# Szilard Languages Associated to Parallel Communicating Grammar Systems

**Valeria Mihalache**

Turku Centre for Computer Science

TUCS Technical Reports

No 14, May 1996

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN-20520 Turku
Finland

http://www.tucs.abo.fi

May 1996

ISBN 951-650-756-5
ISSN 1239-1891

## Abstract

We associate Szilard and extended Szilard languages to parallel communicating grammar systems, starting from labels associated to productions of the components of the systems. We study families of such languages associated to various classes of parallel communicating grammar systems, comparing them to each other, to families of languages generated by regulated rewriting devices or by parallel communicating grammar systems.



**TUCS Research Group**
"Mathematical Structures of Computer Science"

# 1 Introduction

The challenging grammar systems have been intensively dealt with since they were first considered, cooperating distributed grammar systems in [2] (a former variant of them can be also found in [6]; a particular case appears also in [1]) and parallel communicating grammar systems in [8]. Roughly speaking, a grammar system consists of several usual grammars working together. In a cooperating distributed grammar system (CD grammar system, for short) the components work in turn, on the same sentential form, only one being active at a given moment, while in a parallel communicating grammar system (PC grammar system, for short) the components work simultaneously, in a synchronized manner, each having its own sentential form and cooperating to the others by communication, which is done by request. Motivations for the rapid development of grammar systems, as well as the most of the results concerning them (known at the middle of 1992) can be found in [3].

The present paper joins the investigations of PC grammar systems, by considering *Szilard* and *extended Szilard languages* associated to them. For CD grammar systems such a problem has already been approached, namely in [5], where Szilard and extended Szilard languages are associated to CD grammar systems, starting from labels associated to the components of these systems. Due to the fact that the components of the system work simultaneously in the parallel communicating case, we have considered more appropriate to define the Szilard and the extended Szilard languages for PC grammar systems starting from labels associated to the productions of the grammars, instead.

The study of Szilard languages for PC grammar systems is worth not only from the formal languages point of view (they have also been studied for other types of grammars, see [4], [9], [10]), but it also has motivations in the classroom model, which PC grammar system are a formal representation for. Briefly stated, the classroom model, which is a variant of the blackboard model in problem solving ([3]), defines the situation when pupils in a class have particular problems to solve, they use their notebooks for that, they may communicate in each step of problem solving or only in the situation when the leader of the classroom, who operates on the blackboard, requests this, and the global problem is solved through such cooperation on the blackboard. In some cases it might be of great interest to have more informations about the way the team (pupils classroom and the leader) has accomplished its job, and such informations can be inferred if provided a control of the way the team has worked. This leads naturally to control words describing terminal derivations, and therefore to Szilard languages.

We address in this paper Szilard and extended Szilard languages associated to PC grammar systems with regular, linear and context-free components, deriving in a synchronized manner. Some of the results we outline are similar to those known for other classes of Szilard languages or to relations known between certain classes of PC grammar systems.

## 2 Definitions and Examples

For basic notions of formal language theory, we refer to [10] and for grammar systems we refer to [3]. A few notations are specified in what follows.

For an alphabet $V$, $V^*$ denotes the free monoid generated by $V$; the empty string is denoted by $\lambda$, $|x|$ is the length of $x \in V^*$ and $|x|_U$ is the number of occurrences in $x \in V^*$ of symbols of $U \subseteq V$ (if $U = \{a\}$, then we denote $|x|_a$ instead of $|x|_{\{a\}}$). The classes of regular, linear, and context-free grammars are denoted by $REG, LIN, CF$ respectively; the corresponding families of languages are denoted by $\mathbf{L}(X), X \in \{REG, LIN, CF\}$ .

For a set of productions $P = \{x_1 \to y_1, x_2 \to y_2, \ldots, x_k \to y_k\}$, we represent the labels associated to the rules, in a one-to-one manner, by writing $P = \{p_1 : x_1 \to y_1, p_2 : x_2 \to y_2, \ldots, p_k : x_k \to y_k\}$, where for any $i, 1 \leq i \leq k$, $p_i$ is the label associated to the production $x_i \to y_i \in P$.

Let

$$\Gamma = (N, K, T, G_1, G_2, \ldots, G_n)$$

be a PC grammar system of type $X$ with $n$ components of type $Y$, in the notations of [3] (that is $N$ is a nonterminal alphabet, $T$ is a terminal alphabet, $K = \{Q_1, Q_2, \ldots, Q_n\}$, the sets $N, T, K$ being mutually disjoint, and $G_i = (N \cup K, T, P_i, S_i)$, for each $i, 1 \leq i \leq n$; denote $V_\Gamma = N \cup K \cup T$), where $X \in \{CPC, PC, NCPC, NPC\}$ (corresponding to a *centralized and returning, non-centralized and returning, centralized and non-returning, non-centralized and non-returning* PC grammar system, respectively) and $Y \in \{REG, LIN, CF\}$.

