On the Generative Power of Simple H Systems

Lakshminarayanan Subramanian
Muralidhar Talupur
Kamala Krithivasan
C. Pandu Rangan

Department of Computer Science and Engineering
Indian Institute of Technology, Madras
Chennai-600 036, India

Abstract

In this paper, we prove that the power of Simple H-systems of the (2,3) type with permitting contexts and target alphabet is equal to Extended H-systems with permitting contexts and radius of the rules equal to one. We also prove interesting results on Simple Extended H-systems and Extended H-systems with forbidden contexts.

Keywords: Splicing systems, simple H systems, permitting and forbidden contexts and cardinality of context.

1 Introduction

Tom Head [4] initiated a new appealing branch of formal language theory called Splicing Systems. The basic notion is that of *splicing*, a formal model of the recomb inant behavior of DNA sequences under the influence of restriction enzymes and lygases. A slight modification of this system was called as H-system by Paun [5].

By adding the notion of terminal alphabet to a H-system, we obtain an extended H-system [5, 9]. The power of such a system, with the set of splicing rules forming a regular language, turns out to be very large; these systems characterize the family of recursively enumerable languages [1, 7]. In this paper, we concentrate on a specific extended H-system having the radius one.

In [6], the notion of Simple H-systems was introduced. The possibility of permitting contexts and target alphabet for Simple H-systems was studied in [2] and many interesting results were obtained. In this paper, we study SEH systems of the $(2, 3)$ type.

In this paper we prove that the power of SEH system of the $(2, 3)$ type with permitting contexts is equivalent to Extended H-system with radius equal to one and permitting contexts. We also prove interesting results for Simple Extended H-systems with forbidden contexts. This paper also defines a new term called the cardinality of context in Extended H-systems. We prove that cardinality of context adds no power to EH systems with permitting contexts but plays a very important role in forbidden contexts.

In section 2, we give the basic definitions. Section 3 describes the role of cardinality of context in Extended H-systems. In section 4, we prove that $SEH_{2,3}(p)$ is equal to $EH(FIN, p[1])$. In section 5, we prove an interesting result on SEH system of $(2, 3)$ type with forbidden contexts. In section 6, we present our conclusions.

2 Preliminaries

2.1 Extended H Systems

The splicing operation is a formal model of the DNA recombination under the effect of restriction enzymes. A splicing rule (over an alphabet V) is a string $r = u_1\#u_2\$u_3\#u_4$ where $u_1, u_2, u_3, u_4 \in V^*$ and $\#, \$$ are two special symbols not in V.

For $x, y, z, w \in V^*$ and r as above we write $(x, y) \vdash_r w$ iff $x = x_1u_1u_2x_2, y = y_1u_3u_4y_2, w = x_1u_1u_4y_2$ for some $x_1, x_2, y_1, y_2 \in V^*$.

We say that we splice x, y at the sites u_1u_2, u_3u_4. These sites encode the patterns recognized by restriction enzymes able to cut the DNA sequences between u_1, u_2, respectively between u_3, u_4. The radius of a splicing rule is the length of the longest string u_1, u_2, u_3, u_4.

An extended H system is a quadruple $\gamma = (V, T, A, R)$ where V is the total alphabet, $T \subseteq V$ is the target alphabet, $A \subset V^*$ represents a finite set of axioms and $R \subset V^*\#V^*\$V^*\#V^*$ is a set of splicing rules.

For any $L \subseteq V^*$ and $\gamma = (V, T, A, R)$ we define
$\sigma(L) = \{w|(x, y) \vdash_r w \text{ for } x, y \in L, r \in R\}$
$\sigma^0(L) = L$
$\sigma^{i+1}(L) = \sigma^i(L) \cup \sigma(\sigma^i(L)), i \geq 0$
$\sigma^*(L) = \cup_{i \geq 0}\sigma^i(L)$

The language generated by γ is
$L(\gamma) = \sigma^*(A) \cap T^*$

An Extended H-System with permitting contexts is a quadruple $\gamma = (V, T, A, R)$ where V, T, A are the same as defined earlier and R is a finite set of triples $p = (r = u_1\#u_2\$u_3\#u_4, C_1, C_2)$ where $C_1, C_2 \subseteq V$ and r is a usual splicing rule.

In this case $(x, y) \vdash_p w$ iff $(x, y) \vdash_r w$ and all symbols of C_1 appear in x and all symbols of C_2 occur in y.

An Extended H-System with forbidden contexts is a quadruple $\gamma = (V, T, A, R)$ where V, T, A are the same as defined earlier and R is a finite set of triples $p = (r = u_1\#u_2\$u_3\#u_4, C_1, C_2)$ where $C_1, C_2 \subseteq V$ and r is a usual splicing rule.

In this case $(x, y) \vdash_p w$ iff $(x, y) \vdash_r w$ and all symbols of C_1 do not appear in x and all symbols of C_2 do not occur in y.

$EH(FIN, p[k])$ refers to the family of languages generated by Extended H-Systems with permitting contexts, finite set of axioms and rules with maximum radius equal to k for $k \geq 1$. In a similar fashion, one can define $EH(FIN, f[k])$ to be the family of languages generated by Extended H systems with forbidden contexts, finite set of axioms and rules with maximum radius equal to k.

Let us define a new term **cardinality of context** to be the maximum size of a context in a rule in the Extended H system. An Extended H-system γ is said to have a cardinality of context equal to n if every rule $r = (p, C_1, C_2)$ satisfies the constraint $|C_1| \leq n$ and $|C_2| \leq n$ and n is the smallest integer with this property.

Let $EH(FIN, p[k, n])$ define the family of languages generated by Extended H systems with permitting contexts, finite set of axioms and rules with maximum radius equal to k and maximum cardinality of context equal to n. Similarly one can define $EH(FIN, f[k,n])$ for forbidden contexts.

In this paper we will investigate the properties of these languages and associate them with Simple H-Systems. We will prove that the cardinality of context plays no role in permitting contexts but has an important role in forbidden contexts.

2.2 Simple H Systems

A Simple H-System is a triple $\gamma = (V, A, M)$ where V is the total alphabet, A is a finite language over V and $M \subseteq V$. The elements of A are called axioms and those of M are called markers. In [6] where Simple H-Systems were introduced, one takes four ternary relations on the language V^* , corresponding to splicing rules of the form

$a\#\$a\#, \#a\$\#a, a\#\$\#a, \#a\$a\#$

where a is an arbitrary element of M. The rules listed above correspond to splicing rules of type $(1,3), (2,4), (1,4)$ and $(2,3)$ respectively. Clearly rules of types $(1,3)$ and $(2,4)$ define the same operation for $x, y, z \in V^*$ and $a \in M$. We obtain

$(x, y) \vdash^a_{(1,3) or (2,4)} z$ iff $x = x_1 a x_2, y = y_1 a y_2, z = x_1 a y_2$ for some $x_1, x_2, y_1, y_2 \in V^*$

For the $(1,4)$ and the $(2,3)$ types we have

$(x, y) \vdash^a_{(1,4)} z$ iff $x = x_1 a x_2, y = y_1 a y_2, z = x_1 a a y_2$ for some $x_1, x_2, y_1, y_2 \in V^*$

