P Systems Generating Pattern Languages

Christopher Kezia Parimalam
Department of Mathematics
Queen Mary's College
Chennai-600004

Emerald Princess Sheela J. D.
Department of Mathematics
Queen Mary's College
Chennai-600004

D. G. Thomas
Department of Mathematics
Madras Christian College
Chennai 600 059

ABSTRACT
P system is an interesting computing model of natural computing exhibiting very nice decidability and complexity results. As a language generating device, P system produces formal languages. In this paper, a variant of P system called rewriting P system is considered and its generative power of yielding pattern languages is investigated.

General Terms
Membrane structure, P systems, context free grammars, languages

Keywords
Pattern grammars, rewriting P systems with unique parallelism

1. INTRODUCTION
The theory of P system (or membrane computing) is a recent and intensively investigated area of Natural computing, a field of research which tries to imitate nature in the way it computes. Neural networks, Generic algorithms and DNA computing are three areas of Natural computing already well established. Paun [1] introduced a new computability model, called a P system, which is a distributed highly parallel theoretical model based on membrane structure and the behavior of living cells. The basic model processes multi-sets of objects in the regions that are defined by hierarchical arrangement of membranes, by evolution rules associated with the regions. One of the branches of membrane computing is rewriting P system [2] in which objects in membranes are described by strings and these strings are processed by rewriting rules or other string manipulating operations.

A pattern grammar is a generative device starting from a finite set of given strings called axioms and replacing them by the variables in a given set of patterns and continuing the process with the current set of strings obtained by such operations. The class of pattern languages generated by pattern grammar is incomparable with a class of context free languages. A pattern grammar is also a generalization of Marcus contextual grammars [3].

In [4] a P system for generating pattern languages with single pattern is defined. We now extend our study to define P system for generating pattern grammars with more than one pattern variable and more than one axiom.

2. PRELIMINARIES
A pattern language is a string over $\Sigma \cup \Delta$, where Σ is a finite alphabet and Δ is a finite set of elements called variables. We obtain the pattern language by substituting in the pattern all the occurrences of variable by arbitrary strings in Σ^*.In this section we recall the notions of pattern grammar and a P system.

Definition 2.1A pattern grammar [2] is a quadruple

$$G = (\Sigma, \Delta, A, P)$$

Where, $\sum$, Δ are alphabets as above, $A \subseteq \Sigma^*$, is a finite set (its elements are called axioms) and P is a finite subset of $(\Sigma \cup \Delta)^* \Delta\Sigma^*$,. The strings in P are called patterns (every pattern must contain at least one variable occurrence). For a set P of patterns and a language $L \subseteq \Sigma^*$, denote

P(L) =

$$\{u_1 x_{i_1} u_2 x_{i_2} \ldots . u_k x_{i_k} u_{k+1} : u_1 \delta_{i_1} u_2 \delta_{i_2} \ldots . u_k \delta_{i_k} u_{k+1} \in P, u_i \in \Sigma^*, 1 \leq i \leq k+1, \delta_{i_j} \in \Delta, x_{i_j} \in L, 1 \leq j \leq k\}$$

(P (L) is the set of strings obtained by replacing each occurrence of variables in patterns of P by strings in L, the different occurrences of the same variable being replaced by the same string.)

The language generated by G, denoted by L (G) is the smallest language $L \subseteq \sum^*$ for which we have

i. $A \subseteq L$
ii. $P(L) \subseteq L$

Thus, L(G) consists of all strings which can be obtained starting from axioms and using finitely many times the patterns, in the way previously described. Note that this language exists for any A and P, since L(G) = $A \cup P(A) \cup P(P(A)) \cup \ldots$

A language L(G) as above is called a pattern language and its family is denoted by PL .

Example 2.1$G_1 = (\{a, b\}, \{\delta\}, \{ ab \}, \{a\delta b\})$
Starting from ab we obtain in turn, a^2b^2, a^3b^3,…
$L(G_1) = \{a^n b^n : n \geq 1\}$.

Theorem 2.1[3]The family of pattern languages is incomparable with the family of context free languages.

3. REWRITING P SYSTEM WITH UNIQUE PARALLELISM
A variant of a rewriting P system called a rewriting P system with unique parallelism is defined in this section.

Definition 3.1[5]A rewriting P system with unique parallelism (of degree $m \geq 1$) is a construct

$$\Pi = (V, T, \mu, w_1, \cdots w_n, R_1, \cdots R_n, i_{out})$$

Where

- V is the total alphabet of the system;
- $T \subseteq V$ is the terminal alphabet;
- μ is a membrane structure;
- $w_i, 1 \leq i \leq n$, are finite language over V, representing strings initially present in the regions 1…n of μ;
- $R_i, 1 \leq i \leq n$, is a finite set of rewriting *rules* of the form $X \rightarrow v(tar)$, where $X \in V, v \in V^*$ and *tar* $\in \{here, out\} \cup \{in_j \mid 1 \leq i \leq m\}\}$

In a rewriting P system each string which can be rewritten, is rewritten in each unit time. Since the strings and the rules are localized to the regions, the strings in a given region are rewritten only by rules in that region. Thus starting from a given configuration, we pass to another configuration; a sequence of transitions form a computation. We consider as successful computation only the halting ones (the computation which reach a configuration where no rule can be applied). The result of a halting computation consists of all strings over T which are sent out of the system during the computation.