Let us consider each production of the system to be labelled and let $Lab(\Gamma)$ be the set of all such labels for $\Gamma$. We associate to each derivation in the system the sequence of productions used, as follows:

**Definition 1** *Let $\Gamma = (N, K, T, G_1, \ldots, G_n)$ be a PC grammar system. For two n-tuples $((x_1, \alpha_1), (x_2, \alpha_2), \ldots, (x_n, \alpha_n))$ and $((y_1, \beta_1), (y_2, \beta_2), \ldots, (y_n, \beta_n))$ with $x_i, y_i \in V_\Gamma^*, \alpha_i, \beta_i \in Lab(\Gamma)^*$, for each $i, 1 \leq i \leq n$, we write*

$$((x_1, \alpha_1), (x_2, \alpha_2), \ldots, (x_n, \alpha_n)) \Rightarrow ((y_1, \beta_1), (y_2, \beta_2), \ldots, (y_n, \beta_n))$$

*if one of the next two cases holds:*

- *either (i)* $|x_i|_K = 0, 1 \leq i \leq n$*; then for each* $i$, $1 \leq i \leq n$, *we have* $x_i \Rightarrow y_i$ *in the grammar* $G_i$, *by using the production labelled* $p_i$, *and then* $\beta_i = \alpha_i p_i$*; or we have* $x_i = y_i \in T^*$ *and then* $\beta_i = \alpha_i$*;*

- *or (ii) there is* $i$, $1 \leq i \leq n$, *such that* $|x_i|_K > 0$*; then for each such* $i$, *we write* $x_i = z_1 Q_{i_1} z_2 Q_{i_2} \ldots z_t Q_{i_t} z_{t+1}, t \geq 1$, *for* $z_j \in V_\Gamma^*, |z_j|_K = 0, 1 \leq j \leq t+1$*; if* $|x_{i_j}|_K = 0, 1 \leq j \leq t$, *then* $y_i = z_1 x_{i_1} z_2 x_{i_2} \ldots z_t x_{i_t} z_{t+1}$, $\beta_i = \alpha_i \alpha_{i_1} \alpha_{i_2} \ldots \alpha_{i_t}$ *[and* $y_{i_j} = S_{i_j}, \beta_{i_j} = \lambda, 1 \leq j \leq t$*]; when, for some* $j, 1 \leq j \leq t, |x_{i_j}|_K \neq 0$, *then* $y_i = x_i$, $\beta_i = \alpha_i$*; for all* $i$, $1 \leq i \leq n$, *for which* $y_i$ *is not specified above, we have* $y_i = x_i$,$\beta_i = \alpha_i$.

Just as in the definition of the derivation in a PC grammar system, point $(i)$ defines *rewriting steps*, while point $(ii)$ defines *communication steps*. The words written in brackets are erased in case of non-returning PC grammar systems.

**Definition 2** *The* Szilard language associated to a PC grammar system $\Gamma$ *as above is*

$$\begin{aligned} SZ(\Gamma) \;=\; & \{\gamma \mid ((S_1,\lambda),(S_2,\lambda),\ldots,(S_n,\lambda)) \Rightarrow^* ((x,\gamma),(z_2,\beta_2),\ldots,(z_n,\beta_n)), \\ & x \in T^*, z_i \in V_\Gamma^*, \gamma, \beta_i \in Lab(\Gamma)^*, 2 \leq i \leq n\}. \end{aligned}$$

*The* extended Szilard language associated to a PC grammar system $\Gamma$ *as above is*

$$\begin{aligned} ESZ(\Gamma) = & \{x\gamma \mid ((S_1,\lambda),(S_2,\lambda),\ldots,(S_n,\lambda)) \Rightarrow^* ((x,\gamma),(z_2,\beta_2),\ldots,(z_n,\beta_n)), \\ & x \in T^*, z_i \in V_\Gamma^*, \gamma, \beta_i \in Lab(\Gamma)^*, 2 \leq i \leq n\}. \end{aligned}$$

For a class $X$ of PC grammar systems, $X \in \{CPC, PC, NCPC, NPC\}$ and a family $Y$ of grammars, $Y \in \{REG, LIN, CF\}$, we denote by $\mathbf{SZ}(X_nY)$, $\mathbf{ESZ}(X_nY)$ the family of Szilard, respectively extended Szilard languages associated to PC grammar systems of type $X$ with at most $n$ components of type $Y$. As usual, when an arbitrary number of components is to be considered, we use $*$ instead of $n$.