$(x, y) \vdash^a_{(2,3)} z$ iff $x = x_1 a x_2, y = y_1 a y_2, z = x_1 y_2$ for some $x_1, x_2, y_1, y_2 \in V^*$

Similar to Extended H-systems we define for a language $L \subseteq V^*$ and $(i,j) \in \{(1,3), (2,4), (1,4), (2,3)\}$. We denote

$\sigma_{(i,j)}(L) = \{z | z \in V^*, (x, y) \vdash^a_{(i,j)} z \text{ for } x, y \in L, a \in M\}$

Define

$\sigma^0_{(i,j)}(L) = L$

$\sigma^{k+1}_{(i,j)}(L) = \sigma^k_{(i,j)}(L) \cup \sigma_{(i,j)}(\sigma^k_{(i,j)}(L)), k \geq 0$

$\sigma^*_{(i,j)}(L) = \cup_{k \geq 0} \sigma^k_{(i,j)}(L)$

The language generated by γ with splicing rules of type (i,j) is defined as

$L_{(i,j)}(\gamma) = \sigma^*_{(i,j)}(A)$

One can visualize an extension to Simple H-Systems with permitting contexts and terminal alphabet. A Simple H-System with terminal alphabet is one in which a set $T \subseteq V$ is identified as the target alphabet and only elements of T^* which are present in $L(\gamma)$ are accepted by the language. This is called Simple Extended H System(SEH System). A Simple H-System with permitting context has rules of the form (a, b, c) with $a, b, c \in V$. Such a triple represents a splicing rule using the marker a, which is applied to two strings $x, y \in V^*$ only if the symbol b appears in x and c in y.

Similar to permitting context, one can have forbidden context for Simple H systems. A triple (a, b, c) represents a splicing rule using the marker a, which can be applied to two strings $x, y \in V^*$ if and only if b does not appear in x and c does not appear in y.

In this paper we only consider rules of the $(2, 3)$ type into consideration. Formally we define a Simple H-System of (2,3) type with permitting context and target alphabet as a quadruple $\gamma = (V, T, A, R)$ where V is the total alphabet, T is the target alphabet, A is a finite set of axioms and R is a set of splicing rules of the form (a, b, c). For $x, y \in V^*, r = (a, b, c) \in R$

$(x, y) \vdash_r z$ iff $x = x_1 a x_2, y = y_1 a y_2, z = x_1 y_2$ for some $x_1, x_2, y_1, y_2 \in V^*$ and b appears in x and c appears in y.

All languages derivable using this mode of derivation with permitting context and target alphabet belong to the $SEH_{(2,3)}(p)$ family. All languages derivable using the $(2, 3)$ mode of derivation with forbidden context and target alphabet belong to the $SEH_{(2,3)}(f)$ family.

3 The Role of Context in Extended H Systems

In this paper we prove that the power of $SEH_{(2,3)}(p)$ is the same as that of $EH(FIN, p[1])$. There are two features of Simple H-Systems which makes them by definition look like a very special subclass of Extended H Systems. One important feature is that of the structure of the splicing rules in the Simple H-Systems. Another important feature that makes Extended H-systems look very powerful is the presence of permitting contexts of arbitrary sizes. In SEH systems the size of the permitting context is restricted to one. Formally, a rule r in a Extended H System is of the form $(p; C_1, C_2)$ where C_1, C_2 can be arbitrary subsets of the alphabet V but the same rule is valid in Simple H-Systems iff $|C_1| \leq 1, \ |C_2| \leq 1$.

In this section we show that the power of Extended H-Systems is not enhanced by the presence of permitting contexts of arbitrary sizes. For any arbitrary Extended H-System γ, we present an equivalent EH system γ' in which each rule has its permitting context reduced to size one. We also show that for every language L in Extended H-system with forbidden context having a cardinality of context greater than one there is a corresponding marker language $L\$$ in EH systems with cardinality of forbidden context equal to 2.

A splicing system is said to satisfy the property Θ iff every rule $r \in R$ is of the form (p, C_1, C_2) where $p = (u_1\#u_2\$u_3\#u_4)$ is a splicing rule of radius one and $|C_1| = 1, |C_2| = 1$. A splicing system satisfying property Θ has a cardinality of context equal to 1.

Theorem 1: For every splicing system $\gamma = (V, T, A, R) \in EH(FIN, p[1])$ there exists an equivalent splicing system $\gamma' = (V', T, A', R') \in EH(FIN, p[1])$ such that γ' satisfies property Θ and $L(\gamma) = L(\gamma')$.

Proof:

Let $\gamma = (V, T, A, R)$ be a $EH(FIN, p[1])$ system. An equivalent γ', in which each context is of cardinality one, is constructed below. First construct the following sets:

$V_c = \{\gamma_c,\ \delta_c \mid C \subseteq V,\ C \neq \emptyset\}$

$V_j = \{N_a \mid a \in V\}$

$A_c = \{\delta_{c_1}\delta_{c_2},\ \delta_{c_2}\delta_{c_1} \mid C_1 \neq \emptyset,\ C_1,\ C_2 \subseteq V,\ C_2 = C_1 \cup \{a\} \text{ for some } a \in V\}$

$A_a = \{\gamma_c\delta_c,\ \delta_c\gamma_c \mid C \subseteq V,\ C \neq \emptyset\}$

$A_d = \{\gamma_c x \gamma_c \mid x \in A, \text{ all elements in C appear in x}\}$

$A_e = \{D\gamma_c,\ \gamma_c D \mid C \subseteq V,\ C \neq \emptyset\}$

$R_c = \{ (\ \delta_{c_2}\#\delta_{c_1}\$\delta_{c_1}\#,\ \delta_{c_2},\ a), (\ \#\ \delta_{c_1}\$\delta_{c_1}\#\delta_{c_2},\ a,\ \delta_{c_2}) \mid C_2 = C_1 \cup \{a\} \text{ for } a \in V\ \}$

$R_a = \{ (\delta_c\ \#\gamma_c\$\gamma_c\#,\ \delta_c,\ N_a), (\ \#\gamma_c\$\gamma_c\#\ \delta_c, N_a\ ,\ \delta_c) \mid C \subseteq V\ ,\ a \in V\}$

$A_f = \{B\gamma_c,\ \gamma_c B \mid C \subseteq V\}$

$R_f = \{\ (B\#\gamma_c\$\ \gamma_c\#,\ B\ ,\gamma_{c'}\ )\ ,\ (\#\gamma_c\$\gamma_c\#B\ ,\ \gamma_{c'}\ ,\ B) \mid C, C' \subseteq V\ \}$

$A_g = \{B\delta_c,\ \delta_c B \mid C \subseteq V\}$

R_g = { ($\delta_{\{a\}}$#B\$B#, $\delta_{\{a\}}$, a), (#B\$B#$\delta_{\{a\}}$, a, $\delta_{\{a\}}$) | $a \in$V }

R_h = { (D#γ_c\$$\gamma_c$#, D, $\gamma_{c'}$), (# γ_c\$$\gamma_c$#D, $\gamma_{c'}$, D) | $C, C' \subseteq$V }

A_i = {D, γ_cM, Mγ_c}

R_i = {(#D\$D#, M, $\emptyset$), (#D\$D#, $\emptyset$, M), (#γ_c\$$\gamma_c$#, D, $\emptyset$), (# γ_c\$$\gamma_c$#, $\emptyset$, D), $(\#M\$M\#, \emptyset, \emptyset)$ | $C \subseteq$V}

R_s = { (a#N_b\$$N_c$#d , γ_{c_1}, γ_{c_2}) | there exists a rule $(a\#b\$c\#d, C_1$, C_2) in R }

Note that we are considering only the rules of the form (a#b\$c#d, C_1, C_2). There is no loss of generality in this for, any rule in which one or more of 'a', 'b', 'c', or 'd' are missing is equivalent to a set of rules in which each missing position is replaced, successively, by all letters of V.

