ON GENERATIVE POWER OF SYNCHRONOUS GRAMMARS WITH LINKED RULES

Petr Horáček
Doctoral Degree Programme (3), FIT BUT
E-mail: xhorac06@stud.fit.vutbr.cz

Supervised by: Alexander Meduna
E-mail: meduna@fit.vutbr.cz

Abstract: This paper discusses formal models for translation which use the principle of synchronization. It contains definitions of synchronous grammars based on linked rules instead of nonterminals, extending the principle from context-free grammars to models with regulated rewriting, such as matrix grammar and scattered context grammar. The main part presents new results regarding the generative power of such synchronous grammars.

Keywords: synchronous grammars, regulated rewriting, generative power

1 INTRODUCTION

In modern formal language theory (FLT), there is a number of well-known models that can be used in formal description of languages. However, there are tasks, such as natural language translation, where we also want to describe transformations between languages. While we can have separate formal models for each individual language and another formal mechanism linking them, there are also models that allow us to describe the translation directly.

There is a generalization of context-free grammar (CFG) called synchronous CFG (see [2]; this principle is also known as syntax-directed transduction grammar [5] or syntax-directed translation scheme [1]). Informally, the synchronous CFG is a modification of CFG where every rule has two right-hand sides. Thus, the synchronous CFG generates a pair of sentences in one derivation, rather than a single sentence.

In [4], we have proposed synchronization based on linked rules instead of nonterminals, and extended the principle to models with regulated rewriting, introducing the synronous matrix grammar and the synchronous scattered context grammar. In this paper, we discuss some theoretical properties of the proposed models, namely their generative power.

2 PRELIMINARIES

We assume that the reader is familiar with the basic aspects of modern FLT (see [8], [6]). Further information about matrix grammars and scattered context grammars can be found in [3] and [7], respectively.

Definition 1 (Context-free grammar)**.** *A* context-free grammar *(CFG) G is a quadruple $G = (N, T, P, S)$, where N is a finite set of nonterminals, T is a finite set of terminals, $N \cap T = \emptyset$, $P \subset N \times (N \cup T)^*$ is a finite set of rules, $(u, v) \in P$ is written as $u \to v$, and $S \in N$ is the start symbol.*

Definition 2 (Derivation)**.** *Let G be a CFG. Let $u, v \in (N \cup T)^*$ and $p = A \to x \in P$. Then, we say that uAv* directly derives *uxv according to p in G, written as $uAv \Rightarrow_G uXv\,[p]$ or simply $uAv \Rightarrow uxv$. We further define $\Rightarrow^+$ as the transitive closure and $\Rightarrow^*$ as the transitive and reflexive closure of $\Rightarrow$.*

Definition 3 (Generated language)**.** *Let G be a CFG. The* language generated by G*, denoted by $L(G)$, is defined as $L(G) = \{w\colon w \in T^*, S \Rightarrow^* w\}$*

Definition 4 (Matrix grammar)**.** *A* matrix grammar *H is a pair $H = (G,M)$, where $G = (N,T,P,S)$ is a CFG and M is a finite language over P ($M \subset P^*$) – a sentence of this language is called a* matrix.

Definition 5 (Derivation in matrix grammar)**.** *Let $H = (G,M)$ be a matrix grammar, $G = (N,T,P,S)$. Then, for $u,v \in (N \cup T)^*$, $m \in M$ we define $u \Rightarrow v\,[m]$ in H, if there are strings $x_0,\ldots,x_n$ such that $u = x_0$, $v = x_n$ and $x_0 \Rightarrow x_1\,[p_1] \Rightarrow x_2\,[p_2] \Rightarrow \ldots \Rightarrow x_n\,[p_n]$ in G, and $m = p_1 \ldots p_n$.*

Definition 6 (Scattered context grammar)**.** *A* scattered context grammar *(SCG) G is a quadruple $G = (N,T,P,S)$, where N is a finite set of nonterminals, T is a finite set of terminals, $N \cap T = \emptyset$, P is a finite set of rules of the form $(A_1,\ldots,A_n) \to (x_1,\ldots,x_n)$, where $A_1,\ldots,A_n \in N$, $x_1,\ldots,x_n \in (N \cup T)^*$, and $S \in N$ is the start symbol.*

Definition 7 (Derivation in SCG)**.** *Let $G = (N,T,P,S)$ be an SCG. For $u,v \in (N \cup T)^*$, $p \in P$ we define $u \Rightarrow v\,[p]$, if there is a factorization of $u = u_1A_1 \ldots u_nA_nu_{n+1}$, $v = u_1x_1 \ldots u_nx_nu_{n+1}$ such that $p = (A_1,\ldots,A_n) \to (x_1,\ldots,x_n)$ and $u_i \in (N \cup T)^*$ for $1 \leq i \leq n$.*

3 SYNCHRONIZATION AND REGULATED REWRITING

This section contains revised definitions from [4]. First, we define rule-based synchronization for CFGs.