**Example:** Let $\Gamma = (\{S_1, S_2\}, K, \{a,b\}, G_1, G_2)$, with

$$\begin{aligned} P_1 &= \{1 : S_1 \to aS_1, 2 : S_1 \to aQ_2, 3 : S_2 \to b\}; \\ P_2 &= \{4 : S_2 \to bS_2\}. \end{aligned}$$

A terminal derivation in $\Gamma$ proceeds as follows:

$((S_1,\lambda),(S_2,\lambda)) \Rightarrow^* ((a^kS_1, 1^k),(b^kS_2, 4^k)) \Rightarrow ((a^{k+1}Q_2, 1^k2),(b^{k+1}S_2, 4^{k+1}))$
$\Rightarrow ((a^{k+1}b^{k+1}S_2, 1^k24^{k+1}),(\alpha_1,\beta_1)) \Rightarrow ((a^{k+1}b^{k+2}, 1^k24^{k+1}3),(\alpha_2,\beta_2))$, where

either $(\alpha_1, \beta_1) = (S_2, \lambda)$ and $(\alpha_2, \beta_2) = (bS_2, 4)$,
or $(\alpha_1, \beta_1) = (b^{k+1}S_2, 4^{k+1})$ and $(\alpha_2, \beta_2) = (b^{k+2}S_2, 4^{k+2})$,

depending whether $\Gamma$ is a returning or a non-returning PC grammar system. Therefore,

$$\begin{aligned} SZ(\Gamma) &= \{1^k 2 4^{k+1} 3 \mid k \geq 0\}, \\ ESZ(\Gamma) &= \{a^{k+1} b^{k+2} 1^k 2 4^{k+1} 3 \mid k \geq 0\}. \end{aligned}$$

The following lemmas are direct consequences of the definitions of PC grammar systems and of Szilard and extended Szilard languages associated to PC grammar systems.

**Lemma 1** *For any PC grammar systems $\Gamma$, $SZ(\Gamma)$ is prefix-free.*

**Proof:** First, recall that a language is *prefix-free* if $x \in L$ and $xy \in L$ always implies $y = \lambda$. Let $\pi$ be in $SZ(\Gamma)$ for a PC grammar system $\Gamma$. It is not possible to continue the derivation after producing the string $x \in T^*$ which is generated by the derivation $\pi$, because there are no non-terminal, nor query symbols left in $x$. That is, $SZ(\Gamma)$ is prefix-free. $\square$

**Lemma 2** *For any class $Y$ of grammars, and for any $n \geq 1$,*

*(i)* $\mathbf{SZ}(CPC_nY) \subseteq \mathbf{SZ}(PC_nY)$ , $\mathbf{ESZ}(CPC_nY) \subseteq \mathbf{ESZ}(PC_nY)$ ;

*(ii)* $\mathbf{SZ}(NCPC_nY) \subseteq \mathbf{SZ}(NPC_nY)$, $\mathbf{ESZ}(NCPC_nY) \subseteq \mathbf{ESZ}(NPC_nY)$;

*(iii)* $\mathbf{SZ}(X_nY) \subseteq \mathbf{SZ}(X_{n+1}Y)$, $\mathbf{ESZ}(X_nY) \subseteq \mathbf{ESZ}(X_{n+1}Y)$, *for any* $X \in \{CPC, PC, NCPC, NPC\}$.

**Lemma 3** *For any class $X$ of PC grammar systems, for any $Y_1, Y_2 \in \{REG, LIN, CF\}$, such that $\mathbf{L}(Y_1) \subseteq \mathbf{L}(Y_2)$, and for any $n \geq 1$,*

$$\mathbf{SZ}(X_nY_1) \subseteq \mathbf{SZ}(X_nY_2),\ \mathbf{ESZ}(X_nY_1) \subseteq \mathbf{ESZ}(X_nY_2).$$

# 3 Regular PC Grammar Systems

We investigate relations between the Szilard languages associated to regular PC grammar and the languages generated by the same class of grammar systems. Also, Szilard and extended Szilard languages are compared to regular, context-free or finite index matrix languages. It is pointed out that similar results hold also for PC grammar systems with linear components.

**Theorem 1** *For any $X \in \{CPC, PC, NCPC, NPC\}$ and for any $n \geq 1$,*

*(i)* $\mathbf{SZ}(X_nREG) \subseteq \mathbf{L}(X_nREG)$;

*(ii)* $\mathbf{SZ}(X_nLIN) \subseteq \mathbf{L}(X_nLIN)$.

**Proof:** Let

$$\Gamma = (N, K, T, G_1, G_2, \ldots, G_n)$$

be a PC grammar system of type $X$ with regular components. Let

$$\Gamma' = (N', K, Lab(\Gamma), G'_1, G'_2, \ldots, G'_n),$$

where $N' = \{S'_1, S'_2, \ldots, S'_n\} \cup N$, and for each $i, 1 \leq i \leq n$, let

$$\begin{aligned} P'_i \;\; = \;\; & \{S'_i \to pX \mid p : S_i \to aX \in P_i, X \notin K\} \cup \{S'_i \to p \mid p : S_i \to a \in P_i\} \\ & \cup \{X \to pY \mid p : X \to aY \in P_i, X \notin K\} \cup \{X \to p \mid p : X \to a \in P_i\} \\ & \cup \{S'_i \to pQ_j \mid p : S_i \to aQ_j \in P_i, 1 \leq j \leq n\} \\ & \cup \{X \to pQ_j \mid p : X \to aQ_j \in P_i, 1 \leq j \leq n\}. \end{aligned}$$

Obviously $\Gamma'$ is of the same type as $\Gamma$ is and $L(\Gamma') = L(\Gamma)$. This proves point *(i)* of the theorem. Point *(ii)* can be proved in the same manner. □

We mention that a result similar to this theorem holds also for matrix grammars (see [4]).

