On Szilard languages of labelled insertion grammars *

Paul Prithwineel

On Szilard languages of labelled insertion grammars

*Prithwineel Paul
Electronics and Communication Sciences Unit
Indian Statistical Institute
Kolkata - 700108, India
prithwineelpaul@gmail.com

October 26, 2018

Abstract

In this work we initiate the study of Szilard languages of labelled insertion grammars. It is well-known that there exist context-free languages which cannot be generated by any insertion grammar. We show that there exist some regular languages which cannot be Szilard language of any labelled insertion grammar. But any regular language can be given as a homomorphic image of Szilard language obtained by a labelled insertion grammar of weight 1. Also, any context-free language can be obtained as a homomorphic image of Szilard language of a labelled insertion grammar of weight 2. We show that even though insertion grammars of weight 1 can generate only context-free languages, there exist some context-sensitive language which can be obtained as Szilard language of a labelled insertion grammar of weight 1. At the end we show that any recursively enumerable language can be characterized by the homomorphic image of Szilard language obtained by a labelled insertion grammar of weight 5.

Insertion grammar, Szilard languages, Labelled insertion grammar, Chomsky hierarchy

1 Introduction

Insertion and deletion operations are well-known in formal language theory. In insertion operation, a string is inserted in the specified contexts when the insertion rule is applied, i.e., the string uv is transformed into uxv after application of the insertion rule $(u, \lambda/x, v)$. Similarly, the deletion operation removes strings from the specified contexts and the string uxv is transformed into uv after application of the deletion rule $(u, x/\lambda, v)$ where u and v are contexts. Ins-Del (i.e., insertion-deletion) systems work as a language generating device. These systems are powerful and with only finite set of rules and axioms can characterize recursively enumerable languages. Ins-Del systems and their variants have

*prithwineelpaul@gmail.com

been investigated in [30, 34, 29, 7, 32, 9, 18, 33, 23]. The study of insertion grammars (semicontextual grammars) was initiated in [28]. Computational power, closure properties etc. of the insertion systems have been discussed in [2, 32, 3, 27, 6, 25, 26]. In [5] it was proved that the linear language $\{a^n b a^n | n \geq 1\}$ cannot be generated by any insertion grammar. But any recursively enumerable language can be generated by insertion grammars of weight 3 when a homomorphism and weak coding is applied [2, 19]. Moreover, in the study of matrix insertion grammars initiated in [12], it has been shown that matrix insertion grammars can even characterize recursively enumerable languages. The computational power of the insertion-deletion systems and insertion grammars combined with the parallel distributed computing models such as P systems, also have been discussed in [20, 31].

The main contribution of the paper is the association of the well-known concept of Szilard languages with insertion grammars and compare the Szilard languages obtained by these grammars with the family of languages in Chomsky hierarchy. The idea of Szilard languages is well investigated in formal language theory and their closure properties, decidability aspects, complexity aspects for Chomsky grammars, matrix grammars, parallel communicating grammar systems, communicating distributive grammar systems have been investigated in [22, 16, 14, 17, 15]. Also, the idea of derivation languages (as Szilard and Control languages) has been introduced for DNA and membrane computing models in [13, 21]. In [13], derivation languages have been associated with splicing systems and in [21] the same were introduced for splicing P systems.

In this work, we show that there exist some regular languages which cannot be obtained as Szilard language by any insertion grammar. But some labelled insertion grammars of weight 1 can obtain context-sensitive languages as a Szilard language. We also show that labelled insertion grammars with rules of weight 5 can characterize recursively enumerable languages when a morphism is applied and any regular language can be represented as a homomorphic image of a Szilard language obtained by labelled insertion grammar of weight 1. In [24], it has been shown that there exist some context-free languages which cannot be represented as a homomorphic image of any context-free language. But in this paper, we show that any context-free language can be obtained as a homomorphic image of Szilard language of a labelled insertion grammar of weight 2.

The paper is organized as follows. In section 2 we recall the basic definitions required for this paper along with some well-known results of insertion grammars. In section 3, we define labelled insertion grammar and the main results have been discussed in section 4. The section 5 is conclusive in nature.

2 Preliminaries

For the basic definitions and notions of formal language theory we refer to [1].

Chomsky normal form [1]: For every context-free grammar G, a grammar $G^{'} = (N, T, S, P)$ can be effectively constructed where the rules in P are of the form $A \to BC$ and $A \to a$ such that $L(G) \setminus \{\lambda\} = L(G^{'}) \setminus \{\lambda\}$.

Type-0-grammar: A type-0-grammar is a construct $G = (N, T, S,\ P)$ where N is the non-terminal alphabet and T is the terminal alphabet such that $N \cap T = \emptyset$. The starting symbol $S \in N$ and the rules in P are ordered pairs (u, v) where $u \in (N \cup T)^* N (N \cup T)^*$ and $v \in (N \cup T)^*$.

Kuroda normal form: Every type-0 grammar $G = (N, T, S, P)$ is in Kuroda normal form if the rules of the grammar G has one of the following forms:

$A \to BC, AB \to CD, A \to a, A \to \lambda$ for $A, B, C, D \in N$ and $a \in T$.

Homomorphism: A homomorphism is a mapping h from Σ^* to Δ^* where Σ, Δ are alphabets, preserving concatenation, i.e., $h(v.w) = h(v).h(w), v, w \in \Sigma^*$.

Weak coding: A weak coding is a morphism which maps each letter onto a letter or empty string.

Szilard languages [1]: Let $G = (N, T, S, P)$ be Chomsky grammar and F be an alphabet such that the cardinality of the sets F and P is same. Let f be a mapping from P to F such that for each $p \in P$ a unique label $f(p)$ is associated with p and is called the label of the rule p. A derivation in G is called successful if a string over T is generated staring from S. With each successful derivation of G, a string over F can be associated if labels of the any successful derivation are concatenated sequentially. The language generated in this manner is called Szilard language of the grammar G and is denoted by $SZ(G)$.

Example 1. *Let $G = (\{S\}, \{a, b\}, S, \{S \to aSb, S \to ab\})$ be a context-free grammar. The rules are labelled in the following manner: $f_1 : S \to aSb, f_2 : S \to ab$. Hence, the Szilard language obtained by the grammar is $SZ(G) = \{f_1^n f_2 \mid n \geq 0\}$.*

The family of finite, linear, regular, context-free, context-sensitive and recursively enumerable languages is denoted by $FIN, LIN, REG, CF, CS, RE$ respectively.

Insertion grammar [3]: An insertion grammar is a construct $G = (V, A, P)$ where V is the set of alphabets, A is the set of initial strings and P is the set of insertion rules.

Let G be an insertion grammar, then the relation $\Rightarrow$ is defined in the following manner:

$w \Rightarrow z$ if and only if $w = w_1 uv w_2, z = w_1 uxv w_2$ for $(u, \lambda/x, v) \in P, w_1, w_2 \in V^*$.

The language generated by the insertion grammar G is:

$L(G) = \{z \in V^* | w \Rightarrow^* z, w \in S\}$.

Moreover, an insertion grammar G is called of weight n [2] if and only if

$n = max\{|u| \mid (u, \lambda/x, v) \in P$ or $(v, \lambda/x, u) \in P, x \in V^*\}$.

The family of languages generated by the insertion grammars of weight n is denoted as INS_n and union of all these families is denoted as INS_∞.

The followings are well known results in insertion grammars [28, 27, 6, 26]:

(1)$FIN \subset INS_1 \subset INS_2 \subset INS_3 \ldots \subset INS_\infty \subset CS$.

(2) REG is incomparable with all families $INS_n, n \geq 1$ and $REG \subset INS_\infty$.

(3) $INS_1 \subset CF$ but CF is incomparable with all $INS_n, n \geq 2$ and INS_∞. Also INS_2 contains non-semilinear languages.

(4) LIN is incomparable with all $INS_n, n \geq 0$ and INS_∞.

(5) Each regular language is the homomorphic image of a language in INS_1.

The following characterization of recursively enumerable language was proved by Kari and Sosik in [2] and Onodera in [19] independently where S_3 denotes the family of languages generated by insertion grammars of weight at most 3:

Theorem 2.1. *For each recursively enumerable language L there exists a morphism h, a weak coding g and a language $L_1 \in S_3$ such that $L = g(h^{-1}(L_1))$.*

3 Labelled insertion grammar

A labelled insertion grammar is a construct $\Gamma = (V_1, A_1, P_1, Lab)$ where $V_1 \cap Lab = \emptyset$ and the rules of P_1 are labelled in one-to-one manner with the elements from the set Lab. A derivation of insertion grammar is called a *terminal derivation* if it is as follows:

$x_0 \Rightarrow^{a_1} x_1 \Rightarrow^{a_2} x_2 \Rightarrow^{a_3} \ldots \Rightarrow^{a_n} x_n$ where $x_0 \in A_1$ and no rule of Γ is applicable to x_n.

