Regulated Contextual KP System

Ajitha P S* and Dersanambika K.S †

*Department of Mathematics,

College Of Engineering,Kottarakara, Kerala 691531, India

† Department of Mathematics,

Fatima Mata National College, Kollam 691 001, India

Abstract

The basic Chomsky hierarchy of language families was obtained by imposing restrictions on the form of productions. Regulated rewriting is a natural generalization to impose restrictions also in the use of productions. In this paper, we consider contextual way of handling string objects with regulated use of productions.

Keywords: Psystem, KP System, Contextual grammar ,Matrix grammar, Programmed grammar.

1 I NTRODUCTION

Membrane computing has emerged as a vigorous research field as a part of natural computing or unconventional computing. It is a nature inspired computational paradigm including a large variety of models, called membrane systems, well investigated from a computational perspective, especially with respect to their computational power and complexity aspects. The key features of a membrane system are a set of compartments delimited by membranes, multiset of objects contained in these regions, transformation and communication rules describing interactions between objects and a strategy for evolving the system. Starting from an initial configuration and using the evolution rules we get a computation. We consider a computation complete when it halts, no further rules can be applied. A new class of P System called Kernel P System (KP System)has been introduced inorder to generalise various features like the structure of the model, the type of the rules and the execution strategy. Also we define Contextual KP System in which rules consists of attaching context to strings depending upon a choice mapping.In this paper we define contextual KP System with matrix grammars and programmed grammars.In the matrix grammar, instead of a single production one is given a finite sequence of productions. Productions can not be applied separately. A whole sequence always has to be applied.The generative power of context free matrix grammars is remarkably larger than that of context free grammars. Programmed languages are based on a similar method of generating derivations. In a programmed grammar G one is given together with each production f two subsets $\sigma(f)$ and $\phi(f)$ of the entire production set $f(G)$,reffered to as the success and failure field of f, respectively

2 Prerequisites

Contextual grammars were introduced by S.Marcus as a non- chomskian model to describe natural languages. They provide an important tool in the study of formal language theory. Contextual grammars also play a major role in our understanding of grammars without the use of non terminals, called pure grammars.

In contextual grammars, the string xwy is derived by attaching the context (x,y) to the string w.

Definition 2.1. *A contextual grammar with choice is a construct.*

$$G = (V, A, C, \psi),$$

Where V is an alphabet, A is a finite language over V, C is a finite subset of $V^* \times V^*$ *and* ψ: $V^* \to 2^C$. *The strings in A are called* axioms, the elements (u,v) in C are called contexts and ψ is the choice mapping

Definition 2.2. An external contextual grammar is a contextual grammar where all derivations are based on the external mode " ex " that is, $x \Rightarrow_{ex} y$ if and only if $y = uxv$, for a context (u,v) in $\psi(x)$

Definition 2.3. An internal contextual grammar is a contextual grammar where all derivations are based on the internal mode " in " that is, $x \Rightarrow_{in} y$ if and only if $x = x_1x_2x_3,\ y = x_1ux_2vx_3$, for any $x_1, x_2, x_3, \in V^*, (u, v) \in \psi(x_2)$.

Definition 2.4. A total contextual grammar is a system

$$G = (V, A, C, \psi),$$

where V is an alphabet, A is a finite language over V, C is a finite subset of $V^* \times V^*$ *and* $\psi : V^* \times V^* \times V^* \to 2^c$.

Definition 2.5. For a total contextual grammar G = (V,B,C,ψ) *we define the relation* $\Rightarrow_G$ *on*V^* as follows : $x \Rightarrow_G \ y \iff \ x = x_1x_2x_3, y = x_1ux_2vx_3$ for some $x_1, x_2, x_3 \in V^*$,$(u, v) \in C$ such that $(u, v) \in \psi(x_1, x_2, x_3)$

Definition 2.6. A contextual grammar G = (V,B,C,ψ) *is said to be without choice if* $\psi(x) = C$ for all x in V*
Five basic families of languages are obtained, they are

1. *TC = the family of languages generated by total contextual grammars.*
2. *ECC = the family of languages generated by external contextual grammars.*
3. *ICC = the family of languages generated by internal contextual grammars.*

4. EC = the family of languages generated by external contextual grammars without choice.

5. IC = the family of languages generated by internal contextual grammars without choice.

Definition 2.7. Let F be a given family of languages. A contextual Grammar with F selection is a grammar $G = (V, A, (S_1, C_1), ..., (S_n, C_n))$ with $S_i \in F$, for all $1 \leq i \leq n$. We denote by $ICC(F), ECC(F)$ the families of languages generated in the internal and in the external mode, respectively, by contextual grammars with F selection.Here F will be one of the families FIN,REG,CF,CS,RE with emphasis on $F \in \{FIN, REG\}$