What is specific to this system is the way of rewriting the strings and to pass them through the membranes. Specifically, each string is rewritten in a unique parallel manner. For each string all the occurrences of exactly one symbol is rewritten according to exactly one rule, which is non-deterministically chosen between all rules that can be applied to the symbol. Thus given a string $w = x_1 a x_2 a x_3 \cdots x_n a x_{n+1}$ with $x_i \in (V - \{a\})^*, i = 1, \cdots, n+1$, and one context-free rule $r: a \rightarrow \alpha$, we obtain the string $w' = x_1 \alpha x_2 \alpha x_3 \cdots x_n \alpha x_{n+1}$ in one parallel rewriting step.
$L(\Pi)$ denotes the language generated by a rewriting P system Π. The family of all languages of this type, generated by m membranes, is denoted by $UPRP_m$. If no bound on the number of membranes is considered then m is replaced by subscript *.

3.1 Rewriting P System with Unique Parallelism for Pattern Languages

In this section a P system with unique parallelism $UPRP_1$(PL) to generate pattern languages generated by pattern grammars is defined.

Definition 3.1.1A rewriting P system with unique parallelism for a pattern language generated by a pattern grammar

$G = (\Sigma, \Delta, A, P)$ is defined by $\Pi = (V, T, \mu, w_1, (R_1, \rho_1), i_1)$ where
Vis an alphabet set $\Sigma \cup \{\eta_i, \eta_i'\}; \eta_i$ is a special symbol$\notin V \cup T$ is a pattern symbol corresponding to the symbol l$\delta_i \in \Delta$.
$T \subseteq V$is the set of output alphabet
$\mu = [_1\,]_1$
w_1= {A in G}$\cup \left\{ u_{i1}\eta_1 u_{i2} \cdots u_{ij}\eta_j u_{ij+1}, : P_i = u_1\delta_1 u_2 \cdots u_j\delta_j u_{j+1} \in G, \text{ where } 1 \leq i \leq n, 1 \leq j \leq m \right\}$
R_1- {$r_1: \eta_i \rightarrow \eta_i'$ for every pattern variable in G

$r_2: \eta_i' \rightarrow \{A\}$

$r_3: \eta_i' \rightarrow u_1\eta_1' u_2 \cdots u_j\eta_j' u_{j+1} : 1 \leq i \leq n\}$

ρ_1- $(r_2, r_3) > r_1$.

The rules are rewritten using rewriting with unique parallelism. The set of words generated by a rewriting P system with unique parallelism is denoted by PL(Π). The family of all pattern languages generated by systems Π as above with at most m membranes is denoted by $UPRP_m(PL)$.

Example 3.1.1 Consider the Pattern grammar G = ({a,b}, $\{\delta_1, \delta_2\}, \{a, b, \lambda\}, \delta_1\delta_2$)

The rewriting system with unique parallelism $UPRP_1(PL)$ is
$\Pi = (V, T, [_1]_1, w_1, (R_1, \rho_1), i_1)$
$V = \{a, b, \eta_1, \eta_2, \eta_1', \eta_2'\}$
$T = \{a, b\}$
$w_1 = \{a, b, \lambda, \eta_1\eta_2\}$
$R_1 = \{r_1: \eta_1 \rightarrow \eta_i', \eta_2 \rightarrow \eta_2'$

$r_2: \eta_1' \rightarrow a, \eta_1' \rightarrow b, \eta_2' \rightarrow a, \eta_2' \rightarrow b$

$r_3: \eta_1' \rightarrow \eta_1'\eta_2'; \eta_2' \rightarrow \eta_1'\eta_2'\}$

$\rho_1 = (r_2, r_3) > r_1$

$L(\Pi) = L(G) = \{a, b\}^*$

Theorem 3.1.1 For every pattern language generated by a pattern grammar, there exist a rewriting P system with unique Parallelism generating it, i.e. $UPRP_1(PL) = PL$

Proof: Consider a pattern grammar $G = (\Sigma, \Delta, A, P)$ with
Σ – set of alphabet
Δ – set of pattern variables
$A = \{a_1, \cdots a_k\}$
$P = \{ u_{11}\delta_1 \cdots u_{1n}\delta_n u_{1n+1} \cdots u_{n1}\delta_1 \cdots u_{nn}\delta_n u_{nn+1} \}$ where some $\delta_i = \delta_j$and some $u_{ij}, \delta_i = \emptyset$.