If the labels of the applied insertion rules in the above terminal derivation are concatenated in the order of application, a string over Lab is obtained and the set of all such strings forms a language which is different from the language generated by the insertion grammar. It is called Szilard language of the labelled insertion grammar Γ. From the above derivation, the string $a_1a_2 \ldots a_n \in SZINS_m(\Gamma)$ where $m = max\{|u| \mid (u, \lambda/\alpha, v) \in P_1$ or $(v, \lambda/\alpha, u) \in P_1\}$.

The notation $SZINS_m(\Gamma)$ denotes the Szilard language of the labelled insertion grammar Γ of weight m. The family of Szilard languages $SZINS_m(\Gamma)$ of the labelled insertion grammars with insertion rules of size m, is denoted as $SZINS_m$. When m is not specified, it is replaced by $*$.

In the next section, we discuss the main results of the Szilard languages of the insertion grammars with respect to the weight. At first, we prove that there exist some regular languages which cannot be obtained as a Szilard language by any labelled insertion grammar. But any regular language can be given as homomorphic image of a Szilard language of a labelled insertion grammar of weight 1. Also, any context-free language can be given as a homomorphic image of Szilard language of a labelled insertion grammar of weight 2. Furthermore, any recursively enumerable language can be characterized by Szilard language of the labelled insertion grammar of weight 4 when a homomorphism is applied.

4 The Main Results

It is very well-known that languages such as $\{aa\}$ are not a Szilard language of any Chomsky grammar. But we show that it can be Szilard language of a labelled insertion grammar.

Theorem 4.1. *$\{aa\}$ is a Szilard language of a labelled insertion grammar.*

Proof. We construct a labelled insertion grammar Γ such that $SZINS_m(\Gamma) = \{aa\}$. Let $\Gamma = (V_1, A_1, R_1, Lab)$ be a labelled insertion grammar where, $V_1 = \{S, X_a, Y\}$, $T_1 = \{S, Y\}$, $A = \{SYSY\}$, $R = \{a : (S, \lambda/X_a, Y)\}$, $Lab = \{a\}$.

Initially only the string $SYSY$ is present and after application of the a-rule, it is transformed into either $SYSX_aY$ or SX_aYSY. But if the a-rule is applied once again, then the string SX_aYSX_aY is obtained. No further computation is possible. Hence, $SZINS_1(\Gamma) = \{aa\}$. □

In the next theorem, we show that there exist some regular languages which cannot be obtained as Szilard language by any labelled insertion grammar.

Theorem 4.2. $\{a^n \mid n \geq 1\} \notin SZINS_*$.

Proof. Suppose $\{a^n \mid n \geq 1\}$ is Szilard language of a labelled insertion grammar $\Gamma = (V_1, A_1, R_1, Lab)$ where $R_1 = \{a : (u, \lambda/k, v)\}$ and $Lab = \{a\}$. Hence there exists a terminal derivation such that

$x_0^0 \Rightarrow^a x_1^0$. ...(1)

where $x_0^0 \in A_1$ and the insertion rule $a : (u, \lambda/k, v)$ is not applicable to x_1^0.

Similarly, the terminal derivation of the word a^2 is as follows:

$x_0^1 \Rightarrow^a x_1^1 \Rightarrow^a x_2^1$. ...(2)

where $x_0^1 \in A_1$ and the insertion rule $a : (u, \lambda/k, v)$ is applicable to x_1^1 but not to x_2^1.

But, the x_0^0 in (1) and the x_0^1 in (2) cannot be same. Because, if (1) is true, then (2) cannot be true for $x_0^0 = x_0^1$.

Again, following is the terminal derivation for a^3:

$x_0^2 \Rightarrow^a x_1^2 \Rightarrow^a x_2^2 \Rightarrow^a x_3^2$. ...(3)

So, from (3) we can infer that the $x_0^2 \in A_1$ must be different from the x_0^0 in (1) and x_0^1 in (2).

Hence, to obtain $\{a^n \mid n \geq 1\}$ as Szilard language, all $x_0^i (i \in \mathbb{N} \cup \{0\}) \in A_1$ must be distinct.

Since any derivation starts from a $x_0^i \in A_1$, the language $\{a^n \mid n \geq 1\}$ cannot be obtained as Szilard language by the labelled insertion grammar Γ where A_1 contains only finite number of elements. □

Although $\{a^n \mid n \geq 1\}$ cannot be the Szilard language of any labelled insertion grammar, any regular language can be represented as a homomorphic image of the Szilard language of labelled insertion grammar of weight 1.

Theorem 4.3. *Any non-empty regular language can be obtained as a homomorphic image of Szilard language of a labelled insertion grammar of weight* 1.

Proof. Let L be a λ-free regular language and let $G = (N, T, S, P)$ be a right linear grammar such that $L = L(G)$. Suppose $N = \{D_1, D_2, \ldots, D_n\}$ where $D_1 = S$ is the start symbol. Now we construct a labelled insertion grammar Γ such that $L = L(G) = h(SZINS_1(\Gamma))$ where h is a homomorphism. The rules in P are of the form $D_i \to aD_i$, $D_i \to aD_j (i \neq j)$, and $D_i \to a$, $D_i, D_j \in N$, and $a \in T$.
Let $\Gamma = (V_1, A_1, R_1, Lab)$ be a labelled insertion grammar where

- $V_1 = \{X, Y, D_1, D_2, \ldots, D_n\} \cup T$;
- $A_1 = \{XD_1Y\}$;

- The rules in R_1 are of the form

$$a^i_k : (D_i, \lambda/aD_k, Y) \text{ for } D_i \to aD_k, D_k \in N, a \in T$$

$$a^i : (D_i, \lambda/a, Y) \text{ for } D_i \to a, a \in T;$$

- $Lab = \{a^i_k \mid D_i \to aD_k, D_k \in N, a \in T\} \cup \{a^i \mid D_i \to a, a \in T\}$.

The non-erasing homomorphism $h : (Lab)^* \to T^*$ is defined as $h(a^i_k) = a$ and $h(a^i) = a$ where $a^i_k, a^i \in Lab$.

Every non-terminal rule $(D_i \to aD_k)$ in G is simulated by an insertion rule with the label a^i_k and every terminal rule $(D_i \to a)$ of G is associated with label a^i. If the labelled insertion rules simulating the rules of a terminal derivation in G are applied in the same order, then concatenation of the labels of the insertion rules will obtain a string $w_1 \in (Lab)^*$ such that $h(w_1) = w$. Hence, $w \in h(SZINS_1(\Gamma))$.

For the proof of the inclusion $h(SZINS_1(\Gamma)) \subseteq L(G)$, first assume that $w \in h(SZINS_1(\Gamma))$. Hence, there exists a terminal derivation in Γ such that $h(w_1) = w$ where $w_1 \in SZINS_1(\Gamma)$. If the rules in G are applied in the same order as in the terminal derivation obtaining w_1, the string $w \in L(G)$ is generated. Hence, $h(SZINS_1(\Gamma)) \subseteq L(G)$. This will imply, $L(G) = h(SZINS_1(\Gamma))$.

□

It has been shown in [3] that $INS_1 \subset CF$. In fact, the linear language $\{a^n b a^n \mid n \geq 1\}$ cannot be generated by any insertion grammar [5]. We show that there exists a context-sensitive language which is Szilard language of a labelled insertion grammar of weight one.

Theorem 4.4. *$CS \cap SZINS_1 \neq \emptyset$.*

Proof. We construct a labelled insertion grammar Γ which has a context-sensitive language as a Szilard language.

Let $\Gamma = (V_1, A_1, R_1, Lab)$ be a labelled insertion grammar where

- $V_1 = \{X, A, A^{'}, A^{''}, Y\}$;
- $A_1 = \{XAY\}$;
- $R_1 = \{a : (A, \lambda/A, Y), b : (A, \lambda/A^{'}, A), c : (A, \lambda/A^{''}, A^{'})\}$;
- $Lab = \{a, b, c\}$.

Any computation in Γ starts from the string XAY. When the a-rule is applied, one "A" is added between A and Y. Application of the b-rule inserts $A^{'}$ between the two A's of the string AA and hence $AA^{'}A$ is obtained. Similarly, when the c-rule is applied, the string $AA^{'}$ is transformed into $AA^{''}A^{'}$. Hence, if the labeled rules are applied in a particular order, we have $SZINS_1(\Gamma) \cap a^*b^*c^* = \{a^n b^n c^n \mid n \geq 1\}$. Since the intersection of $SZINS_1(\Gamma)$ and the regular language $a^*b^*c^*$ is a context-sensitive language. The language $SZINS_1(\Gamma)$ must be non context-free. □

Păun showed in [24] that there exist context-free languages which cannot be given as a homomorphic image of Szilard language of any context-free language. Also,

Theorem 4.5. *[24] The families of context-free languages and homomorphic image of the Szilard languages of the context-free languages are incomparable.*

But any context-free language can be obtained as a homomorphic image of Szilard language of insertion grammar of weight 2. We show it in the following theorem.