Definition 8 (Rule-synchronized CFG)**.** *A* rule-synchronized CFG *(RSCFG) H is a 5-tuple $H = (G_I, G_O, \Psi, \varphi_I, \varphi_O)$, where $G_I = (N_I, T_I, P_I, S_I)$ and $G_O = (N_O, T_O, P_O, S_O)$ are CFGs, Ψ is a set of* rule labels*, and φ_I is a function from Ψ to P_I and φ_O is a function from Ψ to P_O.*

We use the following notation (presented for input grammar G_I, analogous for output grammar G_O):

$p : A_I \to x_I$ where $p \in \Psi, A_I \to x_I \in P_I$	$\varphi_I(p) = A_I \to x_I$
$x_I \Rightarrow_{G_I} y_I\,[p]$ where $x_I, y_I \in (N \cup T)^*, p \in \Psi$	derivation step in G_I applying rule $\varphi_I(p)$
$x_I \Rightarrow^n_{G_I} y_I\,[p_1 \ldots p_n]$ where $x_I, y_I \in (N \cup T)^*, p_i \in \Psi$ for $1 \leq i \leq n$	derivation in G_I applying rules $\varphi_I(p_1) \ldots \varphi_I(p_n)$

Definition 9 (Translation in RSCFG)**.** *Let $H = (G_I, G_O, \Psi, \varphi_I, \varphi_O)$ be a RSCFG.* Translation $T(H)$ *is the set of pairs of sentences, which is defined as $T(H) = \{(w_I, w_O)\colon w_I \in T_I^*, w_O \in T_O^*, S_I \Rightarrow^*_{G_I} w_I\,[\alpha], S_O \Rightarrow^*_{G_O} w_O\,[\alpha], \alpha \in \Psi^*\}$.*

In [4], we considered RSCFG only as a special case of synchronous CFG. However, there is actually a significant difference. While the latter does not bring any increase in generative power over CFG, RSCFG does, as we will see in the next section.

To define synchronization for SCGs, we simply replace context-free rules with scattered context rules.

Definition 10 (Synchronous SCG)**.** *A* synchronous SCG *(SSCG) H is a 5-tuple $H = (G_I, G_O, \Psi, \varphi_I, \varphi_O)$, where $G_I = (N_I, T_I, P_I, S_I)$ and $G_O = (N_O, T_O, P_O, S_O)$ are SCGs, Ψ is a set of* rule labels*, and φ_I is a function from Ψ to P_I and φ_O is a function from Ψ to P_O.*

Definition 11 (Translation in SSCG)**.** *Let $H = (G_I, G_O, \Psi, \varphi_I, \varphi_O)$ be a SSCG.* Translation $T(H)$ *is the set of pairs of sentences, which is defined as $T(H) = \{(w_I, w_O)\colon w_I \in T_I^*, w_O \in T_O^*, S_I \Rightarrow^*_{G_I} w_I\,[\alpha], S_O \Rightarrow^*_{G_O} w_O\,[\alpha], \alpha \in \Psi^*\}$.*

In the case of matrix grammars, we link whole matrices, rather than individual rules.

Definition 12 (Synchronous matrix grammar)**.** *A* synchronous matrix grammar *(SMAT) H is a 7-tuple $H = (G_I, M_I, G_O, M_O, \Psi, \varphi_I, \varphi_O)$, where (G_I, M_I) and (G_O, M_O) are matrix grammars, Ψ is a set of* matrix labels*, and φ_I is a function from Ψ to M_I and φ_O is a function from Ψ to M_O.*

Definition 13 (Translation in SMAT)**.** *Let $H = (G_I, M_I, G_O, M_O, \Psi, \varphi_I, \varphi_O)$ be a SMAT.* Translation *$T(H)$ is the set of pairs of sentences, which is defined as $T(H) = \{(w_I, w_O) \colon w_I \in T_I^*, w_O \in T_O^*, S_I \Rightarrow^*_{(G_I,M_I)} w_I\,[\alpha], S_O \Rightarrow^*_{(G_O,M_O)} w_O\,[\alpha], \alpha \in \Psi^*\}$.*

4 GENERATIVE POWER OF SYNCHRONOUS GRAMMARS

Synchronous grammars generate pairs of sentences. To be able to compare their generative power with well-known models such as CFGs, we can consider their input or output language.