A_j = { aN_a, N_aa | $a \in$V }

R_j = { (#a\$a#N_a, δ_c, N_a), (N_a#a\$a#, N_a, δ_c) | $C \subseteq$V, a $\in$V}

V' = V $\cup${B , D , M} $\cup V_c \cup V_j$

R' = $R_a \cup R_c \cup R_f \cup R_g \cup R_h \cup R_i \cup R_j \cup R_s$

A' = $A_a \cup A_c \cup A_d \cup A_e \cup A_f \cup A_g \cup A_i \cup A_j$

And $\gamma' = (V', T, A', R')$.

The above splicing system works as follows. The rules and axioms indexed 'i' and 'h' are for removing elements not belonging to V from strings of the form $\gamma_{c_1}\alpha\gamma_{c_2}$to produce the required strings.

The whole mechanism is centered around strings of the form $\gamma_{c_1}\alpha\gamma_{c_2}$.Once such a string is produced rules and axioms in the sets indexed from 'a' to 'g' do the following. First a B is appended to one of the ends of the above string (sets with index 'f' do this). Then the context present in the string α, that is the set of all the distinct letters of V present in α, will be captured in a letter of the form δ_c where C is the set mentioned above. The rules that do this are R_c , R_a.The string α will have letters belonging only to V as will be clear from the description given below. After this the string (which will now be of the form δ_{c_1} $\alpha\gamma_{c_2}$ or $\gamma_{c_2}\alpha\delta_{c_1}$) will be made ready to take part in another splicing .The rules of R_j cut the string at some a $\in$ V and so one end of the string wil be bound by N_a and on the other end the $\delta_{c_1}$is replaced by γ_{c_1}. So the string will now be of the form $\gamma_{c_1}\alpha$'N_a or $N_a\alpha$'γ_{c_1}. This string can now enter splicings with other strings of the same form according to the rules of R_s which simulate the rules of R.

The whole process is shown below. Suppose we start with the strings $\gamma_{c_1}\alpha\gamma_{c_2}$ and $\gamma_{c_3}\beta\gamma_{c_4}$. Then the following splicings will occur

$(B\gamma_{c_1}, \gamma_{c_1}\alpha\gamma_{c_2}) \vdash B\alpha\gamma_{c_2}$ (using a rule from R_f)

$(\delta_{\{a\}}B , B\alpha\gamma_{c_2}) \vdash \delta_{\{a\}}\alpha\gamma_{c_2}$ (using a rule from R_g)

$(\delta_c\delta_{c'} , \delta_{c'}\ \alpha\gamma_{c_2}) \vdash \delta_c\ \alpha\gamma_{c_2}$ (using a rule from R_c)

$(\ \delta_c\alpha'\mathrm{ab}\eta\gamma_{c_2} , \mathrm{bN}_b) \vdash \delta_c\ \alpha'\mathrm{aN}_b$ (using a rule from R_j)

$(\gamma_c\delta_c , \delta_c\ \alpha'\mathrm{N}_a) \vdash \gamma_c\alpha'\mathrm{aN}_b$ (using a rule from R_a)

Similarly $\gamma_{c_3}\beta\gamma_{c_4}$ derives $\mathrm{N}_c\mathrm{d}\beta'\gamma_{c'}$

These two strings will combine according to a rule (a # N_b \$ N_c # d , γ_c, $\gamma_{c'}$) to produce $\gamma_c\alpha'\beta'\gamma_{c'}$
From this string $\alpha'\beta'$ will be produced as follows.

$(D\gamma_c , \gamma_c\alpha'\beta'\gamma_{c'}) \vdash D\alpha'\beta'\gamma_{c'}$ (using a rule from R_h)
$(D\alpha'\beta'\gamma_{c'} , \gamma_{c'}M) \vdash D\alpha'\beta'\ M$ (using a rule from R_i)
$(D ,D\alpha'\beta'\ M) \vdash \alpha'\beta'\ M$ (using a rule from R_i)
$(\ \alpha'\beta'\ M , M) \vdash \alpha'\beta'$ (using a rule from R_i)

We now prove that γ is equivalent to γ'.

Claim 1: If γ' produces a string x then it must be produced by γ.
Proof : For strings derived in four steps in γ' the claim is true from the the fact that $\gamma_c x\gamma_c$ is an axiom of γ' if x is an axiom of γ and only such strings can be produced in three steps(three is the minimum number of steps required to produce a string from V* in γ'). Assume that the assertion is true for strings which are derived in less than k steps where k is the number of steps taken to derive x in γ'. Now x must have been produced from a string of the form $\gamma_{c_1}x\gamma_{c_2}$.This in turn must have been produced from strings of the form $\gamma_{c_1}u'aN_b$ and $N_c dv'\gamma_{c_2}$. These must have been produced from strings $\gamma_{c_1}u\gamma_{c_3}$ and $\gamma_{c_4}v\gamma_{c_2}$for some C_3 and C_4 as can be seen from the example shown above.By induction hypothesis u and v will be produced in

γ (note that from the construction of the system u and v will be produced in γ' and they will be produced in less than k steps.The number of steps taken to capture the context and prepare the string for splicing take at least as many steps as it takes to remove the symbols γ_c, which is four). Also since the strings $\gamma_{c_1} u$aN_b and $\mathrm{N}_c dv\gamma_{c_2}$ are produced from $\gamma_{c_1} u\gamma_{c_3}$ and $\gamma_{c_4} v\gamma_{c_2}$respectively and the former pair of strings combine according to the rule (a # N_b \$ N_c # d ,γ_{c_1},γ_{c_2}) where $\mathrm{C}_1$and C_2 capture the context in u and v respectively ,there must be rule (a # b \$ c # d, C_1, C_2) in R according to which u and v can combine to produce u'v'.Hence the proof.

Claim 2: If a string x is produced in γ then it will be produced in γ'

Proof: We can again use the same approach. The assertion is true for strings derived in one step in γ (the axioms) by the construction of γ' . Suppose the assertion is true for strings which are derived in less than k steps. Let $x = uadv$ be derived from strings of the form $u$abβ and δcdv by the rule (a # b \$ c # , C, C'). By induction hypothesis $u$abβ and δcdv must have been produced in γ'. So by construction of S the strings $\gamma_c u$aN_b and $\mathrm{N}_c dv\gamma_{c'}$ will also be produced in γ'.And the rule (a # N_b \$ N_c # d ,γ_c,$\gamma_{c'}$) will be in R' .So the string $\gamma_c uv\gamma_{c'}$ will be produced. From this string uv will be produced by the rules from R_i and R_h. Hence the proof.

So from the above two claims it can be seen that the γ' produces those strings and only those strings that are produced by γ. Hence cardinality of context does not affect the power of the Extended H system with permiting contexts and rules of radius one.