Theorem 4.6. *Any context-free language can be given as a homomorphic image of Szilard language of a labelled insertion grammar of weight* 2*.*

Proof. Let L be a non-empty context-free language and let $G = (N, T, S, P)$ be a Chomsky normal form grammar such that $L = L(G)$. The rules in P are of the form, $A \to BC$ and $A \to a$, where $A, B, C \in N, a \in T$ and each rule in G is assigned with a unique label r_i. Also each element of L can be obtained by initial application of the non terminal rules and then by application of the terminal rules in the leftmost manner.

We construct a labelled insertion grammar $\Gamma = (V_1, A_1, R_1, Lab)$ such that $L = h(SZINS_m(\Gamma))$ where h is a morphism from Lab^* to T^*. The grammar $\Gamma = (V_1, A_1, R_1, Lab)$ is the labelled insertion grammar where

- $V_1 = \{X, Y, E\} \cup N \cup T \cup \Delta_1 \cup \Delta_2 \cup \{\#, \$\}$ where $\Delta_1 = \{[r_i] \mid r_i : A \to BC\}, \Delta_2 = \{[r_i] \mid r_i : A \to a\}$;
- $A_1 = \{XSEY\}$;
- R_1 contains the following rules:

  For $r_i : A \to BC$:

  $r_i^{'} : (A, \lambda/[r_i]BC, \alpha_1\alpha_2)$ where

  $\alpha_1 \in N \cup \{E\}, \alpha_2 \in N \cup \{Y, E\} \cup \Delta_1, \alpha_1\alpha_2 \notin N\{Y\} \cup \{EE\} \cup \{E\}N \cup \{E\}\Delta_1,$

  For $r_i : A \to a$:

  $r_a^i : (\$A, \lambda/[r_a], \alpha_1), \alpha_1 \in N \cup \{E\}$

  and

  $r_3 : (X, \lambda/\#, \alpha_2), \alpha_2 \in N,$

  $r_4 : (\alpha_1\alpha_2, \lambda/\$, \alpha_3),\ \alpha_1 \in \{\#, \$\}, \alpha_2 \in N \cup \Delta_1 \cup \Delta_2,$

  $\alpha_3 \in \Delta_1 \cup \Delta_2 \cup N, \alpha_1\alpha_2 \notin \{\#\}\Delta_1 \cup \{\#\}\Delta_2.$

- $Lab = \{r_i^{'} \mid [r_i] \in \Delta_1\} \cup \{r_a^i \mid [r_i] \in \Delta_2\} \cup \{r_3, r_4\}.$

Finally, we define the morphism $h : Lab^* \to T^*$ by $h(r_i^{'}) = h(r_3) = h(r_4) = \lambda, h(r_a^i) = a$ where $r_i^{'}, r_a^i, r_3, r_4 \in Lab$ and $a \in T$.

We first prove that $L(G) = L \subseteq h(SZINS_2(\Gamma))$. The rule $r_i : A \to BC$ can be simulated by application of the $r_i^{'}$-rule in the following manner: $Xw_1Aw_2Y \to^{r_i^{'}} Xw_1A[r_i]BCw_2Y$ where the $\alpha_1\alpha_2$ is a subword of w_2Y and the $r_i^{'}$-rule is applicable only when $\alpha_1\alpha_2$ satisfy the predefined conditions. The r_3-rule transforms the word $X\beta wEY$ into $X\#\beta wEY$ where $\beta \in N, w \in V_1^*$. The r_4-rule inserts \$ into the word $X\#wEY, w \in V_1^+$ to identify the leftmost non-terminal where r_a^i-rule can be applied. The $r_i : A \to a$ rule can be simulated in the following manner: $X\#w_1\$A\beta_1w_2Y \to^{r_a^i} X\#w_1\$A[r_a]\beta_1w_2Y$ where $\beta_1 \in N \cup \{E\}, w_1, w_2 \in V_1^*$. Moreover, after application of the r_a^i-rule, no rule will be applicable to the subword $\$A$.

Since, any terminal string $w \in L$ can be obtained by application of non-terminal rules and then by leftmost application of terminal rules, if the corresponding labelled insertion rules are applied in the same order, a terminal derivation can be obtained in Γ. In fact, a string over V_1 can be obtained where no rules further can be applied. If the labels of the applied insertion rules are concatenated in order of application, a string over Lab, say, w_1 is obtained. Application of the morphism h to w_1, replaces each occurrence of $r_i^{'}, r_3$ and r_4 by the empty string and r_a^i is replaced by a. Hence, if $w \in L(G)$, then $w = h(w_1) \in h(SZINS_2(\Gamma))$ where $w_1 \in SZINS_2(\Gamma)$.

Next we prove the inclusion $h(SZINS_2(\Gamma)) \subseteq L(G) = L$. Let $w = h(w_1)$ where $w_1 = a_1a_2 \dots a_n \in SZINS_2(\Gamma)$. If the rules in G are applied in the same order as the labelled rules in Γ, a terminal string is obtained. Moreover, after application of each r_a^i-rule where $r_i : A \to a$, the subword $\$A$ becomes inactive. Hence, no extra derivation is possible in Γ. Again, $h(r_i^{'}) = h(r_3) = h(r_4) = \lambda$ and $h(r_a^i) = a$, and hence, $h(w_1) = w \in L(G)$. So, we can conclude $h(SZINS_2(\Gamma)) \subseteq L(G)$.

□

Kari and Sosik [2] and Onodera [19] proved that insertion grammars of weight 3 can characterize recursively enumerable languages when an inverse morphism and a weak coding is applied. Next, we show that any recursively enumerable language can be obtained as a homomorphic image of Szilard language of a labelled insertion grammar of weight 5. Moreover, we construct the insertion grammar in such a way that it simulates the derivations of G where the terminal symbols in any sentential form are generated from right to left order, i.e., in leftmost manner as in [2, 3].

Theorem 4.7. *Each recursively enumerable language can be obtained as a homomorphic image of the Szilard language of a labelled insertion grammar of weight* 5.

Proof. Let $L \in RE$ and $G = (N, T, S, P)$ be a grammar in Kuroda normal form such that $L(G) = L$. The rules of the grammar G are of the form $A \to BC, AB \to CD, A \to a, A \to \lambda$. Moreover, as in the proof of Theorem 1 in [3], we can assume that each element $x \in L$ can be generated initially by application of the nonterminal rules and then by application of the terminal rules in leftmost manner. In this proof, we construct a labelled insertion grammar $\Gamma = (V_1, A_1, R_1, Lab)$ such that $L = h(SZINS_5(\Gamma))$ where $V_1 \cap Lab = \emptyset$.

Initially, the rules in G are labelled in one-to-one manner, i.e., each rule has a unique label r_i. The set Δ contains the labels of the rules in P. It is defined in the following manner

$\Delta = \Delta_1 \cup \Delta_2 \cup \Delta_3 \cup \Delta_4$, where

$\Delta_1 = \{[r_i] \mid r_i : A \to BC \in P\}$;

$\Delta_2 = \{[r_i] \mid r_i : AB \to CD \in P\}$;

$\Delta_3 = \{[r_i] \mid r_i : A \to a \in P\}$;

$\Delta_4 = \{[r_i] \mid r_i : A \to \lambda \in P\}$;

Let $\Gamma = (V_1, A_1, R_1, Lab)$ be a labelled insertion grammar, where

- $V_1 = \{X, Y\} \cup N \cup \{k_a^i | r_i : A \to a\} \cup \{k_\lambda^i | r_i : A \to \lambda\} \cup \{[r_i] \mid r_i \in \Delta\} \cup \{[r_m]\}$;
- $A_1 = \{XSY\}$;
- R_1 contains the following rules

  (R_{11}) For $r_i : A \to BC$:

  $r_i^1 : (A, \lambda/[r_i]BC, Y)$,

  $r_i^2 : (A, \lambda/[r_i]BC, \alpha_1 Y), \alpha_1 \in N$,

  $r_i^3 : (A, \lambda/[r_i]BC, \alpha_1\alpha_2 Y), \alpha_1, \alpha_2 \in N$,

  $r_i^4 : (A, \lambda/[r_i]BC, \alpha_1\alpha_2\alpha_3 Y), \alpha_1, \alpha_2, \alpha_3 \in N$,

  $r_i^5 : (A, \lambda/[r_i]BC, \alpha_1\alpha_2\alpha_3\alpha_4)$,

  where $\alpha_1 \in N, \alpha_2 \in N \cup \Delta_1, \alpha_3 \in N \cup \Delta_1 \cup \Delta_2, \alpha_4 \in N \cup \Delta_1 \cup \Delta_2$,

  $\alpha_2\alpha_3 \notin (\Delta_1)(\Delta_1 \cup \Delta_2), \alpha_3\alpha_4 \notin (\Delta_1 \cup \Delta_2)(\Delta_1 \cup \Delta_2)$,

  $r_i^6 : (A, \lambda/[r_i]BC, \alpha_1\alpha_2\alpha_3\alpha_4\alpha_5)$, where $\alpha_1 \in N, \alpha_2 \in \Delta_2, \alpha_3 \in N, \alpha_4 \in \Delta_1$ and

  $\alpha_2 = \alpha_5$.