The next two corollaries follow directly from the above theorem and from properties known from [3] or from [7].

**Corollary 1** *(i)* $\mathbf{SZ}(CPC_2REG) \subset \mathbf{L}(CF)$, *strict inclusion;*

*(ii)* $\mathbf{SZ}(PC_2REG) \subseteq \mathbf{L}(CF)$.

**Corollary 2** *(i) Each language in* $\mathbf{SZ}(CPC_*REG)$ *is a finite index matrix language;*

*(ii) Each language in* $\mathbf{SZ}(CPC_*LIN)$ *is semi-linear.*

**Theorem 2** *Let $\Gamma$ be a regular PC grammar system of type $X$, $X \in \{CPC$, $PC$, $NCPC$, $NPC\}$. Then there exists a homomorphism $h$ such that $L(\Gamma) = h(SZ(\Gamma))$.*

**Proof:** Let $h : Lab(\Gamma)^* \to T^*$ be the homomorphism defined by $h(p) = a$ for any label $p$ corresponding to a production $p : A \to aB$ or $p : A \to a, a \in T$. Obviously $L(\Gamma) = h(SZ(\Gamma))$. □

**Corollary 3** $\mathbf{SZ}(CPC_2REG) - \mathbf{L}(REG) \neq \emptyset$;
$\mathbf{ESZ}(CPC_1REG) - \mathbf{L}(CF) \neq \emptyset$.

**Corollary 4** $\mathbf{SZ}(NCPC_2REG) - \mathbf{L}(CF) \neq \emptyset$;
$\mathbf{ESZ}(NCPC_1REG) - \mathbf{L}(CF) \neq \emptyset$.

**Corollary 5** $\mathbf{ESZ}(CPC_2REG) - \mathbf{L}(PC_2REG) \neq \emptyset$.

**Theorem 3** $\mathbf{SZ}(CPC_3REG) - \mathbf{L}(CF) \neq \emptyset$.

**Proof:** The Szilard language associated to the returning PC grammar system

$$\Gamma = (\{S_1, S_2, S_3\}, K, \{a, b, c\}, G_1, G_2, G_3), \text{ with}$$

$$\begin{aligned} P_1 &= \{1 : S_1 \to aS_1, 2 : S_1 \to aQ_2, 3 : S_2 \to bQ_3, 4 : S_3 \to c\}, \\ P_2 &= \{5 : S_2 \to bS_2\}, \\ P_3 &= \{6 : S_3 \to cS_3\}. \end{aligned}$$

is

$$SZ(\Gamma) = \{1^k 2 5^{k+1} 3 6^{k+2} 4 \mid k \geq 0\},$$

which is not a context-free language. □

**Theorem 4** $\mathbf{SZ}(CPC_{n+1}REG) - \mathbf{L}(CPC_nREG) \neq \emptyset$.

**Proof:** Let us consider the returning centralized PC grammar system with $n+1$ regular components

$$\Gamma = (\{S_i \mid 1 \leq i \leq n+1\}, K, \{a_i \mid 1 \leq i \leq n+1\}, G_1, G_2, \ldots, G_{n+1}), \text{ with}$$

$$\begin{aligned} P_1 &= \{1 : S_1 \to a_1S_1, 2 : S_1 \to a_1Q_2, 3 : S_{n+1} \to a_{n+1}\} \\ &\quad \cup \{i+2 : S_i \to a_iQ_{i+1} \mid 2 \leq i \leq n\}, \end{aligned}$$

and for each $i, 2 \leq i \leq n+1$,

$$P_i = \{n + 1 + i : S_i \to a_iS_i\}.$$

One can see that

$$SZ(\Gamma) = \{1^k 2 (n+3)^{k+1} 4 (n+4)^{k+2} 5 (n+5)^{k+3} 6 \ldots (n+2)(2n+2)^{k+n} 3 \mid k \geq 0\}.$$

This language has $n+1$ pumping positions, and hence, according to Theorem 7.19 in [3], it is not in $\mathbf{L}(CPC_nREG)$. □

**Corollary 6** $\mathbf{SZ}(CPC_nREG) \subset \mathbf{SZ}(CPC_{n+1}REG)$, *strict inclusion.*

# 4 Context-Free PC Grammar Systems

For the context-free case, the families of Szilard and extended Szilard languages associated to centralized returning and non-returning PC grammar systems are compared to the family of languages generated by returning non-centralized PC grammar systems. They are also compared to context-free languages.