The $UPRP_1$ system to generate the L(G) is

$\Pi = (V, T, [_1]_1, , w_1, (R_1, \rho_1), i_1)$,With
$V = \{a, b, \eta_1 \cdots \eta_n, \eta_1' \cdots \eta_n'\}$
T = set of the terminal alphabet
w_1
$= \{a_1, \cdots a_k, u_{11}\eta_1 \cdots u_{1n}\eta_n u_{1n+1}, \cdots, u_{n1}\eta_1 \cdots u_{nn}\eta_n u_{nn+1}\}$

$R_1 = \{r_1: \eta_i \rightarrow \eta_i': 1 \leq i \leq n$

$r_2: \eta_i' \rightarrow a_j: 1 \leq i \leq n, 1 \leq j \leq k$

$r_3: \eta_i' \rightarrow u_{i1}\eta_1' u_{i2} \cdots u_{in}\eta_n' u_{in+1}$; $1 \leq i \leq n$, where $\eta_i' = \eta_j'$ *if* $\eta_i = \eta_j$ and $u_{ij}, = \emptyset$ if $u_{ij}, \delta_i = \emptyset$.}

$\rho_1 = (r_2, r_3) > r_1$

Initially all the set of axiom is present in the membrane hence $A \in L(\Pi)$. Now, the rules of R_1become active and by priority only r_1 can be applied at this stage as there are no η_i' to start with. Now a rule from r_1 is chosen and each string inw_1, is rewritten using the rule,$\eta_i \rightarrow \eta_i'$, the strings which do not contain η_i remain unaltered. In the next transition rules of r_2 or r_3 is applied because of priority.

Case (i) If the rules of r_2 are used for rewriting then all the η_i's are rewritten in the subsequent steps sequentially, each step rewrites the η_i' by an axiom non-deterministically chosen. Thus the strings terminate where in each string the halting computation is a word of P(A).

Case (ii) If the rules from r_3 are chosen, then each string is rewritten by the same rules, and at some stage the η_i's in the

strings should be rewritten using rule r_2 as otherwise the computation will never halt. If r_3 is used once and then terminated by using rules of r_2, the computation results in strings$w_i \in P(P(A)$. If rules of r_3 are used twice then all the computations result in strings $w_i \in P(P(P(A))$.

If there are still strings that can be rewritten, then those are the strings that did not have a common η_i to be rewritten by r_1. Hence the other rules of r_1 is used for rewriting and then r_2 and r_3 are chosen as discussed in cases (i) and (ii) resulting in string belonging to P(A)or P(P(A) or P(P(P(A)). Thus any halting computation contain strings that belong to P(A)∪ $P(P(A) \cup P(P(P(A))$. We see that $L(\Pi) =$ L(G)and $UPRP_1(PL)$ = PL.

Note: When the pattern has only one variable the priority rules are not needed. If the membranes are increased so that each variable is active in a particular membrane, then also the priority rules have no importance.

Example 3.1.2

$G = (\{a, b\}, \{\delta\}, \{ab, ba\}, \{\delta a, \delta b, a\delta, b\delta\})$

We construct a rewriting P system

$\Pi = (\{a, b, \eta\}, \{a, b\}, [_1]_1, w_1, R_1)$

$w_1 = \{a, b, \eta a, \eta b, a\eta, \eta b\}$

$R_1 = \{r_1: \eta \to a, \eta \to b$

\ $r_2: \eta \to \eta a, \eta \to \eta b, \eta \to a\eta, \eta \to b\eta\}$

$L(\Pi) = \{w \in \{a, b\}^+; |w|_a \geq 1, |w|_b \geq 1\}$ = L(G).

3.2 Comparison Results

The generative power of the rewriting P systems for pattern languages is brought out in the following theorem. The following Theorem 3.2.1 shows that the rewriting P systems with unique parallelism for pattern languages with two membranes is more powerful more than pattern grammars generating pattern languages.

Theorem 3.2.1 PL ⊂ $UPRP_2$(PL).

Proof:$UPRP_2(PL)$ allows at most two membranes, the inclusion in statement (ii) is clear. The proper inclusion is due to the fact that a language $L = \{a^{2^n}b^{2^n}: n \geq 0\}$ cannot be generated by a pattern grammar [2]. But the following system in $UPRP_2(PL)$ generates L.

$$\Pi = (\{a, b, \eta_1, \eta_2\}, \{a, b\}, [_1[_2]_2]_1, w_1, L_2, R_1, R_2, i_1)$$

$w_1 = \{ab, \eta_1\eta_2\}$
$w_2 = \phi$
$R_1 = \{r_1: \eta_1 \longrightarrow \eta_1\eta_{1_{in}}$
$r_2: \eta_1 \longrightarrow a, \eta_2 \longrightarrow b\}$
$R_2 = \{r_1: \eta_2 \longrightarrow \eta_2\eta_{2_{out}}\}$
$UPRP_2 = \{a^{2^n}b^{2^n}: n \geq 0\}$.