  (R_{12}) For $r_i : AB \to CD$:

  $r_i^7 : (AB, \lambda/[r_i]CD, \alpha_1\alpha_2), \alpha_1 \in N, \alpha_2 \in N$,

  $r_i^8 : (AB, \lambda/[r_i]CD, Y)$,

  $r_i^9 : (AB, \lambda/[r_i]CD, \alpha_1 Y), \alpha_1 \in N$.

  The application of the rules r_i^{10}, r_i^{11} and r_i^{12} in order also can simulate the rule $r_i : AB \to CD$:

  $r_i^{10} : (A, \lambda/[r_i], \alpha_1\alpha_2), \alpha_1 \in N, \alpha_2 \in \Delta_1$,

  $r_i^{11} : ([r_i]\alpha_1\beta_1, \lambda/[r_i], \alpha_2\alpha_3), \alpha_1 \in N, \alpha_2 \in N, \alpha_3 \in \Delta_1 \cup N, [r_i] \in \Delta_2, \beta_1 \in \Delta_1$,

  $r_i^{12} : (\beta_1[r_i]B, \lambda/[r_i]CD, \alpha_1\alpha_2), \alpha_1 \in N, \alpha_2 \in N \cup \{Y\} \cup \Delta_1, [r_i] \in \Delta_2, \beta_1 \in \Delta_1$

  and

  $r_i^{13} : (AB, \lambda/[r_i]CD, \alpha_1\alpha_2\alpha_3\alpha_4\alpha_5)$,

  where $\alpha_1 \in N, \alpha_2 \in \Delta_2, \alpha_3 \in N, \alpha_4 \in \Delta_1$ and $\alpha_2 = \alpha_5$.

  (R_{13}) For $r_i : A \to a$:

$a_i^1 : (XA, \lambda/k_a^i, Y)$,

$a_i^2 : ([r_m]A, \lambda/k_a^i, Y)$,

$a_i^3 : ([r_m]A, \lambda/k_a^i, \alpha_1 Y), \alpha_1 \in N$,

$a_i^4 : ([r_m]A, \lambda/k_a^i, \alpha_1\alpha_2 Y), \alpha_1, \alpha_2 \in N$,

$a_i^5 : ([r_m]A, \lambda/k_a^i, \alpha_1\alpha_2\alpha_3 Y), \alpha_1, \alpha_2, \alpha_3 \in N$,

$a_i^6 : ([r_m]A, \lambda/k_a^i, \alpha_1\alpha_2\alpha_3\alpha_4)$,

where $\alpha_1 \in N, \alpha_2 \in N \cup \Delta_1, \alpha_3 \in N \cup \Delta_1 \cup \Delta_2, \alpha_4 \in N \cup \Delta_1 \cup \Delta_2$,

$\alpha_2\alpha_3 \notin (\Delta_1)(\Delta_1 \cup \Delta_2), \alpha_3\alpha_4 \notin (\Delta_1 \cup \Delta_2)(\Delta_1 \cup \Delta_2)$,

$a_i^7 : ([r_m]A, \lambda/k_a^i, \alpha_1\alpha_2\alpha_3\alpha_4\alpha_5)$,

where $\alpha_1 \in N, \alpha_2 \in \Delta_2, \alpha_3 \in N, \alpha_4 \in \Delta_1$ and $\alpha_2 = \alpha_5$,

$r_{m+1}^i : ([r_m]Ak_a^i, \lambda/[r_m], \alpha_1\alpha_2), \alpha_1 \in N, \alpha_2 \in \{Y\} \cup N \cup \Delta_1 \cup \Delta_2$.

(R_{14}) For $r_i : A \to \lambda$:

$r_i^{14} : (XA, \lambda/k_\lambda^i, Y)$,

$r_i^{15} : ([r_m]A, \lambda/k_\lambda^i, Y)$,

$r_i^{16} : ([r_m]A, \lambda/k_\lambda^i, \alpha_1 Y), \alpha_1 \in N$,

$r_i^{17} : ([r_m]A, \lambda/k_\lambda^i, \alpha_1\alpha_2 Y), \alpha_1, \alpha_2 \in N$,

$r_i^{18} : ([r_m]A, \lambda/k_\lambda^i, \alpha_1\alpha_2\alpha_3 Y), \alpha_1, \alpha_2, \alpha_3 \in N$,

$r_i^{19} : ([r_m]A, \lambda/k_\lambda^i, \alpha_1\alpha_2\alpha_3\alpha_4)$,

where $\alpha_1 \in N, \alpha_2 \in N \cup \Delta_1, \alpha_3 \in N \cup \Delta_1 \cup \Delta_2, \alpha_4 \in N \cup \Delta_1 \cup \Delta_2$,

$\alpha_2\alpha_3 \notin (\Delta_1)(\Delta_1 \cup \Delta_2), \alpha_3\alpha_4 \notin (\Delta_1 \cup \Delta_2)(\Delta_1 \cup \Delta_2)$,

$r_i^{20} : ([r_m]A, \lambda/k_\lambda^i, \alpha_1\alpha_2\alpha_3\alpha_4\alpha_5)$,

where $\alpha_1 \in N, \alpha_2 \in \Delta_2, \alpha_3 \in N, \alpha_4 \in \Delta_1$ and $\alpha_2 = \alpha_5$.

$r_{m+2}^i : ([r_m]Ak_\lambda^i, \lambda/[r_m], \alpha_1\alpha_2), \alpha_1 \in N, \alpha_2 \in \{Y\} \cup N \cup \Delta_1 \cup \Delta_2$.

(R_{15}) $r_m : (X\alpha_1\beta_1, \lambda/[r_m], \alpha_2), \alpha_1, \alpha_2 \in N, \beta_1 \in \Delta_1$,

$r_{m+1} : ([r_m]\alpha_1\alpha_2\beta_1, \lambda/[r_m], \alpha_3),\ \alpha_1, \alpha_2, \alpha_3 \in N, \beta_1 \in \Delta_2$,

$r_{m+2} : ([r_m]\alpha_1\beta_1, \lambda/[r_m], \alpha_2\beta_2\beta_1), \alpha_1, \alpha_2 \in N, \beta_1 \in \Delta_2, \beta_2 \in \Delta_1$,

$r_{m+3} : ([r_m]\alpha_1\beta_1, \lambda/[r_m], \alpha_2\alpha_3), \alpha_1 \in N, \alpha_2 \in N, \alpha_3 \in N \cup \Delta_1 \cup \Delta_2, \beta_1 \in \Delta_1$,

$r_{m+4} : ([r_m]\alpha_1\beta_1\beta_2, \lambda/[r_m], \alpha_2\beta_2), \alpha_1, \alpha_2 \in N, \beta_2 \in \Delta_2, \beta_1 \in \Delta_1$,

$r_{m+5} : ([r_m]\alpha_1\beta_1\beta_2, \lambda/[r_m], \alpha_2\alpha_3\beta_2), \alpha_1 \in N, \alpha_2 \in N, \alpha_3 \in \Delta_1, \beta_1 \in \Delta_1, \beta_2 \in \Delta_2$,

$r_{m+6} : ([r_m]\alpha_1\beta_1, \lambda/[r_m], \alpha_2\alpha_3), \alpha_1, \alpha_2 \in N, \alpha_3 \in N \cup \Delta_1 \cup \Delta_3 \cup \Delta_4, \beta_1 \in \Delta_2$.

- $Lab = \{r_i^1, r_i^2, r_i^3, r_i^4, r_i^5, r_i^6 \mid [r_i] \in \Delta_1\}$

  $\cup\{r_i^7, r_i^8, r_i^9, r_i^{10}, r_i^{11}, r_i^{12}, r_i^{13} \mid [r_i] \in \Delta_2\}$

  $\cup\{a_i^1, a_i^2, a_i^3, a_i^4, a_i^5, a_i^6, r_{m+1}^i \mid [r_i] \in \Delta_3\}$

  $\cup\{r_i^{14}, r_i^{15}, r_i^{16}, r_i^{17}, r_i^{18}, r_i^{19}, r_i^{20}, r_{m+2}^i \mid [r_i] \in \Delta_4\}$

  $\cup\{r_m, r_{m+2}, r_{m+1}, r_{m+3}, r_{m+4}, r_{m+5}, r_{m+6}\}$.

The homomorphism $h : (Lab)^* \to T^*$ is defined by $h(a_i^1) = h(a_i^2) = h(a_i^3) = h(a_i^4) = f(a_i^5) = f(a_i^6) = f(a_i^7) = a$ and $h(l) = \lambda$ for $l \in Lab \setminus \{a_i^1, a_i^2, a_i^3, a_i^4, a_i^5, a_i^6, a_i^7\}$.