Therefore $L(\gamma) = L(\gamma')$.□

Note that the same proof is applicable to radius of arbitrary sizes. One can apply the same proof to show that $EH(FIN, p[k]) = EH(FIN, p[k, 1])$. This clearly indicate that the cardinality of context adds no power to Extended H-systems with permitting context.

Theorem 2: For every language $L \in EH(FIN, f[k])$ there exists a language $L' \in EH(FIN, f[k, 2])$ such that $L' = L£$ for a character £ not present in the alphabet of L.

Proof: Consider a splicing system $\gamma = (V, T, A, R)$ in EH(FIN,f[k]) for some n. For every rule $r \in R$ we introduce two new symbols $\beta_r, \beta_{r'}$. Let

$R = \{r_1, r_2, \ldots r_n\}$.

Let $Y = \beta_{r_1}\beta_{r_2}\ldots\beta_{r_n}$ and let $Y_{r_i} = \beta_{r_1}\beta_{r_2}\ldots\beta_{r_{i-1}}\beta_{r_{i+1}}\ldots\beta_{r_n}$.

Let $Y' = \beta_{r'_1}\beta_{r'_2}\ldots\beta_{r'_n}$ and let $Y'_{r_i} = \beta_{r'_1}\beta_{r'_2}\ldots\beta_{r'_{i-1}}\beta_{r'_{i+1}}\ldots\beta_{r'_n}$.

Let $V_0 = V \cup \{p\}$ for some $p \notin V$. Transform every rule $r = (q, C, D) \in R$ where $C = \phi$ or $D = \phi$ to rules of the form $r = (q, C', D')$ where $C' = \{p\}$ if $C = \phi$ else it is equal to C and $D' = \{p\}$ if $D = \phi$ else it is equal to D.

This transformation has no effect on $L(\gamma)$ since we have introduced a forbidden context on a character not present in V. The proof presented below requires that every rule has non-empty forbidden context. The above transformation does not increase the cardinality of context of γ.

Construct $\gamma' = (V', T', A', R') \in EH(FIN, f[k, 2])$ as follows:

$T' = T \cup \{£\}$

$V_c = \{\gamma_C | C \neq \phi, C \subseteq V_0\}$

$V_r = \{\beta_r, \beta_{r'} | r \in R\}$

$V' = V_0 \cup V_c \cup V_r \cup \{Z, X, X', Z', Z'', X'', £\}$

$A_s = \{Y'ZwXY, Y'XwZY | w \in A\} \cup \{ZZ'', Z''£, Z'Y, Z'ZY, Y'ZX', Y'X', X''\}$

$A_c = \{\gamma_X\gamma_W Y, Y'\gamma_W\gamma_X | X, W \subseteq V, X \neq \phi, W = X \cup \{a\} \text{ for some } a \in V\}$

$A_e = \{\gamma_C Y_{r_i}, Y'_{r_i}\gamma_D | r_i = (p, C, D) \in R\}$

$A_d = \{Z\gamma_C Y, Y'\gamma_C Z || C| = 1, C \subseteq V_0\}$

$A' = A_s \cup A_c \cup A_d \cup A_e$

$R_c = \{(\#\gamma_X\$\gamma_X\#, a, Z), (\#\gamma_X\$\gamma_X\#, Z, a) | X \subseteq V - \{a\}\}$

$R_d = \{(Z\#\$Z\#\gamma_C, a, X), (\gamma_C\#Z\$\#Z, X, a) | a \in V, C = \{a\}\}$

$R_r = \{(p, \{\beta_{r_i}, £\}, \{\beta_{r'_i}, £\}) | r_i = (p, C, D) \in R\}$

$R_e = \{(Z\#\gamma_C\$\gamma_C\#, \phi, \{\beta_r, X\}), (\#\gamma_D\$\gamma_D\#Z, \{\beta_{r'}, X\}, \phi) | r = (p, C, D) \in R\}$

$R_f = \{(\#X\$\#Z', \beta_{r'}, \beta_{r'}), (X'\#\$X\#, \beta_r, \beta_r) | r \in R\}$

$R_p = \{(\#Z\$Z\#Z'', \phi, \phi), (\#Z''\$Z''\#£, \{X, Z\}, \phi), (X\#\$\#X'', \{£, Z\}, \phi)\}$

$R_g = \{(\#Z'\$Z'\#Z, \{Z, X'\}, X), (Z\#X'\$X'\#, X, \{Z, Z'\}) | r \in R\}$

$R' = R_c \cup R_d \cup R_e \cup R_f \cup R_r \cup R_p \cup R_g$

The cardinality of context of γ' is 2 and the radius of γ' is equal to the radius of γ. Therefore one can clearly see that γ' belongs to $EH(FIN, f[k, 2])$. We will now prove that the language generated by γ' is in fact equal to the language generated by γ appended with a constant letter $£$.

A string x is said to satisfy the forbidden context $C \subseteq V_0$ if all the characters of C are not present in x.

The splicing system γ' satisfies the following property:
Any string $w \in V^*$ derivable in any intermediary step of γ' and of the form $Y'XyZ\gamma_C Y$ or $Y'\gamma_C ZyXY$ is such that y is an intermediary string derivable in γ and y satisfies the forbidden context C.

The rules in R_r are used to simulate the rules R of γ. The rules in R_c, R_d are used to generate all the possible forbidden contexts for a single string $x \in V^*$. Every string x is initially appended with the strings ZY and XY' in order to generate all the possible contexts for the string. For a given string $Y'XxZY$, γ' produces all strings of the form $Y'XxZ\gamma_C Y$ where x satisfies the forbidden context C. Similarly, for a given string $Y'ZxXY$, γ' produces all strings of the form $Y'\gamma_C ZxXY$ where x satisfies the forbidden context C.

Suppose $C \subseteq V$ and x satisfies the forbidden context C, then we produce the string $Y'XxZ\gamma_C Y$ from $Y'XxZY$ as follows:
Assume $Y'XxZY$ is derivable in γ'
$(Y'XxZY, Z\gamma_{a_i}Y) \vdash Y'XxZ\gamma_{a_i}Y$ for some $a_i \in C$ using the corresponding rule in R_d.
Let us prove by induction on the size of the context C that $Y'XxZ\gamma_C Y$ is derivable.
Let C' be a subset of C such that $C' = C - \{a_j\}$ for some j
By induction hypothesis $Y'XxZ\gamma_{C'}Y$ is derivable.
$(Y'XxZ\gamma_{C'}Y, \gamma_{C'}\gamma_C Y) \vdash Y'XxZ\gamma_C Y$ using the corresponding rule in R_c

Therefore $Y'XxZ\gamma_C Y$ is derivable iff x satisfies the forbidden context C. Similarly we can show that $Y'\gamma_C ZxXY$ is also derivable in γ'.

So for every string $Y'XwZY$ in γ' all strings of the $Y'XwZ\gamma_C Y$ and $Y'\gamma_C ZwXY$ are derivable iff w satisfies the forbidden context C.

γ' satisfies another important property:
Every string $u \in V'^*$ derivable in γ' and which does not contain the character β_r for some $r = (p, C, D) \in R$ satisfies the forbidden context C and every string that does not contain the character $\beta_{r'}$ satisfies the forbidden context D.