In the above system $\{a^{2^n}: n \geq 0\}$ is a Pattern language with $\delta_1\delta_1$ as the pattern and {a} the axiom and $\{b^{2^n}: n \geq 0\}$is a pattern language with $\delta_2\delta_2$ as the pattern and {b} the axiom.A$UPRP_2(PL)$ with two membranes, with membrane 2 inside membrane 1 is constructed. Starting with the pattern $\eta_1\eta_2$ in the region1, apply the pattern rule $r_1$in region 1 to η_1, sending the resultant string to region 2 making $\eta_1's$ to double and in region 2, the η_2's are doubled using $r_1$in region 2 and sent back to region 1 and is terminated using the r_2 in R_1, and the resultant is a^2b^2. Since no more rules can be applied, the string is added to the language. The process repeats and all strings with equal powers of 2's in a's and b's are generated. The result is achieved by making an membrane active to only to one pattern variable.

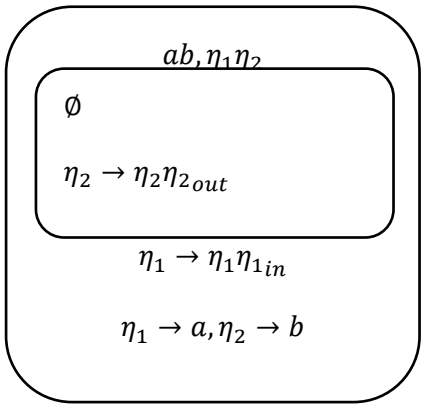

Fig 1: Rewriting P system with unique parallelism to generate $\{a^{2^n}b^{2^n}: n \geq 0\}$

Theorem 3.2.2 $UPRP_2(PL) \backslash UPRP_1(PL) \neq \emptyset$

Proof: Consider the non-context free language

$\{a^nb^nc^n: n \geq 0\}$The rewriting P system with unique parallelism with two membranes $UPRP_2(PL)$to generate the above non-context free language is the construct.

$$\Pi = (\{a, b, c, \eta_1, \eta_2, \eta_3\}, \{a, b\}, [_1[_2]_2]_1, w_1, w_2, R_1, R_2, i_1)$$

$w_1 = \{abc, a\eta_1bc\eta_2\}$
$w_2 = \phi$
$R_1 = \{r_1: \eta_1 \longrightarrow a\eta_1b_{in}$
$r_2: \eta_1 \longrightarrow ab, , \eta_2 \longrightarrow c\}$
$R_2 = \{r_1: \eta_2 \longrightarrow c\eta_{2_{out}}\}$

$UPRP_2(PL) = \{a^nb^nc^n: n \geq 0\}$.

The language cannot be generated by $UPRP_1(PL)$. Note that every word in the language has for some n≥ 0, 'n', a's and b's are followed by n c's. If there are less than two membranes then the more than one rule of r_1 is present in the membranes, when each 'a' and 'b' is increased c's need not be equally increased and vice versa. This will generate words not in the given language. Hence one membrane is not enough to generate the given language.

Theorem3.2.3 For n ≥ 2, $UPRP_n(PL) \setminus UPRP_{n-1}(PL) \neq \emptyset$

Proof: Consider a rewriting P system with unique parallelism for pattern languages $\Pi = (\{a_1, \cdots, a_n, \eta_1, \cdots, \eta_n\}, \{a_1 \cdots a_n\}, [_1[_2]_2 \cdots [_n]_n]_1, w_{1,} \cdots, w_n, R_1, \cdots, R_n, i_1)$

$w_1 = \{a_1 \cdots a_n, \eta_1 \cdots \eta_n\}$

$R_1 = \{\eta_1 \to \eta_1\eta_{1_{in2}}, \eta_1 \to a_1, \cdots, \eta_n \to a_n\}$

The elementary membranes inside the skin membrane contain the rules as given below

For $2 \leq i \leq n-1$,$R_i = \{\eta_i \to \eta_i\eta_{i_{in\,n+1}}\}$

$R_n = \{\eta_n \to \eta_n\eta_{n_{in\,1}}\}$

Π generates the language $\{a_1{}^{2^n} \cdots a_n{}^{2^n}: n \geq 0\}$. In each membrane only one pattern variable is raised to the power of 2, and pushed to the next membrane while the rest of the pattern variables remain the same. In each cycle every pattern

variables are raised to the power of 2, which cannot be generated by membranes less than n.

Theorem3.2.4 CF $\subset UPRP_m(PL)$

Proof: Let G = (N,T,S,P) be a context free grammar with no null productions. The equivalent unique parallel rewriting P system for pattern languages is a construct

$\Pi = \left(V, T, [_1[_2]_2 \cdots [_m]_m]_1, w_1, (R_1, \rho_1) \cdots (R_m, \rho_m), i_1\right)$

Where

V= $\{\eta_1, \cdots, \eta_n : \eta_i \; corresponds \; to \; A_i \epsilon \; N\}$

T= T

The initial word in the skin membrane is a pattern P which is the right hand side of the production of the start symbol with the non-terminals replaced by the corresponding pattern variable. The rewriting rules in the skin membrane contain all the rules $\eta_i \longrightarrow x$ for $A_i \longrightarrow x, x \in T^* \; in \; G$. Every membrane contains the rule $\eta_i \longrightarrow \alpha_{in_{i+1}}$, whenever $A_i \longrightarrow \alpha$, $\alpha\epsilon(N \cup T)^*$ contains a non-terminal A_{i+1} on the right hand side is in G. Every membrane other than the Skin membrane has a rule $\eta_i \longrightarrow \alpha_{out}$, if $A_i \longrightarrow \alpha, \alpha\epsilon(N \cup T)^*$ is in G and has no non-terminal A_{i+1} on the right hand side . In the Skin membrane the rule is $\eta_i \longrightarrow \alpha$ for such a production in G. The relation L(G) = L(Π)

Hence, $CF \subset UPRP_m(PL)$.