In this proof we construct an insertion grammar with labelled rules such that any recursively enumerable language can be obtained as a homomorphic image of the Szilard language of the labelled insertion grammar. Moreover, the labelled insertion grammar Γ is constructed in such a manner that each terminal derivation in G can be properly simulated by the rules in R_1. We also show that no word except the words in $L(G)$ can be obtained as the homomorphic image of the Szilard language of the labelled insertion grammar Γ. In this proof, the rules in (R_{11}) and (R_{12}) simulate the nonterminal rules and the rules in (R_{13}) and (R_{14}) simulate the terminal rules. The rules in (R_{15}) are constructed to simulate the leftmost derivations.

The working of the rules $r_i : A \to BC$, $r_i : AB \to CD$, $r_i : A \to a$ and $r_i : A \to \lambda$ has been discussed in the sections **(I), (II), (III)** and **(IV)** respectively.

(I) The simulation of the rule $r_i : A \to BC$ in different contexts is as follows:

$XwAY \to^{r_i^1} XA[r_i]BCY, w \in V_1^*$

$XwA\alpha Y \to^{r_i^2} XwA[r_i]BC\alpha Y, \alpha \in N, w \in V_1^*$

$XwA\alpha_1\alpha_2 Y \to^{r_i^3} XwA[r_i]BC\alpha_1\alpha_2 Y, \alpha_1, \alpha_2 \in N, w \in V_1^*$

$XwA\alpha_1\alpha_2\alpha_3 Y \to^{r_i^4} XwA[r_i]BC\alpha_1\alpha_2\alpha_3 Y, \alpha_1, \alpha_2, \alpha_3 \in N, w \in V_1^*$

$Xw_1A\alpha_1\alpha_2\alpha_3\alpha_4 w_2 Y \to^{r_i^5} XwA[r_i]BC\alpha_1\alpha_2\alpha_3\alpha_4 Y, w_1, w_2 \in V_1^*$,

where $\alpha_1 \in N, \alpha_2 \in N \cup \Delta_1, \alpha_3 \in N \cup \Delta_1 \cup \Delta_2, \alpha_4 \in N \cup \Delta_1 \cup \Delta_2$,

$\alpha_2\alpha_3 \notin (\Delta_1)(\Delta_1 \cup \Delta_2), \alpha_3\alpha_4 \notin (\Delta_1 \cup \Delta_2)(\Delta_1 \cup \Delta_2)$.

Note that after application of the rules $r_i^1, r_i^2, r_i^3, r_i^4$ and r_i^5, the subword $A[r_i]$ becomes inactive (i.e., no rules can be applied to $A[r_i]$) and becomes active once again when a word $Xw_1[r_m]A[r_i]w_2Y$, w_1, $w_2 \in V_1^+$ is obtained. The r_{m+3}-rule only can be applied to $Xw_1[r_m]A[r_i]w_2Y$. It has been discussed in **Case** 3.

(II) Simulation of the rules $r_i : AB \to CD$ in different contexts:

$XwABY \to^{r_i^8} XwAB[r_i]CDY, w \in V_1^+$. ... (1)

$XwAB\alpha Y \to^{r_i^9} XwAB[r_i]CD\alpha Y, \alpha \in N, w \in V_1^+$. ... (2)

$Xw_1AB\alpha_1\alpha_2 w_2 Y \to^{r_i^7} Xw_1AB[r_i]CD\alpha_1\alpha_2 w_2 Y, w_1 \in V_1^+, w_2 \in V_1^*, \alpha_1, \alpha_2 \in N$. ... (3)

Similarly as above, after application of r_i^8, r_i^9 and r_i^7-rule, no other rule can be applied to the subword $AB[r_i]$. Moreover, this subword cannot be further used for simulation of any other rule $r_j : A_1A \to B_1C_1$. In fact, the r_i^7, r_i^8 and r_i^9-rule cannot be applied to the subword $A_1AB[r_i]CD$ of the word $Xw_1A_1AB[r_i]CDw_2Y, w_1, w_2 \in V_1^*$. Hence, no rule in (R_{12}) is applicable to the subword $AB[r_i]$. So, the subword $AB[r_i]$ becomes inactive. It becomes active once again when the word $Xw_1[r_m]$ $AB[r_i]w_2Y, w_1 \in V_1^+, w_2 \in V_1^+$ is obtained. This has been discussed futher in **Case** 4.

In Case 1 and Case 2 we discuss the simulation of the rule $r_i : AB \rightarrow CD$ in the contexts different than the contexts discussed above.

Case 1: Now, we discuss the simulation of the rule $r_i : AB \rightarrow CD$ in the word

$Xw_1AwB\alpha_{n+1}w_2Y$ where $w_1 \in V_1^*, w_2 \in V_1^*, w = \alpha_1[r_1]\alpha_2[r_2]\dots\alpha_n[r_n], \alpha_i \in N, \alpha_{n+1} \in N, r_j \in \Delta_1$. ...(4)

If $w = \alpha_1[r_1]$ where $\alpha_1 \in N, [r_1] \in \Delta_1$, then

$Xw_1A\alpha_1[r_1]B\alpha_{n+1}w_2Y$

$\rightarrow^{r_i^{10}} Xw_1A[r_i]\alpha_1[r_1]B\alpha_{n+1}w_2Y$

$\rightarrow^{r_i^{11}} Xw_1A[r_i]\alpha_1[r_1][r_i]B\alpha_{n+1}w_2Y$

$\rightarrow^{r_i^{12}} Xw_1A[r_i]\alpha_1[r_1][r_i]B[r_i]CD\alpha_{n+1}w_2Y.$ ...(5)

If $w = \alpha_1[r_1]\alpha_2[r_2]$ where $\alpha_1, \alpha_2 \in N, [r_1], [r_2] \in \Delta_1$, then

$Xw_1A\alpha_1[r_1]\alpha_2[r_2]B\alpha_{n+1}w_2Y$

$\rightarrow^{r_i^{10}} Xw_1A[r_i]\alpha_1[r_1]\alpha_2[r_2]B\alpha_{n+1}w_2Y$

$\rightarrow^{r_i^{11}} Xw_1A[r_i]\alpha_1[r_1][r_i]\alpha_2[r_2]B\alpha_{n+1}w_2Y$

$\rightarrow^{r_i^{11}} Xw_1A[r_i]\alpha_1[r_1][r_i]\alpha_2[r_2][r_i]B\alpha_{n+1}w_2Y$

$\rightarrow^{r_i^{12}} Xw_1A[r_i]\alpha_1[r_1][r_i]\alpha_2[r_2][r_i]B[r_i]CD\alpha_{n+1}w_2Y.$ ...(6)

Hence, the application of the r_{10}-rule is followed by repeated application of the r_{11}-rule and then one time application of r_{12}-rule can simulate the application of the rule $r_i : AB \rightarrow CD$ on the words in (4). Moreover, the above derivations can be further extended for words $Xw_1A\alpha_1[r_1]\alpha_2[r_2]\dots\alpha_n[r_n]\ B\alpha_{n+1}$ w_2Y where $n \geq 3, \alpha_i \in N(1 \leq i \leq n+1), w_1 \in V_1^*, w_2 \in V_1^*, [r_j] \in \Delta_1(1 \leq j \leq n)$.

Also, for each $r_l : A_1 \rightarrow B_1C_1$, the corresponding r_l^5- rule is applicable to the above word $Xw_1^{'}A_1A$ $\alpha_1[r_1]\alpha_2[r_2]\dots\alpha_n[r_n]B\ \alpha_{n+1}w_2Y$ where $w_1 = w_1^{'}A_1, w_1^{'} \in V_1^*, A_1 \in N$. The r_l^5-rule can change it into $Xw_1^{'}A_1[r_l]B_1\ C_1A\alpha_1[r_1]\alpha_2[r_2]\dots\alpha_n[r_n]B\ \alpha_{n+1}w_2Y$. But if the r_i^{10}-rule is applied at first as in Case 1, then $[r_i] \in \Delta_2$ is inserted into the word and $Xw_1^{'}A_1A[r_i]\ \alpha_1[r_1]\alpha_2[r_2]\ \dots\alpha_n[r_n]B\alpha_{n+1}\ w_2Y$ is obtained. This forbids the insertion rules in (R_{11}) to be applied to the subword $A_1A[r_i]\alpha_1[r_1]\alpha_2[r_2]\ \dots\alpha_n[r_n]B$.

Again, if there exist a rule $r_l^{'} : B \rightarrow C^{'}D^{'}$ and the corresponding insertion rule in (R_{11}) is applied to the word $Xw_1A[r_i]\alpha_1[r_1][r_i]\alpha_2\ [r_2][r_i]\dots\alpha_n[r_n][r_i]B\alpha_{n+1}w_2Y$, then the word $Xw_1A[r_i]\alpha_1[r_1][r_i]$ $\alpha_2[r_2][r_i]\dots\alpha_n[r_n][r_i]B[r_l]C^{'}D^{'}\alpha_{n+1}w_2Y$ is obtained. The r_i^{12}-rule is not applicable to it. Hence, if the simulation of the rule $r_i : AB \rightarrow CD$ has started, then no other rule can be applied to the subword $A[r_i]\alpha_1[r_1][r_i]\alpha_2\ [r_2][r_i]\dots\alpha_n[r_n][r_i]B[r_i]$ except the r_i^{10}, r_i^{11} and r_i^{12}-rule.