Definition 2.8. A Matrix Contextual grammar is a system $G = (V, A, C, \psi, M)$, Where (V, A, C, ψ) is a usual contextual grammar and M is a finite set of sequences of the form $[(\omega, (u_1, v_1), ..., (u_n, v_n)], n \geq= 0$ and $[(u_1, v_1), ..., (u_n, v_n)], n \geq= 1$ where $\omega \in A$ and $(u_i, v_i) \in c, 1 \leq i \leq n$. such a sequence is called a matrix. A derivation step in a matrix contextual grammar G consists of using all the contexts in a matrix , in the order of their appearance in the matrix , for matrices starting with an axiom , this axiom is the derived string. A modular representation of a matrix contextual grammar is written in the form $G = (V, A, (s_1, c_1)..., (s_n, c_n), M)$ with M and derivation derived as above. We denote the family of languages generated by matrix contextual grammars corresponding to a usual family X by MX ; in this way we obtain the families MEC, MIC, MECC(REG), MICC(FIN) etc

Definition 2.9.

In a matrix grammar after using a matrix any other one can be used. While in programmed grammar all pairs of consecutively used context are prescribed.The family of languages generated by programmed contextual language corresponding to a usual family X is denoted by PX A Programmed contextual grammar is a system $G = (V, A, C, \psi, V)$ where $(V, A, C, \psi,)$ is a usual contextual grammar (denoted by G^1) and V is a set of pairs of the form

$[w,(u_1,v_1)]$,$[(u_1,v_1),(u_2,v_2)]$ where $\omega \in A$ and $(u,v),(u_1,v_1),(u_2,v_2)$ are context from C.The pairs in V define a graph over the set A U C, controlling the sequencing of the context used in derivations.In particular having defined as usual the relations $\Rightarrow_\alpha, \alpha \in ex, in$ for a derivation $\delta : \omega \Rightarrow_\alpha w_1 \Rightarrow_\alpha ... \Rightarrow_\alpha, \omega_m$ in G^1 using the axiom $\omega \in A$ and the context $(u_1,v_1),...(u_m,v_m)$ (the step $\omega_{i-1} \Rightarrow_\alpha \omega_i$ is performed using the context (u_i,v_i)) we say that $q(\delta) = \omega_1,(u_1,v_1),...(u_m,v_m)$ is the control sequence associated with δ.Then the language generated by G in the mode α is defined by $L_\alpha(G) = \{x \in v^*/$ there is a derivation $\delta : \omega \Rightarrow^*_\alpha$ in G^1 with $\omega \in A$ and $q(\delta) = \omega,(u_1,v_1)...(u_m,v_m)$ such that $(\omega,(u_1,v_1)) \in v, [(u_i,v_i),(u_{i+1},v_{i+1})] \in U, 1 \leq i \leq m-1$. There are some significant difference between the matrix and the programmed grammar. In a programmed grammar every intial part of the derivation produces a string in the generated language, while a derivation in a ,matrix grammar must be completed.

In a matrix grammar after using a matrix any other one can be used. While in programmed grammar all pairs of consecutively used context are prescribed.The family of languages generated by programmed contextual language corresponding to a usual family X is denoted by PX

3 Kernel P System (KP Systems)

A KP System is a formal model that uses some well known features of existing P System and includes some new elements and more importantly, it offers a coherent view on integrating them in to the same formalism. The system was introduced by M Gheorghe etal. Here a broad range of strategies to ues the rule against the multiset of objects available in each compartments is provided. Now we will see the definition of comparments and KP System

Definition 3.1. *Given a finite set A called alphabet of elements, called objects and a finite set L , of elements called labels , a compartment is a tuple* $C = (l, \omega_0, R^\sigma)$ *where* $l \in L$ *is the label of the comparments* ω_0 *is the intial* *multiset over A and R^σ denotes the DNA code of C, which comprises the set of rules, denoted R, applied in this comparments and a regular expression $\sigma, overLab(R)$ the labels of the rule of R*

Definition 3.2. *A kernel P System of degree n is a tuple*
$K\Pi = (A, L, I0, \mu, C_1, C_2, ..., C_n, i_0)$
where A and L are as in definition 3.1, the alphabet and the set of labels respectively; $I0$ is a multisetb of objects from A, called environment; μ defines the membrane structure which is a graph (V, E), where V are vertices,
$V \subseteq L$ (the nodes are labels of these comparments) , and E edges, $C_1, C_2, ..., C_n$ are n compartments of the system - the inner part of each compartment is called region , which is delimited by a membrane , the labels of the comparments are from L and initial multiset are over A . i_0 is the output compartments where the result is obtained .