Example 3.2.1 Consider a context free grammar

$G = (\{A_1, A_2, A_3\}, \{a, b\}, P, S\}$where

$P = \{A_1 \rightarrow abA_2, A_2 \rightarrow bbA_3b, A_3 \rightarrow aaA_2a, A_2 \rightarrow bba\}$

The$UPRP_2(PL)$ is a construct

$$\Pi = (\{a, b, \eta_1, \eta_2, \eta_3\}, , [_1[_2]_2]]_1, w_1, w_2, R_1, R_2, i_1)$$

$w_1 = ab\eta_2$

$w_2 = \phi$

$R_1 = \{\eta_2 \rightarrow bb\eta_3 a_{in}, \eta_2 \rightarrow bba\}$

$R_2 = \{\eta_3 \rightarrow aa\eta_2 b_{out}\}$

Thus $L(\Pi) = L(G) = \{ab(bbaa)^n\text{bba}(ba)^n : n \geq 0\}$.

4. CLOSURE PROPERTIES

The family of Pattern languages is not closed under union with singletons, catenation with singletons, Intersection, complementation, Kleene+, inverse morphisms [3]. But Rewriting P system with unique parallelism for Pattern languages are closed under all the above. The rewriting P systems with unique parallelism is also closed under the operation of Union and concatenation.

Theorem 4.1 The rewriting P systems with unique parallelism for pattern languages are closed under the operations of Union with singletons, catenation with singletons, Intersection, complementation, Kleene+, inverse morphism, Union and concatenation of languages.

Proof:

Closure under Union with Singletons.

Let $G = (\Sigma, \Delta, A, P)$ be a Pattern grammar generating a Pattern language L(G).There exist a rewriting P system $UPRP_1(PL)$, $\Pi = (V, T, \mu, w_1, (R_1, \rho_1), i_1)$ such that $L(\Pi) = \text{L(G)}$. Let 'b' be the singleton there is no Pattern grammar G that can generate $L(G) \cup \{b\}$. But a rewriting P system with unique parallelism $UPRP_1(PL)$, $\Pi_1 = (V, T, \mu, w_1 \cup \{a\}, (R_1, \rho_1), i_1)$ generates $L(\Pi_1) = L(G) \cup \{b\}$. The construct is done by just taking the initial strings in Π union the singleton needed.

Example 4.1

$G_1 = (\{a\}, \{\delta\}, \{a\}, \{a\delta\})$; $L(G) = \{a^n | n \geq 1\}$.

But $\{a^n \cup b\} \notin PL$ i.e, there is no Pattern grammar G to generate the given language.

$\Pi_1 = (\{a, b\}, \{\eta_1\}, [_1]_1, \{a, a\,\eta_1\}, R_1, i_1)$

$R_1 = \{\eta_1 \rightarrow a\,\eta_1, \eta_1 \rightarrow a\}$

$L(\Pi_1) = \text{L}(\text{G}_1)$.

Define $\Pi_1' = (\{a, b\}, \{\eta_1\}, [_1]_1, \{a, b, a\,\eta_1\}, R_1, i_1)$

R_1 is as defined in Π_1.

Then, $L(\Pi_1') = \text{L(G)} \cup \{b\}$.

Closure under Catenation with singletons

Let G and Π be as defined in (1). bL(G) or L(G)b is not a pattern language. The rewriting P system with unique parallelism $UPRP_1(PL)$, $\Pi_2 = (V, T, \mu, w_1, (R_1, \rho_1)\; i_1)$, generates$L(\Pi_2) = bL(G)$. This can be obtained by taking the initial string in the system as {singleton}A U {singleton}P, where P, are the patterns defined in G.

Example 4.2

Consider $G_2 = (\{a, b\}, \{\delta_1\}, \{ab\}, \{\delta_1 b\})$

$\Pi_2 = (\{a, b\}, \{\eta_1\}, [_1]_1, \{ab, \eta_1 b\}, R_1, i_1)$

$\text{R}_1 = \{\eta_1 \rightarrow ab,\ \eta_1 \rightarrow \eta_1 b\}$

$L(\Pi_2) = L(G_2) = ab^+$

But ab^+a is not a PL.