Definition 14 (Input and output language)**.** *Let H be a synchronous grammar. Then, the input language and the output language of H, denoted by, respectively, $L_I(H)$ and $L_O(H)$, are defined as follows: $L_I(H) = \{w_I \in T_I^* \colon \exists w_O \colon (w_I, w_O) \in T(H)\}$, $L_O(H) = \{w_O \in T_O^* \colon \exists w_I \colon (w_I, w_O) \in T(H)\}$.*

Example 1. Consider a RSCFG $H = (G_I, G_O, \Psi, \varphi_I, \varphi_O)$ with the following rules (G_I on the left, G_O on the right, nonterminals are in capitals, linked rules share the same label):

$$
\begin{array}{rlcl@{\qquad\qquad}rlcl}
1: & S_I & \rightarrow & ABC & 1: & S_O & \rightarrow & A \\
2: & A & \rightarrow & aA & 2: & A & \rightarrow & B \\
3: & B & \rightarrow & bB & 3: & B & \rightarrow & C \\
4: & C & \rightarrow & cC & 4: & C & \rightarrow & A \\
5: & A & \rightarrow & \varepsilon & 5: & A & \rightarrow & B' \\
6: & B & \rightarrow & \varepsilon & 6: & B' & \rightarrow & C' \\
7: & C & \rightarrow & \varepsilon & 7: & C' & \rightarrow & \varepsilon
\end{array}
$$

Derivation example:

$$
\begin{array}{rcll@{\qquad\qquad}rcll}
S_I & \Rightarrow & ABC & [1] & S_O & \Rightarrow & A & [1] \\
 & \Rightarrow & aABC & [2] & & \Rightarrow & B & [2] \\
 & \Rightarrow & aAbBC & [3] & & \Rightarrow & C & [3] \\
 & \Rightarrow & aAbBcC & [4] & & \Rightarrow & A & [4] \\
 & \Rightarrow & aaAbBcC & [2] & & \Rightarrow & B & [2] \\
 & \Rightarrow & aaAbbBcC & [3] & & \Rightarrow & C & [3] \\
 & \Rightarrow & aaAbbBccC & [4] & & \Rightarrow & A & [4] \\
 & \Rightarrow & aabbBccC & [5] & & \Rightarrow & B' & [5] \\
 & \Rightarrow & aabbccC & [6] & & \Rightarrow & C' & [6] \\
 & \Rightarrow & aabbcc & [7] & & \Rightarrow & \varepsilon & [7]
\end{array}
$$

We can easily see that $L_I(H) = \{a^n b^n c^n \colon n \geq 0\}$, which is well known not to be a context-free language. This shows that RSCFG is stronger than (synchronous) CFG (strictly speaking, to make this claim, we also have to show that every context-free language can be generated by a RSCFG, but that is evident from the definition). Where exactly do synchronous grammars (rule-synchronized) stand in terms of generative power?

Let $\mathscr{L}(RSCFG)$, $\mathscr{L}(SMAT)$, and $\mathscr{L}(SSCG)$ denote the class of languages generated by RSCFGs, SMATs, and SSCGs, respectively, as their input language (note that the results presented below would be the same if we considered the output language instead).

Theorem 1. $\mathscr{L}(RSCFG) = \mathscr{L}(MAT)$

Proof. First, we prove that $\mathscr{L}(RSCFG) \subseteq \mathscr{L}(MAT)$. For every RSCFG $H = (G_I, G_O, \Psi, \varphi_I, \varphi_O)$, where $G_I = (N_I, T_I, P_I, S_I)$, $G_O = (N_O, T_O, P_O, S_O)$, we can construct a matrix grammar $H' = (G, M)$, where $G = (N, T, P, S)$, such that $L(H') = L_I(H)$, as follows. Without loss of generality, assume $N_I \cap N_O = \emptyset, S \notin N_I \cup N_O$. Set $N = N_I \cup N_O \cup \{S\}$, $T = T_I$, $P = \{S \to S_I S_O\}$, $M = \{S \to S_I S_O\}$. For every label $p \in \Psi$, add rules p_I, p_O to P, add matrix $p_I p_O$ to M, where $p_I = \varphi_I(p)$ and $p_O = A_0 \to A_1 \ldots A_n$ such that $\varphi_O(p) = A_0 \to x_0 A_1 x_1 \ldots A_n x_n$, where $A_i \in N_O$, $x_i \in T_O^*$ for $0 \leq i \leq n$ (this removes all terminals from the right-hand side of the rule).