Note that the only strings $w£ \in V^*£$ derivable in γ' are derivable only from $Y'XwZY$. For every $w \in V^*$ derivable in γ, $Y'XwZY, Y'ZwXY$ are derivable in γ'.

We will show the above result using induction on the number of splicing steps required to produce a string $w \in V^*$ in γ.

Since $\{Y'XxZY, Y'ZxXY | x \in A\} \subset A'$, the basis step of induction is true.

Assume that for all strings x derivable in at most k splicing steps in γ, $Y'XxZY, Y'ZxXY$ are derivable in γ'. Let $w \in V^*$ be derivable in γ in $k+1$ steps. Let w be derived from strings $u, v \in V^*$ using rule $r_i \in R$. By induction hypothesis $Y'XuZY, Y'ZvXY$ are derivable in γ'. Let $r_i = (p, C_1, C_2) \in R$. Since $(u, v) \vdash_{r_i} w$, u satisfies the forbidden context C_1 and v satisfies the forbidden context C_2. Therefore the strings $Y'XuZ\gamma_{C_1}Y$ and $Y'\gamma_{C_2}ZvXY$ are also derivable. Using rules of R_e we can also derive $Y'_{r_i}ZvXY$ and $Y'XuZY_{r_i}$. Using these strings and the rules of R_f one can derive $Y'_{r_i}ZvXY, Y'_{r_i}ZvZ'Y, Y'XuZY_{r_i}$ and $Y'X'uZY_{r_i}$.

Using the rules of R' one can derive $Y'XwZY, Y'ZwXY$ in the following way:
Using the rule $(p, \{\beta_{r_i}, £\}, \{\beta'_{r_i}, £\})$ in R_r the strings $Y'XuZY_{r_i}, Y'X'uZY_{r_i}$ can splice with the strings $Y'_{r_i}ZvXY, Y'_{r_i}ZvZ'Y$ to produce the strings $Y'XwXY$, $Y'XwZ'Y, Y'X'wXY$ and $Y'X'wZ'Y$. Among the four strings produced the strings $Y'XwXY$ and $Y'X'wZ'Y$ are rendered inactive since they cannot splice anymore. Using the rules of R_g the string $Y'XwZY$ and $Y'ZwXY$ are derivable from the strings $Y'XwZ'Y$ and $Y'X'wXY$ respectively.

Therefore one can obtain $Y'XwZY$ and $Y'ZwXY$ in γ'. From $Y'XwZY$ we obtain $w£$ using the splicing rules listed below.
$(Y'XwZY, ZZ'') \vdash Y'XwZ''$ using rule $(\#Z\$Z\#Z'', \phi, \phi)$
$(Y'XwZ'', X'') \vdash wZ''$ using rule $(X\#\$\#X'', \{Z, £\}, \phi)$
$(wZ'', Z''£) \vdash w£$ using rule $(\#Z''\$Z''\#£, \{X, Z\}, \phi)$

From this we can note that for every string w derivable in γ, $w£$ is derivable in γ'.

Now we will prove that for every string $w£ \in V^*£$ derivable in γ', w is precisely derivable in γ. In γ' the strings $w£ \in V^*£$ are derivable only from strings of the form $Y'XwZY$.

Let us prove the above step using induction on the number of splicing steps required to produce a string $w£$ in γ'. If $Y'XwZY \in A'$ then $w \in A$ and $w£$ is derivable in γ'. Therefore the basis step of induction is true. A string of the form $Y'XwZY$ can be derived in γ' only from a string of the form $Y'XwZ'Y$ which in turn can be produced only from two strings of the form $Y'XuZY_r, Y_rZvXY$ for some $r = (p, C, D) \in R$. The strings $Y'XuZY_r, Y'_rZvXY$ are derivable only from $Y'XuZ\gamma_CY$ and $Y'\gamma_DZvXY$. From this one can infer that the strings $Y'XuZY$ and $Y'XvZY$ are derivable in γ' and that u, v satisfy the forbidden context C, D respectively.

By induction hypothesis we get that u, v are derivable in γ. In γ one can have the following splicing action:
$(u, v) \vdash_r w$
Therefore w is derivable in γ.

By induction one can conclude that for every $w\pounds \in V^*\pounds$ derivable in γ', w is precisely derivable in γ.

Therefore $L(\gamma') = L(\gamma)\pounds$.

From the above two theorems one can infer that the cardinality of permitting context does not add power to Extended H-Systems but the cardinality of forbidden context seems to play an important role in Extended H-Systems.

4 Equivalence of $SEH_{2,3}(p)$ and $EH(FIN, p[1])$

In the previous section, we proved that the power of Extended H-Systems is not enhanced by the presence of permitting contexts of arbitrary sizes. In this section we derive the equivalence of $SEH_{2,3}(p)$ with Extended H-Systems with cardinality of context restricted to one. For any arbitrary Extended H-System γ in which each rule has its permitting context reduced to size one, we present an equivalent SEH system γ' with rules of type $(2, 3)$ having permitting context and terminal alphabet.

4.1 Notations

Let $\gamma = (V, T, A, R)$ be an extended H system of radius 1 with permitting context having a cardinality of context equal to 1. We introduce new symbols of the form $X_{a,b}$ for all $a, b \in V \cup \{\epsilon\}$ with the exception of $X_{\epsilon,\epsilon}$.

Let $V_e = \{X_{a,b} | a, b \in V \cup \{\epsilon\}\} - \{X_{\epsilon,\epsilon}\}$
$V_0 = V \cup V_e$
A string $w \in V_0^*$ is said to be valid iff

1. Two symbols of V_e do not occur adjacent to each other.

2. if $X_{a,b}$ is present in w then the left adjacent symbol of $X_{a,b}$ has to be a and b its right adjacent symbol.

3. The first character c of w must be either an element of V or should be of the form $X_{\epsilon,a}$ for some $a \in V$.

4. The last character d of w must be either an element of V or should be of the form $X_{a,\epsilon}$ for some $a \in V$.

The boolean function *valid* assumes the value *true* for a string w if it is valid, else it takes the value *false*.

Define a function $g : V_0^* \to V^*$. For every $u \in V_0^*$, $g(u)$ is obtained by substituting ϵ for all characters of V_e present in u. From the definition one can infer that $g(w) = w$ iff $w \in V^*$.

Define a function $f : V^* \to P(V_0^*)$. The function f is defined as the valid preimage of a word $w \in V^*$ under the function g. Note that $P(X)$ denotes the power set of the set X.

Mathematically, we obtain

$f(w) = \{u | u \in V_0^*, valid(u), g(u) = w\}$

The function f can be extended to all languages $L \subseteq V^*$:

$f(L) = \cup_{w \in L} f(w)$

It is not difficult to see that

Lemma 1: For every finite language L, $f(L)$ is finite.

A splicing system is said to satisfy property α if and only if the following conditions are satisfied:

For every rule $r = (a\#b\$c\#d, C_1, C_2)$ where some of the alphabets a, b, c, d are ϵ, there exists rules of the form $(e\#f\$g\#h, C_1, C_2)$ such that the symbols corresponding to the $\epsilon-$ alphabets in rule r assume all possible characters in $V \cup \{\epsilon\}$.