**Theorem 5** *For any* $n \geq 1$ *and for any* $X \in \{CPC, PC, NCPC, NPC\}$ $\mathbf{SZ}(X_nCF) - \mathbf{L}(CF) \neq \emptyset$; $\mathbf{ESZ}(X_nCF) - \mathbf{L}(CF) \neq \emptyset$.

**Proof:** This is a consequence of the fact that $\mathbf{SZ}(CF) - \mathbf{L}(CF) \neq \emptyset$ ([10]). □

**Theorem 6** *(i)* $\mathbf{ESZ}(CPC_*CF) \subseteq \mathbf{L}(PC_*CF)$;

*(ii)* $\mathbf{SZ}(CPC_*CF) \subseteq \mathbf{L}(PC_*CF)$.

**Proof:** Let
$$\Gamma = (N, K, T, G_1, G_2, \ldots, G_n)$$
be a returning centralized PC grammar system with $n$ context-free components ($n \geq 1$) and for each $i, 1 \leq i \leq n$, let $G_i = (N \cup K, T, P_i, S_i)$, $r_i = card(P_i)$ and $P_i = \{(i,k) : A_{ik} \to \alpha_{ik} \mid 1 \leq k \leq r_i\}$. Then $Lab(\Gamma) = \{(i,k) \mid 1 \leq i \leq n, 1 \leq k \leq r_i\}$.

Let $\overline{S_i}, S_i', \overline{X_i}, X_i', S_i^j, S_{ik}^j, Y_i^j, Y_{ik}^j, 1 \leq i \leq n, 1 \leq j \leq 2, 1 \leq k \leq r_i+1$, and $\overline{M}, M, Z, Z', S', S'', S_{a1}, S_{a2}, D_1, F_1, H_1, L_t, V_t, 0 \leq t \leq 4$, be new symbols, let $N'$ be the union of $N$ with the set of all these symbols and let

$P_s = \{S_s \to S_s, S_s \to Q_1^1 Q_1^2\} \cup \{X_i \to \lambda \mid 1 \leq i \leq n\} \cup \{M \to \lambda\}$.

For each $i, 1 \leq i \leq n$, let

$$\begin{aligned} P_i^1 &= \{S_i^1 \to S', S' \to \overline{S_i M}\} \cup \{S_i^1 \to D_1\} \cup \{D_1 \to Q_{ik}^1 \mid 1 \leq k \leq r_i + 1\} \\ &\quad \cup \{A \to A \mid A \in N \cup \{M, Z\}\} \cup \{S_i^1 \to Y_i^1, Y_i^1 \to Y_i^1\}; \\ P_i^2 &= \{S_i^2 \to S'', S'' \to \overline{X_i}\} \cup \{S_i^2 \to E_1\} \cup \{E_1 \to Q_{ik}^2 \mid 1 \leq k \leq r_i + 1\} \\ &\quad \cup \{X_i \to X_i\} \cup \{S_i^2 \to Y_i^2, Y_i^2 \to Y_i^2\}. \end{aligned}$$

For each $k, 1 \leq k \leq r_i$, let

$$\begin{aligned} P_{ik}^1 &= \{A_{ik} \to A_{ik}', A_{ik}' \to \widehat{\alpha_{ik}} \mid (i,k) : A_{ik} \to \alpha_{ik} \in P_i\} \\ &\quad \cup \{\overline{S_i} \to \widehat{\alpha_{ik}} \mid (i,k) : S_i \to \alpha_{ik} \in P_i\} \cup \{\overline{M} \to M\} \\ &\quad \cup \{A \to Z', Z' \to Z \mid A \in N \cup \{M\}\} \cup \{S_{ik}^1 \to F_1, F_1 \to Q_i^1\} \\ &\quad \cup \{S_{ik}^1 \to Y_{ik}^1, F_1 \to Y_{ik}^1, \overline{S_i} \to Y_{ik}^1, Y_{ik}^1 \to Y_{ik}^1\}; \\ P_{ik}^2 &= \{X_i \to X_i', \overline{X_i} \to X_i', X_i' \to (i,k)\widehat{\beta_{ik}} X_i \mid (i,k) : S_i \to \alpha_{ik} \in P_i\} \\ &\quad \cup \{S_{ik}^2 \to H_1, H_1 \to Q_i^2\} \cup \{S_{ik}^2 \to Y_{ik}^2, H_1 \to Y_{ik}^2, \overline{X_i} \to Y_{ik}^2, Y_{ik}^2 \to Y_{ik}^2\} \end{aligned}$$