H' simulates the principle of linked rules in H by matrices. That is, for every pair of rules (p_I, p_O) such that $p_I = \varphi_I(p), p_O = \varphi_O(p)$ for some $p \in \Psi$ in H, there is a matrix $m = p_I p'_O$ in H', where p'_O is equal to p_O with all terminals removed (formally defined above). If, in H, $x_I \Rightarrow y_I\,[p]$ in G_I and $x_O \Rightarrow y_O\,[p]$ in G_O, then there is a derivation step $x_I x'_O \Rightarrow y_I y'_O\,[m]$ in H', where x'_O and y'_O are equal to x_O and y_O with all terminals removed, respectively. Note that since the rules are context-free, the presence (or absence) of terminals in a sentential form does not affect which rules we can apply. Furthermore, because the nonterminal sets N_I and N_O are disjoint, the first rule in each matrix, simulating the derivation step in G_I, and the second rule, simulating the derivation step in G_O, always rewrite distinct parts of the current sentential form.

Now we have to show that $\mathscr{L}(MAT) \subseteq \mathscr{L}(RSCFG)$ holds. For every matrix grammar $H = (G, M)$, where $G = (N, T, P, S)$, we can construct a RSCFG $H' = (G_I, G_O, \Psi, \varphi_I, \varphi_O)$, where $G_I = (N_I, T_I, P_I, S_I)$, $G_O = (N_O, T_O, P_O, S_O)$, such that $L_I(H') = L(H)$, as follows. Without loss of generality, assume $N \cap \{S_I, S_O, X\} = \emptyset$. Set $N_I = N \cup \{S_I, X\}$, $T_I = T$, $P_I = \{S_I \to SX, X \to \varepsilon\}$, $N_O = \{S_O, X\}$, $T_O = \emptyset$, $P_O = \{S_O \to X, X \to \varepsilon\}$. Set $\Psi = \{0, 1\}$, $\varphi_I = \emptyset$, $\varphi_O = \emptyset$, $\varphi_I(0) = S_I \to SX$, $\varphi_O(0) = S_O \to X$, $\varphi_I(1) = X \to \varepsilon$, $\varphi_O(1) = X \to \varepsilon$. For every matrix $m = p \in M$, add rule p to P_I, $X \to X$ to P_O, add label m to Ψ, and set $\varphi_I(m) = p$, $\varphi_O(m) = X \to X$. For every matrix $m = p_1 \ldots p_n \in M$, where $n > 1$, add rules $p_1, \ldots, p_n$ to P_I, add nonterminals $X_m^1, \ldots, X_m^{n-1}$ to N_O, add rules $X \to X_m^1, X_m^1 \to X_m^2, \ldots, X_m^{n-2} \to X_m^{n-1}, X_m^{n-1} \to X$ to P_O, add labels $m_1, \ldots, m_n$ to Ψ, and set $\varphi_I(m_1) = p_1$, $\varphi_O(m_1) = X \to X_m^1$, $\varphi_I(m_i) = p_i$, $\varphi_O(m_i) = X_m^{i-1} \to X_m^i$ for $1 < i < n$, and $\varphi_I(m_n) = p_n$, $\varphi_O(m_n) = X_m^{n-1} \to X$.

H' simulates matrices in H by derivation in G_O. That is, if $x \Rightarrow y\,[m]$ in H, where $m = p_1 \ldots p_n$, then there is a sequence of derivation steps $X \Rightarrow X_m^1\,[m_1] \Rightarrow X_m^2\,[m_2] \Rightarrow \ldots \Rightarrow X_m^{n-2}\,[m_{n-2}] \Rightarrow X_m^{n-1}\,[m_{n-1}] \Rightarrow X\,[m_n]$ in G_O and $\varphi_I(m_i) = p_i$ for $1 \leq i \leq n$. Now observe that in G_O constructed by the above algorithm, every nonterminal except X can only appear as the left-hand side of at most one rule. This means that after rewriting X to X_m^1, the only way for the derivation to proceed is the above sequence, and the entire matrix is simulated. □

Note that G_O constructed by the above algorithm is not only context-free, but also regular.