It is straightforward to note that

Lemma 2: Given a splicing system $\gamma = (V, T, A, R)$ one can transform γ to $\gamma' = (V, T, A, R')$ such that γ' satisfies property α and $L(\gamma) = L(\gamma')$.

Theorem 3: $SEH_{2,3}(p) = EH(FIN, p[1])$.

Proof: Consider an extended H-system $\gamma = (V, T, A, R)$ that satisfies properties α and Θ. We will form a simple H system γ' of the $(2, 3)$ type with permitting context and target alphabet which generates $L(\gamma)$.

Let V_e, V_0, f and g be as defined earlier. $\gamma' = (V', T, A', R')$ where :
$V' = V_0 \cup \{\gamma_r | r \in R\} \cup \{M\}$
$A' = f(A) \cup \{MX_{a,b}\gamma_r M', M'\gamma_r X_{a,d}X_{c,d}M, M'\gamma_r X_{c,d}M | r = (a\#b\$c\#d, C_1, C_2) \in R\}$
$R' = \{(X_{a,b}, C_1, \{\gamma_r\}), (X_{c,d}, \{\gamma_r\}, C_2), (\gamma_r, \{a\}, \{d\}) | r = (a\#b\$c\#d, C_1, C_2) \in R\}$

Since A is a finite language over V we can directly infer that $f(A)$ is also finite.

A string $w \in V^*$ is said to be $\gamma-$ derivable, if w can be derived from the set of rules and axioms in a sequence of splicing steps. Note that w can be any intermediary string derived in γ and need not be present in T^*. We extend the same definition to γ' over the set V_0.

We will show that for every w that is $\gamma-$ derivable, all strings of $f(w)$ are derivable in γ'. We will also show that for every string $v \in V_0^*$ derivable in γ', $g(v)$ is derivable in γ. We will prove this assertion using induction.

The induction will be on the number of splicing steps required to produce a string $w \in V^*$. Since $f(A) \subset A'$, for all strings $w \in V^*$ which are $\gamma-$ derivable in zero steps, $f(w)$ is γ' derivable.

Assume that for all strings $w \in V^*$ which are $\gamma-$derivable in atmost k steps, $f(w)$ is $\gamma'-$ derivable. Consider a string $w \in V^*$ which is derived in $k+1$ splicing steps. Let $r \in R$ be the final rule applied to obtain w from strings u, v.

If $r = (a\#b\$c\#d, C_1, C_2)$ then $u = u_1abu_2, v = v_1cdv_2$ and $w = u_1adv_2$ for some strings $u_1, u_2, v_1, v_2 \in V^*$.

Let $P(w)$ denote those sets of strings in $f(w)$ which do not end in a symbol of the form $X_{a,\epsilon}$ and $Q(w)$ denote those set of strings in $f(w)$ that do not start with a symbol of the form $X_{\epsilon,b}$ for some $a, b \in V$.

By induction hypothesis since u, v are $\gamma-$ derivable in atmost k splicing steps in γ, all strings in $f(u)$ and $f(v)$ are $\gamma'-$derivable.

If $s = s_1abs_2 \in V^*$ then :
$f(s) = \{w_1w_2, w_1X_{a,b}w_2 | w_1 \in P(s_1a), w_2 \in Q(bs_2), a, b \in V\}$

Therefore any string of $f(w)$ is of the form w_1w_2 or $w_1X_{a,d}w_2$ where $w_1 \in P(u_1a)$ and $w_2 \in Q(dv_2)$. Consider an arbitrary $w_1 \in P(u_1a)$ and an arbitrary $w_2 \in Q(dv_2)$. We will show that both w_1w_2 and $w_1X_{a,d}w_2$ are derivable in γ' for this arbitrary choice of w_1 and w_2. Since $f(u)$ and

$f(v)$ are $\gamma'-$derivable there exists two strings $u' \in f(u), v' \in f(v)$ such that $u' = w_1 X_{a,b} u'_2$ and $v' = v'_1 X_{c,d} w_2$ for some $u'_2 \in Q(bu_2)$ and $v'_1 \in P(cv_1)$.

Now w is derived in γ from u, v using rule $r \in R$.

To derive $w_1 w_2$ and $w_1 X_{a,d} w_2$ in γ' we splice in the following way:
$(w_1 X_{a,b} u'_2, M X_{a,b} \gamma_r M') \vdash w_1 \gamma_r M'$ using $(X_{a,b}, C_1, \{\gamma_r\})$
$(M' \gamma_r X_{a,d} X_{c,d} M, v'_1 X_{c,d} w_2) \vdash M' \gamma_r X_{a,d} w_2$ using $(X_{c,d}, \{\gamma_r M'\}, C_2)$
$(M' \gamma_r X_{c,d} M, v'_1 X_{c,d} w_2) \vdash M' \gamma_r w_2$ using $(X_{c,d}, \{\gamma_r M'\}, C_2)$
$(w_1 \gamma_r M', M' \gamma_r X_{a,d} w_2) \vdash w_1 X_{a,d} w_2$ using $(\gamma_r, \{a\}, \{d\})$
$(w_1 \gamma_r M', M' \gamma_r w_2) \vdash w_1 w_2$ using $(\gamma_r, \{a\}, \{d\})$

Therefore $w_1 w_2$ and $w_1 X_{a,d} w_2$ are $\gamma'-$derivable for every $w_1 \in P(u_1 a), w_2 \in Q(bv_2)$. Therefore $f(w)$ is $\gamma'-$ derivable for every w that is $\gamma-$ derivable.

The splicing system γ' satisfies the following property:
Every string $w = w_1 w_2 \in V_0^*$ is derived from two strings of the form $w_1 \gamma_r$ and $\gamma_r w_2$ where $w_1, w_2 \in V_0^*$.

We will prove by induction that for every $w \in V_0^*$ that is derivable in γ', $g(w)$ is $\gamma-$ derivable. We again apply induction on the number of splicing steps needed to derive w.

Note that $A' \cap V_0^* = f(A)$ and $g(f(A)) = A$. Thereby for all strings $w \in V_0^*$ which are derivable in zero steps, $g(w)$ is γ derivable.

Assume that for all strings $w \in V_0^*$ which are $\gamma'-$derivable in atmost k steps, $g(w)$ is $\gamma-$ derivable. Consider a string $w \in V_0^*$ derived in $k+1$ steps. $w = w_1 w_2$ or $w_1 X_{a,d} w_2$ derived from strings of the form $u_1 = w_1 \gamma_r, u_2 = \gamma_r w_2, u_3 = \gamma_r X_{a,d} w_2$ where $w_1, w_2 \in V_0^*$ and $r = (a\#b\$c\#d, C_1, C_2) \in R$. Note that u_1, u_2, u_3 are derived from v_1, v_2 where $v_1 = w_1 X_{a,b} v'_1$ and $v_2 = v'_2 X_{c,d} w_2$ for $v'_1, v'_2 \in V_0^*$.
$(w_1 X_{a,b} v'_1, M X_{a,b} \gamma_r) \vdash w_1 \gamma_r$
$(\gamma_r X_{c,d} M, v'_2 X_{c,d} w_2) \vdash \gamma_r w_2$
$(\gamma_r X_{a,d} X_{c,d} M, v'_2 X_{c,d} w_2) \vdash \gamma_r X_{a,d} w_2$

v_1 and v_2 are $\gamma'-$derivable in atmost k steps. By induction hypothesis we get that $g(v_1)$ and $g(v_2)$ are $\gamma-$derivable.