For $k = r_i + 1$, let

$$\begin{aligned}
P_{ik}^1 &= \{A \to Z, A \to Y_{ik}^1, \mid A \in N \cup \{\overline{S_i}\}\} \cup \{S_{ik}^1 \to F_1, F_1 \to Q_i^1\} \\
&\quad \cup \{S_{ik}^1 \to Y_{ik}^1, F_1 \to Y_{ik}^1, Y_{ik}^1 \to Y_{ik}^1\} \cup \{M \to \lambda, \overline{M} \to \lambda\}; \\
P_{ik}^2 &= \{X_i \to \lambda\} \cup \{S_{ik}^2 \to H_1, H_1 \to Q_i^2\} \\
&\quad \cup \{S_{ik}^2 \to Y_{ik}^2, H_1 \to Y_{ik}^2, \overline{X_i} \to Y_{ik}^2, Y_{ik}^2 \to Y_{ik}^2\},
\end{aligned}$$

where for $\alpha_{ik} = x_1 Q_{l_1} x_2 Q_{l_2} \ldots x_j Q_{l_j} x_{j+1}, x_t \in (N \cup T)^*$ for each $t, 1 \leq t \leq j+1$, we have $\widehat{\alpha_{ik}} = x_1 Q_{l_1}^1 x_2 Q_{l_2}^1 \ldots x_j Q_{l_j}^1 x_{j+1}$ and $\widehat{\beta_{ik}} = Q_{l_1}^2 Q_{l_2}^2 \ldots Q_{l_j}^2$.

$$\begin{aligned}
P_{a1} &= \{S_{a1} \to L_0, L_0 \to L_1, L_2 \to L_3, L_3 \to L_4, L_4 \to L_1\} \\
&\quad \cup \{L_1 \to Q_{11}^1 Q_{11}^2 \ldots \overline{Q_{pl_p}^1 Q_{pl_p}^2} \ldots Q_{n(r_n+1)}^1 Q_{n(r_n+1)}^2 L_2 \mid 1 \leq p \leq n, \\
&\quad 1 \leq l_p \leq r_p + 1\}; \\
P_{a2} &= \{S_{a2} \to V_0, V_0 \to V_1, V_1 \to V_2, V_3 \to V_4, V_4 \to V_1\} \\
&\quad \cup \{V_2 \to Q_{11}^1 Q_{11}^2 \ldots Q_{n(r_n+1)}^1 Q_{n(r_n+1)}^2 V_3\}.
\end{aligned}$$

Consider the returning PC grammar system

$$\begin{aligned}
\Gamma' &= (N', K, T \cup Lab(\Gamma), G_s, G_1^1, G_1^2, G_{11}^1, G_{11}^2, \ldots, G_{1(r_1+1)}^1, G_{1(r_1+1)}^2, \ldots, \\
&\quad G_n^1, G_n^2, G_{n1}^1, G_{n1}^2, \ldots, G_{n(r_n+1)}^1, G_{n(r_n+1)}^2, G_{a1}, G_{a2}).
\end{aligned}$$

In the production sets of $\Gamma'$, we have denoted by $Q_{11}^1 Q_{11}^2 \ldots Q_{n(r_n+1)}^1 Q_{n(r_n+1)}^2$ the sequence of all query symbols between $Q_{11}^1$ and $Q_{n(r_n+1)}^2$, in the order provided by the components' order in the system, and by $Q_{11}^1 Q_{11}^2 \ldots \overline{Q_{pl_p}^1 Q_{pl_p}^2} \ldots Q_{n(r_n+1)}^1 Q_{n(r_n+1)}^2$ we have denoted the sequence of all query symbols as above but $Q_{pl_p}^1$ and $Q_{pl_p}^2$.

One can prove that $L(\Gamma') = ESZ(\Gamma)$, and thus point $(i)$ of the theorem follows. The other inclusion can be proved with a similar construction, but replacing all terminal symbols in strings $\widehat{\alpha_{ik}}$ by corresponding non-terminals (considering for each $a \in T$ a new symbol $U_a$ and including all $U_a$ in $N'$) and adding to $P_s$ productions for erasing these non-terminals ($U_a \to \lambda$, for each $a \in T$). □

**Theorem 7** *(i)* $\mathbf{ESZ}(NCPC_*CF) \subseteq \mathbf{L}(PC_*CF)$;

*(ii)* $\mathbf{SZ}(NCPC_*CF) \subseteq \mathbf{L}(PC_*CF)$.

**Proof:** Let

$$\Gamma = (N, K, T, G_1, G_2, \ldots, G_n)$$

be a non-returning centralized PC grammar system with $n$ context-free components, where $G_i, P_i, 1 \leq i \leq n$, and $Lab(\Gamma)$ are just as in the proof of the previous theorem.