The construct $UPRP_1(PL)$, defined by

$\Pi_2' = (\{a, b\}, \{\eta_1\}, [_1]_1, \{aba, \eta_1 ba\}, R_2, i_1)$
$R_2 = R_1$
generates the language $L(\Pi_2') = ab^+a$

Closure under Intersection

Let $G_1 = (\Sigma_1, \Delta_1, \text{A}_1, \text{P}_1)$ and $G_2 = (\Sigma_2, \Delta_2, \text{A}_2, \text{P}_2)$ be two pattern grammars and $\Pi_1 = (V_1, T_1, [_1]_1, w_{11}, (R_{11}, \rho_{11}), i_1)$ and $\Pi_2 = (V_2, T_2, [_1]_1, w_{21}, (R_{21}, \rho_{21}), i_1)$ be their corresponding rewriting P systems with unique parallelism such that $L(\Pi_1) = L(\text{G}_1)$and$(\Pi_2) = L(G_2)$. $\text{L}(\text{G}_1) \cap L(G_2)$ is not a pattern language.

But $\Pi = (V, T, [_1]_1, w, (R, \rho), i_1)$

V= V_1

T= T_1
$w = \mathrm{A}_1 \cap A_2 \cup A_3$
Where $A_3$is the set of patterns in w_{11} that produce strings in $\mathrm{L}(\mathrm{G}_1) \cap L(G_2)$.
$(R, \rho) = (R_{11}, \rho_{11})$
Thus, $\mathrm{L}(\Pi) = \mathrm{L}(\mathrm{G}_1) \cap L(G_2)$.

[The same can be done by choosing patterns in Π_2 and its rewriting rules.]

Example 4.3

$G_1 = (\{a, b\}, \delta_1, \{aa, ab\}, \{\delta_1 a, \delta_1 b\}$

$\Pi_1 = (\{a, b, \eta_1\}, \{a, b\}, [_1]_1, w_{11}, R_{11}, i_1)$

$w_{11} = \{aa, ab, \eta_1 a, \eta_1 b\}$

$R_{11} = \{r_1 : \eta_1 \rightarrow aa, \eta_1 \rightarrow ab,$

$r_2 : \eta_1 \rightarrow \eta_1 a, \eta_1 \rightarrow \eta_1 b\}$

$L(G_1) = L(\Pi_1) = \{a\}\{a, b\}^+$

$G_2 = (\{a, b\}, \delta_2, \{aa, ba\}, \{a\delta_2, b\delta_2\})$

$\Pi_2 = (\{a, b, \eta_2\}, \{a, b\}, [_1]_1, w_{21}, R_{21}, i_1)$

$w_{21} = \{aa, ba, a\eta_2, b\eta_2\}$

$R_{21} = \{r_1 : \eta_2 \rightarrow aa, \eta_2 \rightarrow ba$

$r_2 : \eta_2 \rightarrow a\eta_1, \eta_2 \rightarrow b\eta_2\}$

$L(G_2) = L(\Pi_2) = \{a, b\}^+\{a\}$

Now, $\Pi = (\{a, b, \eta_1\}, \{a, b\}, [_1]_1, w, R, i_1)$

$w = \{aa, \eta_1 a\}$

R= $\{r_1 : \eta_1 \rightarrow aa, \eta_1 \rightarrow ab$

$r_2 : \eta_1 \rightarrow \eta_1 a,\ \eta_1 \rightarrow \eta_1 b\}$

$L(\Pi) = \{a\}\{ab\}^+\{a\} = L(G_1) \cap L(G_2)$

Closure under Inverse morphism

Let $G = (\Sigma, \Delta, A, P)$ be a pattern grammar generating PL, and h: $\Sigma^* \rightarrow \Sigma^*$be a morphism defined by $\mathrm{h}(w_i) = w_i'$. Then $h^{-1}(L(G))$ is not a PL. The construct $UPRP_1$(PL), defined by $\Pi = (V, T, \mu, w_1, (R_1, \rho_1), i_1)$is a rewriting P system with unique parallelism where the pattern and the rewriting rules are defined as follows. For $w \in L(G)$, factorize the word into $w = {w_1}'^{n1} \cdots {w_n}'^{nn}$ where $\mathrm{h}(w_i) = w_i'$,then there are three possible cases for ni,

(i) If $ni = 0$, then w_i is a part of the initial string present in the membrane
(ii) If $ni = 1$, then there is a pattern variable η_i in the initial string and a rewriting rule $\eta_i \longrightarrow w_i$, in the membrane.
(iii) If $ni > 1$, then there is a pattern variable η_i in the initial string and a set of rewriting rule $\eta_i \longrightarrow w_i\eta_i$, $\eta_i \longrightarrow w_i$, in the membrane.

The construct$L(\Pi) = h^{-1}(L(G))$.

Example 4.4

$G = (\{a, b, c\}, \{\delta_1\}, \{abc\}, \{\delta_1 bc\})$; $L(G) = \{a\}\{bc\}^+$

Consider a morphism $h: \{a, b, c\}^* \rightarrow \{a, b, c\}^*$; defined by

$h(a) = ab, h(b) = cb, h(c) = c$

Then $h^{-1}(L(G))$= {ab^*c } is not a pattern language.