Moreover, once the simulation is complete, the subword $A[r_i]\alpha_1[r_1][r_i]\alpha_2\ [r_2][r_i]\dots\alpha_n[r_n][r_i]B\ [r_i]$ becomes inactive and will be active again when the subword $[r_m]A[r_i]\alpha_1[r_1][r_i]\alpha_2\ [r_2][r_i]\dots\alpha_n\ [r_n]\ [r_i]B[r_i]$ is obtained. We discuss about it in **Case** 5.

Case 2: In this case we discuss the simulation of consecutive application of two rules in (R_{12}).

If a word of the form $Xw_1A_1B_1A[r_i]\alpha_1[r_1][r_i]\alpha_2\ [r_2][r_i]\ldots\alpha_n[r_n][r_i]\ B[r_i]\ w_2Y, w_1 \in V_1^*, w_2 \in V_1^+$ is obtained during any stage of the computation and also there exist a rule $r_j : A_1B_1 \rightarrow C_1D_1$, then the application of the rule can be simulated by r_j^{13}-rule.

$Xw_1A_1B_1A[r_i]\alpha_1[r_1][r_i]\alpha_2\ [r_2][r_i]\ldots\alpha_n[r_n][r_i]B[r_i]w_2Y$

$\rightarrow^{r_j^{13}} Xw_1A_1B_1[r_j]C_1D_1A[r_i]\alpha_1[r_1][r_i]\alpha_2\ [r_2][r_i]\ldots\alpha_n[r_n][r_i]B[r_i]w_2Y.$

$\ldots(7)$

Also, since no rule from $(R_{11}), (R_{12}), (R_{13})$ and (R_{14}) is applicable to the subword $A_1B_1[r_j]$, it becomes inactive after the simulation. It becomes active again when the word $Xw_1'[r_m]\ A_1B_1[r_j]\ w_2'Y, w_1', w_2' \in V_1^+$ is obtained.

(III) Now, we discuss the Simulation of the rule $r_i : A \rightarrow a$ in different contexts:

$XAY \rightarrow^{a_i^1} XAk_a^iY,$

$Xw[r_m]AY \rightarrow^{a_i^2} Xw[r_m]Ak_a^iY, w \in V_1^+,$

$Xw[r_m]A\alpha_1Y \rightarrow^{a_i^3} Xw[r_m]Ak_a^i\alpha_1Y, w \in V_1^+, \alpha_1 \in N,$

$Xw[r_m]A\alpha_1\alpha_2Y \rightarrow^{a_i^4} Xw[r_m]Ak_a^i\alpha_1\alpha_2Y, w \in V_1^+, \alpha_1, \alpha_2 \in N,$

$Xw[r_m]A\alpha_1\alpha_2\alpha_3Y \rightarrow^{a_i^5} Xw[r_m]Ak_a^i\alpha_1\alpha_2\alpha_3Y, w \in V_1^+, \alpha_1, \alpha_2, \alpha_3 \in N,$

$Xw[r_m]A\alpha_1\alpha_2\alpha_3\alpha_4Y \rightarrow^{a_i^6} Xw[r_m]Ak_a^i\alpha_1\alpha_2\alpha_3\alpha_4Y, w \in V_1^+,$

where $\alpha_1 \in N, \alpha_2 \in N \cup \Delta_1, \alpha_3 \in N \cup \Delta_1 \cup \Delta_2, \alpha_4 \in N \cup \Delta_1 \cup \Delta_2,$

$\alpha_2\alpha_3 \notin (\Delta_1)(\Delta_1 \cup \Delta_2), \alpha_3\alpha_4 \notin (\Delta_1 \cup \Delta_2)(\Delta_1 \cup \Delta_2).$

(IV) Again, the simulation of the rule $r_i : A \rightarrow \lambda$ in different contexts is as follows:

$XAY \rightarrow^{r_i^{14}} XAk_\lambda^iY,$

$Xw[r_m]AY \rightarrow^{r_i^{15}} Xw[r_m]Ak_\lambda^iY, w \in V_1^+,$

$Xw[r_m]A\alpha_1Y \rightarrow^{r_i^{16}} Xw[r_m]Ak_\lambda^i\alpha_1Y, w \in V_1^+, \alpha_1 \in N,$

$Xw[r_m]A\alpha_1\alpha_2Y \rightarrow^{r_i^{17}} Xw[r_m]Ak_\lambda^i\alpha_1\alpha_2Y, w \in V_1^+, \alpha_1, \alpha_2 \in N,$

$Xw[r_m]A\alpha_1\alpha_2\alpha_3Y \rightarrow^{r_i^{18}} Xw[r_m]Ak_\lambda^i\alpha_1\alpha_2\alpha_3Y, w \in V_1^+, \alpha_1, \alpha_2, \alpha_3 \in N,$

$Xw_1[r_m]A\alpha_1\alpha_2\alpha_3\alpha_4w_2Y \rightarrow^{r_i^{19}} Xw[r_m]Ak_\lambda^i\alpha_1\alpha_2\alpha_3\alpha_4Y, w_1, w_2 \in V_1^+,$

where $\alpha_1 \in N, \alpha_2 \in N \cup \Delta_1, \alpha_3 \in N \cup \Delta_1 \cup \Delta_2, \alpha_4 \in N \cup \Delta_1 \cup \Delta_2,$

$\alpha_2\alpha_3 \notin (\Delta_1)(\Delta_1 \cup \Delta_2), \alpha_3\alpha_4 \notin (\Delta_1 \cup \Delta_2)(\Delta_1 \cup \Delta_2).$

Now, we discuss the simulations of the leftmost derivation process in Γ in the following cases in detail. To simulate the leftmost derivations in G, the rules in (R_{15}) are constructed. They are constructed in such a way that the symbol $[r_m]$ identifies the leftmost non-terminal in the word $XwY, w \in V_1^+$. The working of the rules in (R_{15}) has been discussed in detail in the **Case** 3 to **Case** 6.

At first, the simulation of the leftmost derivations start with the application of the r_m-rule.

$X\alpha[r_j]w_2Y \rightarrow^{r_m} X\alpha[r_j][r_m]w_2Y$, where $\alpha \in N, w_2 \in V_1^+, [r_j] \in \Delta_1.$ $\ldots(8)$

Case 3: The subword $A[r_i]$ in the word $Xw_1[r_m]A[r_i]w_2Y, w_1 \in V_1^+, w_2 \in V_1^+$ becomes active again when the subword $[r_m]A[r_i]$ is obtained and is followed by the following step:

$Xw_1[r_m]A[r_i]w_2Y \rightarrow^{r_{m+3}} Xw_1[r_m]A[r_i][r_m]w_2Y.$ $\ldots(9)$

Case 4:

The subword $AB[r_i]$ in **(II)** will be activated once again after obtaining the subword $[r_m]AB[r_i]$. It is followed by insertion of the symbol $[r_m]$ into $Xw_1[r_m]\alpha_k[r_k]AB[r_i]w_2Y, w_1, w_2 \in V_1^+$.

$Xw_1[r_m]\alpha_k[r_k]AB[r_i]w_2Y$

$\rightarrow^{r_{m+3}} Xw_1[r_m]\alpha_k[r_k][r_m]AB[r_i]w_2Y$

$\rightarrow^{r_{m+1}} Xw_1[r_m]\alpha_k[r_k][r_m]AB[r_i][r_m]w_2Y$,

where $\alpha_k \in N, w_1, w_2 \in V_1^+, [r_k] \in \Delta_1, [r_i] \in \Delta_2$. ... (10)

Moreover, in (10) after application of the r_{m+3}-rule only the r_{m+1}-rule is applicable to $AB[r_i]$.

Case 5:

In this case we discuss how the insertion rules insert $[r_m]$ such that the subword $A[r_i]\alpha_1[r_1][r_i]\alpha_2[r_2][r_i]\ldots\alpha_n[r_n][r_i]B[r_i]$ becomes active again in $Xw_1[r_m]\alpha_0[r_0]A[r_i]\alpha_1[r_1]\alpha_2[r_2][r_i]\ldots\alpha_n[r_n]B\ [r_i]\ w_2Y$ where $[r_j] \in \Delta_1 (j = 1, 2, \ldots, n), [r_i] \in \Delta_2, w_1 \in V_1^+,\ w_2 \in V_1^+$ and $r_i : AB \rightarrow CD$.

At first we explain the derivation for the word $Xw_1[r_m]\alpha_0[r_0]A[r_i]\alpha_1[r_1]B[r_i]\ w_2Y$.