Let $\overline{S_i}, S_i', \overline{X_i}, X_i', S_i^j, S_{ik}^l, Y_i^j, Y_{ik}^l, 1 \leq i \leq n, 1 \leq j \leq 4, 1 \leq l \leq 2, 1 \leq k \leq r_i+1$, and $\overline{M}, M, Z, Z', S', S'', S_{a1}, S_{a2}, S_{a3}, D_t, E_t, 0 \leq t \leq 2, F_1, H_1, A_0, B_0, U_0, L_p, V_p, 0 \leq p \leq 4$, be new symbols, let $N'$ be the union of $N$ with the set of all these symbols. We consider

$P_s = \{S_s \to S_s, S_s \to Q_1^1 Q_1^2\} \cup \{X_i \to \lambda \mid 1 \leq i \leq n\} \cup \{M \to \lambda\}$.

For each $i, 1 \leq i \leq n$,

$$\begin{aligned}
P_i^1 &= \{S_i^1 \to S', S' \to \overline{S_i M}\} \cup \{S_i^1 \to D_0, D_0 \to D_1, D_1 \to D_2\} \\
&\quad \cup \{D_2 \to Q_{ik}^1 \mid 1 \leq k \leq r_i+1\} \cup \{A \to A \mid A \in N \cup \{M, Z\}\} \\
&\quad \cup \{S_i^1 \to Y_i^1, Y_i^1 \to Y_i^1\}, \\
P_{i2} &= \{S_i^2 \to S'', S'' \to \overline{X_i}\} \cup \{S_i^2 \to E_0, E_o \to E_1, E_1 \to E_2\} \\
&\quad \cup \{E_2 \to Q_{ik}^2 \mid 1 \leq k \leq r_i+1\} \cup \{X_i \to X_i\} \cup \{S_i^2 \to Y_i^2, Y_i^2 \to Y_i^2\}, \\
P_i^3 &= \{S_i^3 \to A_0, A_0 \to Q_i^1\} \cup \{A \to A \mid A \in N \cup \{Z, \overline{S_i}\}\} \\
&\quad \cup \{S_i^3 \to Y_i^3, Y_i^3 \to Y_i^3\}, \\
P_i^4 &= \{S_i^4 \to B_0, B_0 \to Q_i^2\} \cup \{X_i \to X_i, \overline{X_i} \to \overline{X_i}\} \cup \{S_i^4 \to Y_i^4, Y_i^4 \to Y_i^4\}.
\end{aligned}$$

For each $k, 1 \leq k \leq r_i$,

$$\begin{aligned}
P_{ik}^1 &= \{A_{ik} \to A_{ik}', A_{ik}' \to \widehat{\alpha_{ik}} \mid (i,k) : A_{ik} \to \alpha_{ik} \in P_i\} \\
&\quad \cup \{A \to Z', Z' \to Z \mid A \in N \cup \{M\}\} \cup \{S_{ik}^1 \to F_1, F_1 \to Q_i^3\} \\
&\quad \cup \{S_{ik}^1 \to Y_{ik}^1, F_1 \to Y_{ik}^1, \overline{S_i} \to Y_{ik}^1, Y_{ik}^1 \to Y_{ik}^1\} \cup \{\overline{M} \to M\} \\
&\quad \cup \{\overline{S_i} \to \widehat{\alpha_{ik}} \mid (i,k) : S_i \to \alpha_{ik} \in P_i\}, \\
P_{ik}^2 &= \{X_i \to X_i', X_i' \to (i,k)\widehat{\beta_{ik}} X_i \mid (i,k) : A_{ik} \to \alpha_{ik} \in P_i\} \\
&\quad \cup \{S_{ik}^2 \to H_1, H_1 \to Q_i^4\} \cup \{\overline{X_i} \to X_i'\} \\
&\quad \cup \{S_{ik}^2 \to Y_{ik}^2, H_1 \to Y_{ik}^2, \overline{X_i} \to Y_{ik}^2, Y_{ik}^2 \to Y_{ik}^2\}.
\end{aligned}$$

For $k = r_i + 1$,

$$\begin{aligned}
P_{ik}^1 &= \{A \to Z, A \to Y_{ik}^1, \mid A \in N \cup \{\overline{S_i}\}\} \cup \{S_{ik}^1 \to F_1, F_1 \to Q_i^1\} \\
&\quad \cup \{S_{ik}^1 \to Y_{ik}^1, F_1 \to Y_{ik}^1, Y_{ik}^1 \to Y_{ik}^1\} \cup \{M \to \lambda, \overline{M} \to \lambda\}, \\
P_{ik}^2 &= \{X_i \to \lambda\} \cup \{S_{ik}^2 \to H_1, H_1 \to Q_i^2\} \\
&\quad \cup \{S_{ik}^2 \to Y_{ik}^2, H_1 \to Y_{ik}^2, \overline{X_i} \to Y_{ik}^2, Y_{ik}^2 \to Y_{ik}^2\},
\end{aligned}$$

where $\widehat{\alpha_{ik}}$ and $\widehat{\beta_{ik}}$ are just as in the proof of the previous theorem. We still consider