Let $g(v_1) = s_1$ and $g(v_2) = s_2$.
$s_1 = g(w_1) g(v'_1), s_2 = g(v'_2) g(w_2)$
Since v_1 is a valid string, w_1 must end with a and v'_1 must start with b. Therefore the site ab is present in $g(v_1)$. Similarly the site cd must be present in $g(v_2)$.

Since $w_1\gamma_r$ is derivable from v_1, v_1 satisfies the permitting context C_1. Similarly we can prove that v_2 satisfies context C_2. Since $C_1, C_2 \subset V$ we have that $g(v_1), g(v_2)$ satisfy contexts C_1, C_2 respectively.

$s_1 = x_1abx_2, s_2 = y_1cdy_2$
$r = (a\#b\$c\#d, C_1, C_2)$ and s_1 satisfies C_1 ands_2 satisfies C_2
$(s_1, s_2) \vdash_r s$ where $s = x_1ady_2$

Since s_1, s_2 are $\gamma-$ derivable s is $\gamma-$ derivable. $w = w_1w_2$ or $w_1X_{a,d}w_2 \Rightarrow g(w) = g(w_1)g(w_2) = s$

Therefore $g(w)$ is $\gamma-$derivable.

By induction we thereby infer that for all $w \in V_0^*$ which is $\gamma-'$ derivable $g(w)$ is $\gamma-$derivable.

We have shown that for every w that is $\gamma-$derivable, $f(w)$ is $\gamma-'$ derivable. Since $f(w) \cap V^* = \{w\}$, $wis\gamma'-$ derivable. (1) Similarly for every $w \in V_0^*$ that is $\gamma'-$derivable, $g(w)$ is $\gamma-$derivable. For every $w \in V^*g(w) = w$.(2)

From (1) we infer that all the strings that are derived in γ are derivable in γ'. From (2) we infer that the only strings of V^* that are derivable in γ' are precisely the strings that are derivable in γ.

Therefore one can infer that the set of terminal strings derived by both these languages are the same.

Therefore $L(\gamma) = L(\gamma')$.

The power of $EH(FIN, p[1])$ is not reduced by adding properties α and Θ to the splicing system. This can be seen from the lemmas proved before.

For any arbitrary $\gamma \in EH(FIN, p[1])$ we can generate a language $\gamma' \in SEH_{2,3}(p)$ such that the languages generated are the same.

Therefore $EH(FIN, p[1]) \subseteq SEH_{2,3}(p)$.

By definition all splicing systems $\gamma \in SEH_{2,3}(p)$ belong to $EH(FIN, p[1])$.

$SEH_{2,3}(p) \subseteq EH(FIN, p[1])$.

Hence it follows that $EH(FIN, p[1]) = SEH_{2,3}(p)$.

5 Simple Extended H-Systems of (2,3) Type with Forbidden Contexts

In this section we prove an interesting result on Simple Extended H-Systems of the (2,3) type with forbidden context and terminal alphabet. As defined earlier, we will refer to the languages in this class as $SEH_{2,3}(f)$. We will show that the two classes of languages $SEH_{2,3}(f)$ and $EH(FIN, f[1,1])$ are equal.

5.1 Notations

We introduce two new symbols of the form $X_{a,b}, X'_{a,b}$ for all $a, b \in V \cup \{\epsilon\}$ with the exception of $X_{\epsilon,\epsilon}, X'_{\epsilon,\epsilon}$.

Let $V_e = \{X_{a,b}, X'_{a,b} | a, b \in V \cup \{\epsilon\}\} - \{X_{\epsilon,\epsilon}, X'_{\epsilon,\epsilon}\}$

$V_0 = V \cup V_e$

A string $w \in V_0^*$ is said to be valid iff

1. Two symbols of V_e do not occur adjacent to each other.
2. If $X_{a,b}$ or $X'a, b$ is present in w then the site where $X_{a,b}$ or $X'_{a,b}$ occurs in w should be of the form $aX_{a,b}X'_{a,b}b$.
3. The leftmost substring c of w must either begin with an element of V or should be of the form $X_{\epsilon,a}X'_{\epsilon,a}a$ for some $a \in V$.
4. The rightmost substring d of w must be either an element of V or should be of the form $aX_{a,\epsilon}X'_{a,\epsilon}$ for some $a \in V$.

The boolean function *valid* assumes the value *true* for a string w if it is valid, else it takes the value *false*.

We define two functions f, g in a similar fashion to the one defined in the earlier section.

Lemma 3: For every finite language L, $f(L)$ is finite.

Proof: The proof is similar to the proof for *Lemma 1*.

Lemma 4: Every splicing system $\gamma \in EH(FIN, f[1,1])$ can be transformed to an equivalent splicing system $\gamma' \in EH(FIN, f[1,1])$ satisfying property Θ.

Proof: Let $\gamma = (V, T, A, R) \in EH(FIN, f[1,1])$. Construct a splicing system $\gamma' = (V', T, A, R')$ as follows:
Let p be an alphabet not in V. $V' = V \cup \{p\}$
Let $\psi : V \cup \{\epsilon\} \rightarrow V'$ such that $\psi(v) = v$ if $v \in V$ and $\psi(\epsilon) = p$.
$R' = \{(q, \psi(C), \psi(D) | r = (q, C, D) \in R\}$

Clearly γ' satisfies property Θ and the presence of the alphabet p in the forbidden context of a rule does not change the set of derivable strings in the splicing system γ'.

Note that *Lemma* 2 is independent of the type of context i.e forbidden or permitting. Therefore for a given splicing system in $EH(FIN, f[1,1])$ one can construct an equivalent splicing system in the same class satisfying properties α and Θ.

Theorem 4**:** $SEH_{2,3}(f) = EH(FIN, f[1,1])$.

Proof: Consider a splicing system $\gamma = (V, T, A, R)$ in the extended H system that satisfies properties α and Θ. We will form a simple H system γ' with forbidden context and target alphabet which generates $L(\gamma)$.

Let $V_e = \{X_{a,b}, X'_{a,b} | a, b \in V \cup \{\epsilon\}\} - \{X_{\epsilon,\epsilon}, X'_{\epsilon,\epsilon}\}$

Enumerate the rules of the set R as $r_1, r_2, \ldots r_n$. Introduce n new symbols $\beta_1, \beta_2, \ldots \beta_n$ corresponding to each rule in R. Let $V_r = \{\beta_1, \beta_2, \ldots \beta_n\}$ and let Y denote the string $\beta_1\beta_2 \ldots \beta_n$. Let Y_{r_i} denote the string $\beta_1 \ldots \beta_{r_{i-1}}\beta_{r_{i+1}} \ldots \beta_{r_n}$.
$V'_r = \{\beta'_r, \gamma_r | r \in R\}$
$V_0 = V \cup V_e$

Let f, g be the same functions as defined before.

