Wort $a^{k+1}\bar{\beta}c^{k+1}\bar{D}$ mit $\bar{\beta} \in \{\, U', \bar{B} \,\}$ oder $a^{k+1}\beta c^{k+1}D$ mit $\beta \in \{\, T', B \,\}$. Falls $\bar{\beta} = U'$ bzw. $\beta = T'$ gilt, gelangt U' bzw. T' in die erste Komponente und bleibt für immer dort (es kann nicht abgeleitet werden und wird auch nicht „weggefragt“). So wird kein Terminalwort erreicht. Folglich muss in der zweiten Komponente die Regel $U \to \bar{B}$ bzw. $T \to B$ genau dann angewendet werden, wenn in der ersten Komponente Z_3 zu Q_2 abgeleitet wird. Danach ist das Wort in der ersten Komponente $a^{k+1}\bar{B}c^{k+1}\bar{D}$ bzw. $a^{k+1}Bc^{k+1}D$ und das in der zweiten wieder S_2. Handelt es sich um $a^{k+1}\bar{B}c^{k+1}\bar{D}$, so wird es in zwei weiteren Schritten zu $a^{k+1}bc^{k+1}d$. Wird also zu Beginn $S_2 \to X$ angewendet, entstehen ausschließlich Wörter der Form $a^n bc^n d$ mit $n \geq 1$. Wird zu Beginn $S_2 \to Y$ angewendet, so beinhalten nach $4+2k$ Ersetzungsschritten und einem Kommunikationsschritt die erste Komponente das Wort $a^{k+1}Bc^{k+1}D$ und die zweite S_2. Auf das Wort $a^{k+1}Bc^{k+1}D$ muss zuerst die Regel $D \to Q_3$ und anschließend die Regel $B \to Q_2$ angewendet werden, da in sonst kein terminales Wort entsteht.

Der „Trick“ bei diesem PC-Grammatiksystem ist im Wesentlichen, dass die eine Anfrage nur in geradzahligen Ersetzungsschritten und die andere Anfrage nur in ungeradzahligen Ersetzungsschritten Erfolg hat. Dadurch wird ein abwechselndes Erzeugen von bs und ds erzwungen.

Nun wird ein kontextfreies, zentralistisches PC-Grammatiksystem mit zwei Komponenten angegeben, das im Fortsetzmodus die Sprache K_2 erzeugt. Es gelte

$$\Gamma_3 = (\{\, S_1, S_2, S_1', S_2', B, B', D, D', \bar{T}, \bar{B}, E \,\}, \{\, a, b, c, d \,\}, \{\, Q_1, Q_2 \,\}, (P_1, S_1), (P_2, S_2))$$

mit

$$P_1 = \{\, S_1 \to aQ_2cd, S_1 \to aQ_2cD, S_1 \to S_1', S_1' \to S_1, D \to dD', D' \to D, D' \to d, \bar{T} \to Q_2, \bar{T} \to aQ_2c, \bar{B} \to b \,\},$$
$$P_2 = \{\, S_2 \to \bar{T},\, S_2 \to S_2',\, S_2' \to aS_2c,\, \bar{T} \to \bar{B},\, \bar{B} \to E,\, \bar{T} \to bB,\, B \to B',\, B' \to bB,\, B' \to \bar{B} \,\}.$$

Auch dieses System nutzt die Gerad- und Ungeradzahligkeit der Ersetzungsschritte aus und erzwingt so eine kontrollierte Kommunikation.

Die zweite Komponente erzeugt paarweise as und cs und ab einem gewissen Zeitpunkt bs. Die erste Komponente wartet zunächst ab, bis die zweite Komponente anfängt, bs zu bilden, und fängt dann ihrerseits an, die ds zu generieren. Die erste Komponente muss damit rechtzeitig aufhören, damit sie ihre Nichtterminale noch abbauen kann, bevor die zweite Komponente ihre Arbeit einstellt.

Die erste Komponente fragt zum ersten Mal in einem ungeradzahligen Schritt. Von den Nichtterminalen, die nach ungeradzahlig vielen Schritten vorliegen können, ist nur $\bar{T}$ in der ersten Komponente ableitbar. Zum zweiten Mal muss die erste Komponente genau dann fragen, wenn in der zweiten Komponente $\bar{B}$ vorkommt. Bis dahin erzeugen die erste Komponente ds und die zweite bs. Nach dem zweiten Fragen kann die zweite Komponente nur noch einen Ersetzungsschritt ausführen. Diese Zeit benötigt auch die erste Komponente, um das gerade erhaltene Nichtterminal $\bar{B}$ durch b zu ersetzen. Danach muss ihr Wort terminal sein, denn die zweite Komponente verhindert ein Weiterarbeiten. Folglich muss die erste Komponente das Nichtterminal D' rechtzeitig zu d verwandeln. Da das System im Fortsetzmodus arbeitet, erhält die erste Komponente die as und cs auf beide Fragen. Damit auch eine ungerade Anzahl von a-c-Paaren möglich ist und mindestens ein solches Paar auftritt, erzeugt die erste Komponente selbst eines.

3 Zusammenfassung

In der vorliegenden Arbeit wurden drei kontextfreie parallel kommunizierende Grammatiksysteme angegeben, die die nicht-kontextfreie Sprache $K_2 = \{\, a^n b^m c^n d^m \mid m \geq 1,\ n \geq 1 \,\}$ erzeugen, aber weniger Komponenten haben als bisher veröffentlichte Systeme. Für den Fortsetzmodus wurde die Komponentenanzahl von fünf auf zwei reduziert, für den Rücksetzmodus von drei auf zwei. In diesen beiden Fällen ist das Ergebnis optimal. Für den Rücksetzmodus wurde die Komponentenanzahl bei zentralistischen PC-Grammatiksystemen von vier auf drei verkleinert.

Literatur

[1] A. CHIŢU, PC Grammar Systems Versus Some Non-Context-Free Constructions from Natural and Artificial Languages. In: G. PAUN, A. SALOMAA (eds.), *New Trends in Formal Languages*. Lecture Notes in Computer Science 1218, Springer, 1997, 278–287.

[2] E. CSUHAJ-VARJÚ, J. DASSOW, J. KELEMEN, G. PĂUN, *Grammar Systems: A Grammatical Approach to Distribution and Cooperation*. Topics in Computer Science 5, Gordon and Breach Science Publishers, 1994.

[3] J. DASSOW, G. PĂUN, G. ROZENBERG, Grammar systems. In: G. ROZENBERG, A. SALOMAA (eds.), *Handbook of Formal Languages*. Springer, 1997, 155–213.

[4] J. DASSOW, B. TRUTHE, On the degree complexity of special non-context-free languages with respect to PC grammar systems. In: H. LEUNG, G. PIGHIZZINI (eds.), *8th International Workshop on Descriptional Complexity of Formal Systems, DCFS 2006, Las Cruces, NM, USA, June 21–23, 2006, Proceedings*. NMSU-CS-2006-001, 2006, 241–249.