Theorem 2. $\mathscr{L}(SMAT) = \mathscr{L}(MAT)$

Proof. The inclusion $\mathscr{L}(MAT) \subseteq \mathscr{L}(SMAT)$ follows from definition. It only remains to prove that $\mathscr{L}(SMAT) \subseteq \mathscr{L}(MAT)$. For every SMAT $H = (G_I, M_I, G_O, M_O, \Psi, \varphi_I, \varphi_O)$, where $G_I = (N_I, T_I, P_I, S_I)$, $G_O = (N_O, T_O, P_O, S_O)$, we can construct a matrix grammar $H' = (G, M)$, where $G = (N, T, P, S)$, such that $L(H') = L_I(H)$, as follows. Without loss of generality, assume $N_I \cap N_O = \emptyset$, $S \notin N_I \cup N_O$. Set $N = N_I \cup N_O \cup \{S\}$, $T = T_I$, $P = \{S \to S_I S_O\}$, $M = \{S \to S_I S_O\}$. For every label $p \in \Psi$, add rules $p_I^1, \ldots, p_I^n$, $p_O^1, \ldots, p_O^m$ to P, add matrix $p_I^1 \ldots p_I^n p_O^1 \ldots p_O^m$ to M, where $p_I^1 \ldots p_I^n = \varphi_I(p)$ and for $1 \leq j \leq m$, $p_O^j = A_0^j \to A_1^j \ldots A_n^j$ such that $\varphi_O(p)[j] = A_0^j \to x_0^j A_1^j x_1^j \ldots A_n^j x_n^j$, where $A_i^j \in N_O$, $x_i^j \in T_O^*$ for $0 \leq i \leq n$ (again, this removes all terminals from the right-hand side of the rules; $m[i]$ denotes the i-th rule in matrix m).

H' simulates H by combining the rules of each two linked matrices in H into a single matrix in H'. That is, for every pair of matrices (m_I, m_O) such that $m_I = \varphi_I(p), m_O = \varphi_O(p)$ for some $p \in \Psi$ in H, there is a matrix $m = m_I m'_O$ in H', where m'_O is equal to m_O with all terminals removed (formally defined above). If, in H, $x_I \Rightarrow y_I\,[p]$ in G_I and $x_O \Rightarrow y_O\,[p]$ in G_O, then there is a derivation step $x_I x'_O \Rightarrow y_I y'_O\,[m]$ in H', where x'_O and y'_O are equal to x_O and y_O with all terminals removed, respectively. Note that since the rules are context-free, the presence (or absence) of terminals in a sentential form does not affect which rules we can apply. Furthermore, because the nonterminal sets N_I and N_O are disjoint, the first part of each matrix, simulating the derivation step in G_I, and the second part, simulating the derivation step in G_O, always rewrite distinct parts of the current sentential form. □

Theorem 3. $\mathscr{L}(SSCG) = RE$

Proof. Clearly, $\mathscr{L}(SSCG) \subseteq RE$ holds. From definition, it follows that $\mathscr{L}(SCG) \subseteq \mathscr{L}(SSCG)$. Because $\mathscr{L}(SCG) = RE$, $RE \subseteq \mathscr{L}(SSCG)$ must also hold. □

5 CONCLUSION

In this paper, we have discussed the generative power of synchronous grammars based on linked rules. To summarize the main results, we have obtained the following hierarchy of language classes:

$$CF \subset \mathscr{L}(RSCFG) = \mathscr{L}(MAT) = \mathscr{L}(SMAT) \subset \mathscr{L}(SSCG) = RE$$

Further research prospects in this direction include study of the properties of synchronous grammars with additional restrictions, such as leftmost derivation. We can also consider synchronization of other well-known formal models, using the same principle.

ACKNOWLEDGEMENT

This work was partially supported by the research plan MSM0021630528, the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), and the BUT FIT grant FIT-S-11-2.

REFERENCES

[1] Aho, A.V., Ullman, J.D.: Syntax directed translations and the pushdown assembler. In *Journal of Computer and System Sciences Volume 3*, 1969, p. 37–56.

[2] Chiang, D.: An Introduction to Synchronous Grammars [online]. Part of a tutorial given at *44th Annual Meeting of the Association for Computational Linguistics*, 2006,
`http://www.isi.edu/~chiang/papers/synchtut.pdf`

[3] Dassow, J., Păun, Gh.: *Regulated Rewriting in Formal Language Theory*. Springer, 1989, ISBN 3-540-51414-7.

[4] Horáček, P., Meduna, A.: Regulated Rewriting in Natural Language Translation, In: *7th Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, Brno, CZ, 2011, ISBN 978-80-214-4305-1, p. 35–42

[5] Lewis, P.M., Stearns, R.E.: Syntax-directed transduction. In *Journal of the ACM Volume 15*, 1968, p. 465–488.

[6] Meduna, A.: *Automata and Languages: Theory and Applications*. Springer, 2005, ISBN 1-85233-074-0, 892 p.

[7] Meduna, A., Techet, J.: *Scattered Context Grammars and their Applications*. WIT Press, UK, 2010, ISBN 978-1-84564-426-0, 224 p.

[8] Rozenberg, G., Salomaa, A.: *Handbook of Formal Languages: Volume I*. Springer, 1997.