$Xw_1[r_m]\alpha_0[r_0]A[r_i]\alpha_1[r_1][r_i]B[r_i]w_2Y$

$\rightarrow^{r_{m+3}} Xw_1[r_m]\alpha_0[r_0][r_m]A[r_i]\alpha_1[r_1][r_i]B[r_i]w_2Y$

$\rightarrow^{r_{m+2}} Xw_1[r_m]\alpha_0[r_0][r_m]A[r_i][r_m]\alpha_1[r_1][r_i]B[r_i]w_2Y$

$\rightarrow^{r_{m+4}} Xw_1[r_m]\alpha_0[r_0][r_m]A[r_i][r_m]\alpha_1[r_1][r_i][r_m]B[r_i]w_2Y$

$\rightarrow^{r_{m+6}} Xw_1[r_m]\alpha_0[r_0][r_m]A[r_i][r_m]\alpha_1[r_1][r_i][r_m]B[r_i][r_m]w_2Y$. ... (11)

Next we describe the above process for the word $Xw_1[r_m]\alpha_0[r_0]A[r_i]\alpha_1[r_1][r_i]\ \alpha_2[r_2][r_i]B[r_i]\ w_2Y, w_1 \in V_1^+, w_2 \in V_1^+, [r_i] \in \Delta_2, [r_j] \in \Delta_1 (1 \leq j \leq 3)$.

$Xw_1[r_m]\alpha_0[r_0]A[r_i]\alpha_1[r_1][r_i]\alpha_2[r_2][r_i]B[r_i]w_2Y$

$\rightarrow^{r_{m+3}} Xw_1[r_m]\alpha_0[r_0][r_m]A[r_i]\alpha_1[r_1][r_i]\alpha_2[r_2][r_i]B[r_i]w_2Y$

$\rightarrow^{r_{m+2}} Xw_1[r_m]\alpha_0[r_0][r_m]A[r_i][r_m]\alpha_1[r_1][r_i]\alpha_2[r_2][r_i]B[r_i]w_2Y$

$\rightarrow^{r_{m+5}} Xw[r_m]\alpha_0[r_0][r_m]A[r_i][r_m]\alpha_1[r_1][r_i][r_m]\alpha_2[r_2][r_i]B[r_i]w_2Y$

$\rightarrow^{r_{m+4}} Xw_1[r_m]\alpha_0[r_0][r_m]A[r_i][r_m]\alpha_1[r_1][r_i][r_m]\alpha_2[r_2][r_i][r_m]B[r_i]w_2Y$

$\rightarrow^{r_{m+6}} Xw_1[r_m]\alpha_0[r_0][r_m]A[r_i][r_m]\alpha_1[r_1][r_i][r_m]\alpha_2[r_2][r_i][r_m]B[r_i][r_m]w_2Y$.

... (12)

The above derivations can be further extended for $Xw_1[r_m]\alpha_0[r_0]A[r_i]\alpha_1[r_1]\alpha_2[r_2]\ldots\alpha_n\ [r_n]B\ w_2Y$ where $\alpha_i \in N (0 \leq i \leq n), [r_j] \in \Delta_1 (0 \leq j \leq n), [r_i] \in \Delta_2$. In this case, the symbol $[r_m]$ can be inserted in specified location by application of the rules with label $r_{m+2}, r_{m+3}, r_{m+4},\ r_{m+5}$ and r_{m+6}.

Moreover, after application of the r_{m+3}-rule to $Xw_1[r_m]\alpha_0[r_0]A[r_i]\alpha_1\ [r_1]\alpha_2[r_2]\ldots\alpha_n[r_n]Bw_2Y$, the word $Xw_1\ [r_m]\alpha_0[r_0][r_m]A[r_i]\alpha_1[r_1]\alpha_2[r_2]\ldots\alpha_n[r_n]Bw_2Y$ is obtained. No insertion rule except the rules in the above derivation can be applied further to the subword $[r_m]\alpha_0[r_0][r_m]A[r_i]\alpha_1[r_1]\alpha_2[r_2]\ldots\alpha_n[r_n]B$. Hence, for further steps a subwords of the form $A[r_i]\alpha_1[r_1][r_i]\ldots\ \alpha_n[r_n][r_i]\ B[r_i]$ must be obtained whenever $r_i : AB \rightarrow CD$ is simulated.

Case 6:

Now we discuss the application of the rules r_i^6, a_i^7 and r_i^{20} where $r_i : AB \rightarrow CD, r_i : A \rightarrow a, r_i : A \rightarrow \lambda$.

The word $Xw_1A[r_i]\alpha_1[r_1][r_i]\ldots\alpha_n[r_n][r_i]B[r_i]w_2Y$ is obtained after simulation of the rule $r_i : AB \rightarrow$

CD in case (1) of **(II)**. Moreover, if $w_1 = w_1^{'} A_1, w_1^{'} \in V_1^*, A_1 \in N$ and there exist a rule $r_k : A_1 \to B_1 C_1$, then

$Xw_1^{'} A_1 A[r_i]\alpha_1[r_1][r_i] \ldots \alpha_n[r_n][r_i]B[r_i]w_2Y \to^{r_i^6}$
$Xw_1^{'} A_1[r_k]B_1C_1A[r_i]\alpha_1[r_1][r_i] \ldots \alpha_n[r_n][r_i]B[r_i]w_2Y, w_1^{'} \in V_1^*, w_2 \in V_1^+.$ ...(13)

The rule $r_j : A_1 \to a$ can be simulated in the following manner when the subword $[r_m]A_1A[r_i]\alpha_1 [r_1][r_i] \ldots \alpha_n[r_n]B[r_i]$ is obtained.

$Xw_1[r_m]A_1A[r_i]\alpha_1[r_1][r_i] \ldots \alpha_n[r_n][r_i]B[r_i]w_2Y \to^{a_j^7}$
$Xw_1[r_m]A_1k_a^jA[r_i]\alpha_1[r_1][r_i] \ldots \alpha_n[r_n][r_i]B[r_i]w_2Y.$ ...(14)

Similarly, r_l^{20}-rule can simulate $r_l : A_1 \to \lambda$ as in (14). Moreover, the subwords $A_1[r_k]$ becomes inactive. But there exist insertion rules which can be applied to $[r_m]A_1k_a^j$ and $[r_m]A_1k_\lambda^l$. It has been discussed in **Case 7**.

Case 7:

The application of a_j^7-rule and r_l^{20}-rule is followed by the application of the insertion rules with label r_{m+1}^j and r_{m+2}^l to proceed further. Note that no other insertion rule is applicable to the subwords $[r_m]A_1k_a^jA[r_i]A[r_i]\alpha_1[r_1][r_i] \ldots \alpha_n[r_n][r_i]B[r_i]$ and $[r_m]A_1k_\lambda^lA[r_i]\alpha_1[r_1][r_i] \ldots \alpha_n[r_n][r_i]B[r_i]$.

$Xw_1[r_m]A_1k_a^jA[r_i]\alpha_1[r_1][r_i] \ldots \alpha_n[r_n][r_i]B[r_i]w_2Y \to^{r_{m+1}^j}$
$Xw_1[r_m]A_1k_a^j[r_m]A[r_i]\alpha_1[r_1][r_i] \ldots \alpha_n[r_n][r_i]B[r_i]w_2Y.$ ...(15)

$Xw_1[r_m]A_1k_\lambda^lA[r_i]\alpha_1[r_1][r_i] \ldots \alpha_n[r_n][r_i]B[r_i]w_2Y \to^{r_{m+2}^l}$
$Xw_1[r_m]A_1k_\lambda^l[r_m]A[r_i]\alpha_1[r_1][r_i] \ldots \alpha_n[r_n][r_i]B[r_i]w_2Y.$ ...(16)

In fact, the r_{m+1}^j and r_{m+2}^l-rule are applicable to words $Xw_1[r_m]A_1k_a^j\ w_2Y$ and $Xw_1[r_m]A_1k_\lambda^l\ w_2Y$, respectively where $w_1, w_2 \in V_1^+$. Also, no insertion rules are applicable to the subwords $[r_m]A_1k_a^j$ and $[r_m]A_1k_\lambda^l$ further.

From the above discussions we can say that whenever the non-terminal rules are applied, corresponding labelled insertion rules in (R_{11}) and (R_{12}) can simulate it properly in the word $XwY, w \in V_1^+$. Also, the application of the rules in $(R_{11}), (R_{12}), (R_{13})$ and (R_{14}) , inactivates the subwords $A[r_i], AB[r_i], A[r_i]\alpha_1[r_1][r_i]\alpha_2[r_2][r_i] \ldots \alpha_n[r_n][r_i]B[r_i]$ and $A[r_i]$ to be reactivated again by application of the rules in (R_{15}). Moreover, these subwords become active again only during the simulation of leftmost derivations by the rules in $(R_{13}), (R_{14})$ and (R_{15}). Hence, no extra derivation is possible in Γ.