$\gamma' = (V', T, A', R')$ where :
$V' = V_0 \cup V_r \cup V'_r \cup \{M\}$
$A_a = \{M\beta_r X_{a,b}\beta'_r Y_r, Y_r\beta'_r X'_{c,d}\beta_r M, Y_r\beta'_r X_{a,d}X'_{a,d}X'_{c,d}\beta_r M$
$| r = (a\#b\$c\#d, C_1, C_2) \in R, a \neq c\}$
$A_b = \{\beta_r M X_{a,b}X_{a,d}\beta'_r, \beta'_r X'_{a,d}X'_{a,d}M\beta_r, MX_{a,b}\gamma_r, \gamma_r X'_{a,d}M$
$| r = (a\#b\$a\#d, C_1, C_2) \in R\}$
$A' = f(A) \cup A_a \cup A_b$
$R_a = \{(X_{a,b}, C_1, X'_{a,b}), (X'_{c,d}, X_{c,d}, C_2), (\beta'_r, \beta_r, \beta_r)$
$| r = (a\#b\$c\#d, C_1, C_2) \in R$ and $a \neq c\}$
$R_b = \{(X_{a,b}, C_1, X'_{a,b}), (X'_{a,d}, X_{a,d}, C_2), (\beta'_r, \beta_r, \beta_r), (\gamma_r, M, M)$
$| r = (a\#b\$a\#d, C_1, C_2) \in R\}$

$R' = R_a \cup R_b$

Since A is a finite language over V we can directly infer that $f(A)$ is also finite.

Now we will show how γ' simulates a particular rule $r \in R$ of γ. Let $r = (a\#b\$c\#d, e, f)$ and let two strings u, v splice using rule r and produce w.

We can infer that $u = x_1abx_2$ and satisfies the forbidden context e and $v = y_1cdy_2$ and satisfies the forbidden context f.

Since u, v are derivable in γ and by induction on the number of splicing steps required to produce a string in γ, we have that $f(u)$ and $f(v)$ to be derivable in γ'.

Consider two strings $u' \in f(u)$ and $v' \in f(v)$ such that $u' = z_1aX_{a,b}X'_{a,b}bz_2$ and $v' = w_1cX_{c,d}X'_{c,d}dw_2$.

Case 1: $(a \neq c)$

$(u', M\beta_rX_{a,b}\beta'_rY_r) \vdash z_1a\beta'_rY_r$ using the rule $(X_{a,b}, e, X'_{a,b})$
$(Y_r\beta'_rX'_{c,d}\beta_rM, v') \vdash Y_r\beta'_rdw_2$ using the rule $(X'_{c,d}, X_{c,d}, f)$
$(Y_r\beta'_rX_{a,d}X'_{a,d}X'_{c,d}\beta_rM, v') \vdash Y_rX_{a,d}X'_{a,d}\beta'_rdw_2$ using the rule $(X'_{c,d}, X_{c,d}, f)$
$(z_1a\beta'_rY_r, Y_r\beta'_rdw_2) \vdash z_1adw_2$ using the rule $(\beta'_r, \beta_r, \beta_r)$
$(z_1a\beta'_rY_r, Y_rX_{a,d}X'_{a,d}\beta'_rdw_2) \vdash z_1aX_{a,d}X'_{a,d}dw_2$ using the rule $(\beta'_r, \beta_r, \beta_r)$

Case 2: $a = c$

$(u', \beta_rMX_{a,b}X_{a,d}\beta'_r) \vdash z_1aX_{a,d}\beta'_r$ using the rule $(X_{a,b}, e, X'_{a,b})$
$(u', MX_{a,b}\gamma_r) \vdash z_1a\gamma_r$ using the rule $(X_{a,b}, e, X'_{a,b})$
$(\beta'_rX'_{a,d}X'_{a,d}M\beta_r, v') \vdash \beta'_rX'_{a,d}dw_2$ using the rule $(X'_{c,d}, X_{c,d}, f)$
$(\gamma_rX'_{a,d}M, v') \vdash \gamma_rdw_2$ using the rule $(X'_{c,d}, X_{c,d}, f)$
$(z_1aX_{a,d}\beta'_r, \beta'_rX'_{a,d}dw_2) \vdash z_1aX_{a,d}X'_{a,d}dw_2$ using the rule $(\beta'_r, \beta_r, \beta_r)$
$(z_1a\gamma_r, \gamma_rdw_2) \vdash z_1adw_2$ using the rule (γ_r, M, M)

In both cases we can produce the strings z_1adw_2 and $z_1aX_{a,d}X'_{a,d}dw_2$ which belong to $f(w)$.

In a similar fashion all the elements of $f(w)$ can be obtained by choosing the corresponding elements u', v' from $f(u), f(v)$.

Therefore w is the only element of V^* obtainable in γ' using the simulation of the rule $r \in R$, since $w \in f(w)$ and $f(w) \cap V^* = \{w\}$.

The rest of the proof is very similar to the proof method for Theorem 3.

6 Conclusion

In this paper we have proved that the power of simple H systems of the $(2, 3)$ type is equivalent to that of Extended H systems with splicing rules of radius one. First, we prove that multiple context does not add to the power of extended H-systems. We then provide a construction of a Simple H-system which generates the same language. This result is an interesting one since this class by definition appears as a small subclass of $EH(FIN, p[1])$. This paper has initiated work in the direction of providing forbidden context for simple H-systems. We have also proved that $SEH_{2,3}(f)$ is equal to $EH(FIN, f[1, 1])$.

In [8] it is conjectured that $EH(FIN, p[1]) = EH(FIN, f[1]) = CF$. In [3] it has been proved that $CF \subseteq EH(FIN, p[1])$ and $CF \subseteq EH(FIN, f[1])$. In [2] it has been proved that $CF \subseteq SEH_{2,3}(p)$. If the conjecture in [8] is proved positively, then all these classes will become equal to CF.

There are several directions worth pursuing. The role of cardinality of forbidden context in Extended H-Systems is an interesting open problem. The power of simple H-systems of $(1, 4)$ and $(1, 3)$ types with forbidden context is also an exciting area to attack and is open for research.

References

[1] G.Alford, An explicit construction of a universal extended H system, *Workshop on Molecular Computing*, Mangalia,1997.

[2] V.T. Chakaravarthy and K.Krithivasan, Some results on simple extended H-systems, *Romanian Journal of Information Science and Technology*, Vol. 1, Number 3, 1998, 203-215.

[3] V.T. Chakaravarthy and K.Krithivasan, A note on Extended H systems with permitting/ forbidden context of radius one, *Bulletin of EATCS*, 62, 1997, 208-213.

[4] T.Head, Formal Language theory and DNA: an analysis of the generative capacity of specific recombinant behaviours, *Bulletin of Math. Biology*, 49(1987), 737-759.

[5] T.Head, Gh. Paun, D.Pixton , Language theory and molecular genetics , chapter 7 in vol.2 of [9], 295-360.

[6] A.Mateescu, Gh.Paun, G.Rozenberg, A.Salomaa, Simple splicing systems, *Discrete Applied Mathematics*, 1997, 84; 1998, 145-163 .

[7] Gh.Paun. Regular Extended H systems are computationally universal, *Journal on Automata, Languages and Combinatorics*,1996 , 27-36.

[8] Gh.Paun, Computing by Splicing. How simple rules?. *Bulletin of the EATCS*, 60, 1996 , 145-150.

[9] G.Rozenberg, A.Salomaa, *Handbook of Formal Languages*, 3 volumes, Springer-Verlag, Heidelberg, 1997.