We define But $\Pi = (V, T, [_1]_1, L, (R, \rho), i_1)$

$V = \{a, b, c, \eta_1\}$

$T = \{a, b, c\}$

$w_1 = \{ac, a\eta_1 c\}$

$R_1 = \{\eta_1 \rightarrow b\eta_1, \eta_1 \rightarrow$b}

$h^{-1}(L(\Pi))$= {ab^*c}

Closure under Kleene +

Let $G = (\Sigma, \Delta, A, P)$ be a pattern grammar generating the pattern language L= L(G) . The rewriting P system with unique parallelism $\Pi = (V, T, [_1]_1, L_1, (R_1, \rho_1), i_1)$. Where

$V = \Sigma \cup \{\eta_i, \eta_i' \ for\ every \quad \delta_i \in \Delta\} \cup \{\eta_{11}, \eta_{12}, \eta_{11}'\}$

$T = \Sigma$

$w_1 = A \cup \{\eta_{11}\eta_{12}\}\}$

$R_1 = \{r_1 : \eta_i \rightarrow \eta_i'$ for every i

$r_2 : \eta_i' \rightarrow a_k$for every $a_k \in$ A

$r_3 : \eta_{11} \rightarrow {\eta_{11}}', {\eta_{12}} \rightarrow {\eta_{11}}'$

$r_4 : \eta_{11}' \rightarrow P_j'$, for every$P_j \in P$,

$(where\ P_j', is\ a\ pattern\ corresponding\ to\ P_j\ in\ which\ each\ \eta_i\ is\ replaced\ by\ \eta_{i'}\ \}$

Thus, $L(\Pi) = L^+$

Example 4.5

Let $G = (\{a, b\}, \{\delta_1\}, \{ab\}, \{a\ \delta_1 b\})$; $L(G) = \{a^n b^n : n \geq 1\}$

There is no pattern grammar that generates $L^+ = (a^n b^n)^+$

$\Pi = \left(\{a, \mathrm{b}, \eta_1, \eta_1', \eta_2\}, \{a, b\}, [_1]_1, w_1, R_1, i_1\right)$

$w_1 = \{ab, \eta_1\eta_2\}$

$R_1 = \{ r_1 : \eta_1 \longrightarrow {\eta_1}'$

$r_2 : \eta_1' \longrightarrow$ ab

$r_3 : \eta_2 \longrightarrow {\eta_1}'$

$r_4 : \eta_1' \longrightarrow a\eta_1' b\}$

Thus, $L(\Pi) = \{(a^n b^n)^+ : n \geq 1\}$

Closure under Union

Proof: Let $G_1 = (\Sigma_1, \Delta_1, A_1, P_1)$ and $G_1 = (\Sigma_2, \Delta_2, A_2, P_2)$ be two pattern grammars generating pattern languages L_1 and L_2 and let $\Pi_1 = (V_{11}, T_{11}, [\,]_1, w_{11}, (R_{11}, \rho_{11}), i_1)$and $\Pi_2 = (V_{21}, T_{21}, [\,]_1, w_{21}, (R_{21}, \rho_{21}), i_1)$ be their rewriting P systems with unique parallelism generating L_1 and L_2respectively.
The $UPRP_1(PL)$, generating $L_1 \cup L_2$ is defined as

$\Pi = (V, T, [\,]_1, w_1, (R_1, \rho_1), i_1)$

$V = V_1 \cup V_2$

$T = T_1 \cup T_2$

$w_1 = \{w_{11}, w_{21}\}$

$R_1 = R_{11} \cup R_{21}$

$\rho_1 = \rho_{11} \cup \rho_{21}$

$L(\Pi) = L_1 \cup L_2$

Initially the membrane contains all the axioms and the patterns in both G_1 andG_2 which independently generate strings in $L(G_1)$ and $L(G_2)$, using rules R_{11} and R_{21} respectively.

Example 4.6

$G_1 = (\{a\}, \{\delta_1\}, \{a\}, \{a\delta_1\})$ and $G_2 = (\{b\}, \{\delta_2\}, \{a\}, \{a\delta_2\})$ be the pattern grammars generating $L(G_1) = \{a^*\}$ and $L(G_2) = \{b^*\}$, the rewriting P systems with unique parallelism for the pattern languages defined above are

$\Pi_1 = (\{a\}, \{\eta_1\}, [\,]_1, \{a, \lambda, a\,\eta_1\}, R_{11}, i_1)$
$R_{11} = \{r_1 : \eta_1 \longrightarrow a\eta_1$
$r_2 : \eta_1 \longrightarrow a\}$
$\Pi_2 = (\{b\}, \{\eta_2\}, [\,]_1, \{b, b\,\eta_2\}, R_{21}, i_1)$
$R_{21} = \{r_1 : \eta_2 \longrightarrow b\eta_2$
$r_2 : \eta_2 \longrightarrow b\}$
The construct for the union of the languages is defined by