Now at first we prove the inclusion $L(G) \subseteq h(SZINS_5(\Gamma))$. Let $w \in L(G)$. If the insertion rules in Γ are applied in the same order as in a derivation of G obtaining w, a string over V_1 is obtained where no insertion rule can be applied further. Also, if the labels of the applied rules are concatenated, a string over Lab is obtained. If the morphism h is applied to the string over Lab, then the terminal string $w \in L(G)$ is obtained. Hence, $L = L(G) \subseteq h(SZINS_5(\Gamma))$.

Now we prove the inclusion $h(SZINS_5(\Gamma)) \subseteq L(G)$, i.e., no word except the elements of $L(G)$ can be obtained as homomorphic image of the Szilard language of insertion grammar of weight 5.

Let $x \in h(SZINS_5(\Gamma))$. Then there exists a $x_1 \in SZINS_5(\Gamma)$ such that $x = h(x_1)$. The string x_1 is obtained when the labels of the rules in a terminal derivation of Γ are concatenated. In the string

$x_1 \in Lab^*$, all the symbols except the symbols $a_i^1, a_i^2, a_i^3, a_i^4, a_i^5, a_i^6$ and a_i^7 for each rule $r_i : A \rightarrow a$ are mapped to λ by the morphisn h. Since no extra terminal derivation is possible in Γ, the string x is obtained when the rules of the grammar G are applied in the same order. So, $x \in L$. Hence, $h(SZINS_5(\Gamma)) \subseteq L(G)$. □

5 Conclusion

In this work we investigated Szilard languages obtained by the labelled insertion grammars and compared them with the family of languages in Chomsky hierarchy. We showed that there exist regular languages which cannot be obtained as a Szilard language by any labelled insertion grammar. But every regular, context-free and recursively enumerable language can be obtained as a homomorphic image of the Szilard language of labelled insertion grammars with some restricted bounds. The bounds obtained in this paper are not optimal. One of the future direction of research can be to obtain the optimal bounds of these results.

References

[1] Rozenberg, Grzegorz and Salomaa, Arto, Handbook of Formal Languages, Vol. 1: Word, Language, Grammar, Berlin, Heidelberg, Springer-Verlag, 1997.

[2] Lila and Petr Sosík, On the weight of universal insertion grammars, Theor. Comput. Sci., 1-3, 396, 264–270, 2008.

[3] Carlos Martín-Vide and Gheorghe Păun and Arto Salomaa, Characterizations of Recursively Enumerable Languages by Means of Insertion Grammars, Theor. Comput. Sci., 1-2, 205, 195–205, 1998.

[4] Harvey Rubin and David Harlan Wood, DIMACS/AMS, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, DNA Based Computers, Proceedings of a DIMACS Workshop, Philadelphia, Pennsylvania, USA, June 23-25, 1997, 48, 1999.

[5] Gheorghe Păun and Grzegorz Rozenberg and Arto Salomaa, DNA Computing - New Computing Paradigms, Springer, Texts in Theoretical Computer Science. An EATCS Series, 1998.

[6] Gheorghe Păun, Two theorems about Galiukschov semicontextual languages, Kybernetika, 5, 21, 360–365, 1985.

[7] Author = Maurice Margenstern and Gheorghe Păun and Yurii Rogozhin and Sergey Verlan, Context-free insertion-deletion systems, Theoretical Computer Science, 2, 330, 339–348, 2005.

[8] Viliam Geffert and Giovanni Pighizzini, 9th International Workshop on Descriptional Complexity of Formal Systems - DCFS 2007, High Tatras, Slovakia, July 20 - 22, 2007, University of Kosice, Slovakia, 2007.

[9] Sergey Verlan, On Minimal Context-Free Insertion-Deletion Systems, Journal of Automata, Languages and Combinatorics, 1-2, 12, 317–328, 2007.

[10] Carlos Martín-Vide and Friedrich Otto and Henning Fernau, Language and Automata Theory and Applications, Second International Conference, LATA 2008, Tarragona, Spain, March 13-19, 2008. Revised Papers, Springer, Lecture Notes in Computer Science, 5196, 2008.

[11] Jérôme Olivier Durand-Lose and Maurice Margenstern, Machines, Computations, and Universality, 5th International Conference, MCU 2007, Orléans, France, September 10-13, 2007, Proceedings, Springer, Lecture Notes in Computer Science, 4664, 2007.

[12] Henning Fernau and Lakshmanan Kuppusamy and Sergey Verlan, Universal Matrix Insertion Grammars with Small Size, Unconventional Computation and Natural Computation - 16th International Conference, UCNC 2017, Fayetteville, AR, USA, June 5-9, 2017, Proceedings, 182–193, 2017.

[13] Kalpana Mahalingam and Prithwineel Paul and Erkki Mäkinen, On Derivation Languages of a Class of Splicing Systems, Acta Cybern., 4, 1–13, 23, 2018.

[14] Liliana Cojocaru and Erkki Mäkinen, On some derivation mechanisms and the complexity of their Szilard languages, Theor. Comput. Sci., 87–96, 537, 2014.

[15] Liliana Cojocaru and Erkki Mäkinen, The Complexity of Szilard Languages of Matrix Grammars Revisited, Fundam. Inform., 4, 381–399, 123, 2013.

[16] Liliana Cojocaru and Erkki Mäkinen, On the Complexity of Szilard Languages of Matrix Grammars, 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2011, Timisoara, Romania, September 26-29, 2011, 339–347, 2011.

[17] Erkki Mäkinen, On Context-Free and Szilard Languages, BIT, 2, 24, 164–170, 1984.

[18] Lila Kari and Gabriel Thierrin, Contextual Insertions/Deletions and Computability, Information and Computation, 1, 131, 47–61, 1996.

[19] K Onodera, A note on homomorphic representation of recursively enumerable languages with insertion grammars, IPSJ Journal, 5, 44, 1424–1427, 2003.

[20] A Krassovitskiy, On the power of Insertion P Systems of small size, International Journal of Computers, Communications and Controls, 2, 6, 266–277, 2011.

[21] K Mahalingam and P Paul and B Song and L Pan and K. G. Subramanian, Derivation languages of Splicing P systems, BIC-TA 2017, CCIS, 487–501, 791, 2017.

[22] E Mäkinen, On homomorphic images of Szilard languages, International Journal of Computer Mathematics, 239–245, 18, 1986.

[23] Artiom Matveevici and Yurii Rogozhin and Sergey Verlan, Insertion-Deletion Systems with One-Sided Contexts, Machines, Computations, and Universality, 5th International Conference, MCU 2007, Orléans, France, September 10-13, 2007, Proceedings, 205–217, 2007.

[24] G Păun, On some families of Szilard languages, Bull. Math. de la Soc. Sci. Math. de la R. S. de Roumanie Tome, 75, 27, 259–265, 1983.

[25] S Marcus, Contextual grammars, Revue Roumaine de Mathematiques Pures et Appliquees, 1525–1534, 14, 1969.

[26] C.C Squier, Semicontextual grammars: An example, Bull. Math. Soc. Sci. Math Roumanie, 80, 32, 167–170, 1988.

[27] G Păun, On semicontextual grammars, Bull. Math. Soc. Sci. Math. Roumanie, 76, 28, 63–68, 1984.

[28] B. S. Galiukschov, Semicontextual grammars, Mathematika Logicia i Mathematika Linguistika, Tallin University, 38–50, 1981.

[29] Author = S Verlan, Recent developments on insertion-deletion systems, Computer Science Journal of Moldova, 2, 18, 210–245, 2010.

[30] Akihiro Takahara and Takashi Yokomori, On the computational power of insertion-deletion systems, Natural Computing, 4, 2, 321–336, 2003.

[31] Artiom Alhazov and Alexander Krassovitskiy and Yurii Rogozhin and Sergey Verlan, P Systems with Insertion and Deletion Exo-Operations, Fundam. Inform., 1-4, 110, 13–28, 2011.

[32] Gheorghe Păun and Mario J. Pérez-Jiménez and Takashi Yokomori, Representations and Characterizations of Languages in Chomsky Hierarchy by Means of Insertion-Deletion Systems, 9th International Workshop on Descriptional Complexity of Formal Systems - DCFS 2007, High Tatras, Slovakia, July 20 - 22, 2007, 129–140, 2007.

[33] Alexander Krassovitskiy and Yurii Rogozhin and Sergey Verlan, Further Results on Insertion-Deletion Systems with One-Sided Contexts, Language and Automata Theory and Applications, Second International Conference, LATA 2008, Tarragona, Spain, March 13-19, 2008. Revised Papers, 10.1007/978-3-540-88282-4_31, 333–344, 2008,

[34] Lila Kari and Gheorghe Păun and Gabriel Thierrin and Sheng Yu, At the crossroads of DNA computing and formal languages: Characterizing recursively enumerable languages using insertion-deletion systems, DNA Based Computers, Proceedings of a DIMACS Workshop, Philadelphia, Pennsylvania, USA, June 23-25, 1997, 329–346, 1997.