$$P_{a1} = \{S_{a1} \to L_0, L_0 \to L_1, L_1 \to L_2, L_2 \to L_3\}$$

$$
\begin{aligned}
& \cup\{L_3 \to Q^1_{11}Q^2_{11}\ldots\overline{Q^1_{pl_p}Q^2_{pl_p}}\ldots Q^1_{n(r_n+1)}Q^2_{n(r_n+1)}L_0 \mid 1 \leq p \leq n, \\
& 1 \leq l_p \leq r_p + 1\}, \\
P_{a2} \quad = \quad & \{S_{a2} \to V_0, V_1 \to V_2, V_2 \to V_3, V_3 \to V_0\} \\
& \cup\{V_0 \to Q^1_{11}Q^2_{11}\ldots Q^1_{n(r_n+1)}Q^2_{n(r_n+1)}V_1\}, \\
P_{a3} \quad = \quad & \{S_{a3} \to Q^1_{11}Q^2_{11}\ldots Q^1_{n(r_n+1)}Q^2_{n(r_n+1)}U_0, U_0 \to U_0\}.
\end{aligned}
$$

(In the productions of components $P_{aj}, 1 \leq j \leq 3$, we have considered the same conventions as in the preceding theorem.) Then for the returning PC grammar system

$$
\begin{aligned}
\Gamma' \quad = \quad & (N', K, T \cup Lab(\Gamma), G_s, G^1_1, G^2_1, G^3_1, G^4_1, G^1_{11}, G^2_{11}, \ldots, G^1_{1(r_1+1)}, G^2_{1(r_1+1)}, \\
& \ldots, G^1_n, G^2_n, G^3_n, G^4_n, G^1_{n1}, G^2_{n1}, \ldots, G^1_{n(r_n+1)}, G^2_{n(r_n+1)}, G_{a1}, G_{a2}, G_{a3})
\end{aligned}
$$

one can prove that $L(\Gamma') = ESZ(\Gamma)$ and hence the first point of the theorem follows. The second one follows with similar arguments as in the proof of the preceding theorem. □

# 5 Conclusions

The properties outlined in this paper point out that in the regular, as well as in the linear case the PC grammar systems (of any type) are powerful enough to generate also the Szilard languages associated to them. We do not know whether this is also the case for context-free PC grammar systems or not. But we have proved once again the power of returning non-centralized PC grammar systems, which can generate the Szilard and the extended Szilard languages associated to centralized returning and non-returning PC grammar systems. The situations we have dealt with concern only synchronized derivation. It remain to be investigated Szilard and extended Szilard languages when considering the unsynchronized or the terminal mode of derivation.

As a final remark, note that by counting in a word of the Szilard language associated to a PC grammar system the labels of a component of the system, one gets a measure of the amount of work that component has accomplished in deriving the corresponding word. Moreover, from the Szilard language associated to a PC grammar system one can extract informations about the number of communications accomplished during the derivation of a terminal word. And this can be viewed as a measure of how cooperating the system was (how communicating the team of the classroom was).

## References


[1] A. Atanasiu, V. Mitrana, *The modular grammars*, Internat. J. Comp. Math., 30 (1989), 101-122

[2] E. Csuhaj-Varju, J. Dassow, *On cooperating distributed grammar systems*, J. Inf. Processing and Cybern. EIK, 26 (1990), 49-63.

[3] E. Csuhaj-Varju, J. Dassow, J. Kelemen, Gh. Păun, *Grammar Systems: A Grammatical Approach to Distribution and Cooperation*, Gordon and Breach Science Publishers Ltd, London 1994

[4] J. Dassow, Gh. Păun, *Regulated Rewriting in Formal Language Theory*, Springer-Verlag, Berlin, Heidelberg, 1989

[5] J. Dassow, V. Mitrana, Gh. Păun, *Szilard Languages Associated to Cooperating Distributed Grammar Systems*, Stud. Cercet. Mat., 45 (1993), 403-413

[6] R. Meersman, G. Rozenberg, *Cooperating grammar systems*, in Proc. MFCS'78 Symp. LNCS 64, Springer-Verlag, Berlin, 1978, 364-374

[7] V. Mihalache, *On the Generative Capacity of Parallel Communicating Grammar Systems with Regular Components*, Computers and Artificial Intelligence, to appear

[8] Gh. Păun, L. Sântean, *Parallel communicating grammar systems: the regular case*, Ann. Univ. Buc., Ser. Matem.-Inform., 38 (1989), 55-63.

[9] M. Penttonen, *On derivation languages corresponding to context-free grammars*, Acta Informatica, 3 (1974), 285-291

[10] A. Salomaa, *Formal Languages*, Academic Press, New York, London, 1973

**Turku Centre for Computer Science**
**Lemminkäisenkatu 14**
**FIN-20520 Turku**
**Finland**

http://www.tucs.abo.fi

**University of Turku**
- **Department of Mathematical Sciences**

**Åbo Akademi University**
- **Department of Computer Science**
- **Institute for Advanced Management Systems Research**

**Turku School of Economics and Business Administration**
- **Institute of Information Systems Science**