$\Pi = (\{a, b\,\eta_1, \eta_2\}, \{a, b\}, [\,]_1, \{a, b.\lambda, a\eta_1, b\eta_2 R_1, i_1)$

$R_1 = \{r_1 : \eta_1 \longrightarrow a\eta_1, \eta_2 \longrightarrow b\eta_2$

$r_2 : \eta_1 \longrightarrow a, \eta_2 \longrightarrow b\}$

Then, $L(\Pi) = a^* \cup b^*$

Closure under Concatenation

Let $G_1$and G_2 be as defined in closure under union, then the rewriting P system with unique parallelism generating $L_1 \circ L_2$ is defined by $\Pi_3 = (V_3, T_3, [\,]_1, w_{31}, (R_{31}, \rho_{31}), i_1)$

$V_{31} = V_1 \cup V_2$

$T_{31} = T_1 \cup T_2$

$w_{31} = \{\, concatenated\ strings\ in\ A_1\ and A_2\} \cup \{P_1P_2$ with the $\delta_i's$ $replaced\ with\ \eta_i's.\}$

$R_{31} = R_{11} \cup R_{21}$

$\rho_{31} = \rho_{11} \cup \rho_{21}$

Initially all concatenated strings of the axioms in both the languages are present in the membrane. The pattern is the concatenation of the pattern in the grammar and is rewriting using their corresponding rules in the P system.

Thus $L(\Pi_3) = L_1 \circ L_2$.

Example 4.7

$G_1 = (\{a, b, c\}, \{\delta_1\}, \{c\}, \{a\delta_1 b\})$ and

$G_2 = (\{a, b, d\}, \{\delta_2\}, \{d\}, \{a\delta_2 b\})$be the pattern grammars and the rewriting P systems with unique parallelism for L(G_1) and L(G_2) is defined respectively by Π_1 and Π_2 as

$\Pi_1 = (\{a, b, c, \eta_1\}, \{\eta_1\}, [\,]_1, \{c, a\,\eta_1 b\}, R_{11}, i_1)$

$R_{11} = \{r_1 : \eta_1 \longrightarrow a\eta_1 b$

$r_2 : \eta_1 \longrightarrow c\}$

$\Pi_2 = (\{a, b, d, \eta_2\}, \{\eta_2\}, [\,]_1, \{d, a\,\eta_2 b\}, R_{21}, i_1)$

$R_{21} = \{r_1 : \eta_2 \longrightarrow a\eta_2 b$

$r_2 : \eta_2 \longrightarrow d\}$

$\Pi = (\{a, b, c, d\,\eta_1, \eta_2\}, \{a, b\}, [\,]_1, \{c, d, a\eta_1 b a\eta_2 b, R_1, i_1)$

$R_1 = \{r_1 : \eta_1 \longrightarrow a\eta_1 b, \eta_2 \longrightarrow a\eta_2 b$

$r_2 : \eta_1 \longrightarrow a, \eta_2 \longrightarrow b\}$

$L(\Pi) = \{a^m c b^m a^n d b^n : n, m \geq 0\} = L(G_1) o L(G_2)$

5. CONCLUSION

In this paper a rewriting P system with unique parallelism has been introduced to obtain pattern languages. In fact these pattern languages are generated by pattern grammars with purely context free rules with m patterns and n pattern variables. We proved that the generative capacity of such P systems is higher than that of the pattern grammars. Our investigation of study is only with unique parallelism. For our future work we use other parallelism modes. Again in a P system, the depth is the maximum number of membranes in the nesting of membranes [6]. In this paper, the results have been obtained with a depth of the P system equal to 2. Another direction of future investigation is to increase the depth of the P system greater than two and study further properties.

6. ACKNOWLEDGEMENT

The authors wish to thank Dr. T. Robinson for his motivation, guidance and discussion in preparing this paper.

7. REFERENCES

[1] Paun Gh., 2000 Computing with membranes, Journal of Computer and System Sciences, 61**,** 108-143 and Turrku Center for Computer Science- TUCS Report No.208, (1998).

[2] M. Mutyam, 2005 Rewriting P systems: improved hierarchies, Theoretical Computer Science 334,pp 161-175.

[3] J.Dassow, Gh.Paun, A. Salomaa. 1993 Grammars Based on Patterns, International Journal of Computer Science, Vol 4 No: 1, pp 15-30

[4] Christopher Kezia Parimalam, J.D. Emerald, 2014 Learning of P systems for subclass of pattern languages, Proceedings of the Asian Conference on Membrane Computing, pp 162-170

[5] Gheorghe Paun, GrzegorzRozenberg, ArtoSalomaa, 2010 The Oxford Handbook of Membrane Computing, Oxford University Press,pp 168-197.

[6] K.G. Subramanian, S.Hemalatha C. Sri HariNagore, M. Margenstern, 2007 On the Power of P Systems with Parallel rewriting and Conditional Communication, Romanian Journal Of Information Science And Technology, Volume 10, Number2,PP 137-144.