4 Contextual KP Systems

Definition *A contextual KP system is a construct*

$$K\Pi = (A, \mu, C_1, C_2, C_3..., C_n, i_0)$$

where A is a finite set of elements called objects ,
μ is a membrane structure, $C_1, C_2, ..., C_n$ are n compartments with

$$C_i = (t_i, w_i)$$

$$t_i = (R_i, \sigma_i)$$

R_i is a contextual rule of the form $(x, (u, v), tar)$ (attaching evolution rules) where $x, u, v \in A$ and $tar \in \{here, in, out\}$ and σ_i is an execution strategy in KP System, i_0 is the output compartment where the result is obtained

The family of all languages generated by contextual KP Systems of degree $n, n \geq 1$ in the mode $X \in \{IC, ICC, EC, ECC\}$ with attaching evolution rules and by using the target indications of the form $\{here, out, in\}$ *is denoted by* KCP (X, n)

5 Contextual KP Systems with Matrix Grammars

A contextual KP System with matrix grammars is a construct $KCP_m = (A, \mu, C_1, C_2, ..., C_n, i_0)$ where A is called alphabet of elements μ is a membrane structure $C_1, C_2, ..., C_n$ are n compartments with $C_i = (t_i, w_i)$, $t_i = (R_i, \sigma_i)$

R_i is contextual rule of the form (x, M, tar) (attaching evolution rule) where $x \in A$ and M is a matrix of contexts in the present comparment and the context are used as in the same way as the usual matrix grammars and $tar \in \{here, out, in\}$. $L_m(KCP)$ denote the language generated in the contextual kernel P System using the matrix grammars.

Theorem 5.1. $MECC(CF) - ECC(CF) \neq \phi$

Proof. Consider a contextual KP System with matrix grammar

$$KCP_m = (A, \mu, C_1, i_0)$$

where

$$A = \{a, b, c, d\}$$

$$\mu = [_1]_1$$

$$C_1 = (t_1, \omega_1)$$

$$\omega_1 = \{\lambda\}$$

$$t_1 = (R_1, \sigma_1)$$

$$i_0 = environment$$

$$R_1 = \{r_1 : (\lambda, [(\lambda, a), (\lambda, b), (\lambda, c), here\}$$

$$r_2 : (a^n b^n c^n, [(\lambda, d), (d, \lambda)], out\}$$

$$\sigma_1 = r_1^* r_2$$

The language generated is

$$L_m(KCP) = \{da^n b^n c^n d / n \geq 1\}$$

Clearly this language canot be generated by the external contextual grammars with context free selection (by Marcus). Hence

$$MECC(CF) - ECC(CF) \neq \phi$$

□

Theorem 5.2.

$$MIC - TC \neq \phi$$

Consider a contextual KP System with matrx grammar

$$kcp_m = (A, \mu, c_1, i_0)$$

$$A = \{a, b, c, d, e\}$$

$$\mu = [_1]_1$$

$$c_1 = (t_1, \omega_1)$$

$$\omega_1 = \{\lambda\}$$

$$t_1 = (R_1, \sigma_1)$$

$$i_0 = environment$$

$$R_1 = \{r_1 : (\lambda, [(\lambda, a), (\lambda, b), (\lambda, c), (\lambda, d), (\lambda, e)], here)\}$$

$$\sigma_1 = r_1^*$$

Every derivation has to use all the context in the matrix format without choice consequently

$$L_m(KCP) = \{x \in \{a, b, c, d, e\}^*/$$

$$|x|_a = |x|_b = |x|_c = |x|_d = |x|_e\}$$

This language does not have the IBS property (no context can be removed from a string $a^n b^n c^n d^n e^n$, with arbitrarly large n), so it is not in the family TC Hence

$$MIC - TC \neq \phi$$

6 Contextual KP Systems with Programmed Grammars

A contextual KP System with Programmed Grammars is a construct $KCP_{pg} = (A, \mu, C_1, C_2, ..., C_n, u, i_0)$ where A is called alphabet of elements μ is a membrane structure $C_1, C_2, ..., C_n$ are n compartments with $C_i = (t_i, w_i)$ $t_i = (R_i, \sigma_i)$, ω_i is the initial multiset present in the i^{th} comparment.

R_i is contextual rule of the form (x, p, tar) (attaching evolution rule) where $x \in A$ and P is a Programmed Grammar of contexts as explained in definition 2.9 and

$$tar \in \{here, out, in\}$$

U is the restriction imposed by the Programmed Grammer i_0 is the output compartment where the result is obtained.$L_{Pg}(KCP)$ denote the language generated in the contextual KPSystem using the programmed grammars..

Theorem 6.1. *$PEC - EC \neq \phi$*

Proof. Consider a contextual KP System with programmed grammar

$$KCP_{pg} = (A, \mu, C_1, C_2, u, i_0)$$

where

$$A = \{a, b, \}$$

$$\mu = [_1[_2]_2]_1$$

$$C_1 = (t_1, \omega_1)$$

$$C_2 = (t_2, \omega_2)$$

$$\omega_1 = \{a\}$$

$$\omega_2 = \{b\}$$

$$t_1 = (R_1, \sigma_1)$$

$$t_2 = (R_2, \sigma_2)$$

$$U = \{[a, (\lambda, a)], [(\lambda, a), (\lambda, a)], [b, (\lambda, b)], [(\lambda, b), (\lambda, b)]\}$$

$$i_0 = c_2$$

$$R_1 = \{r_1 : (a^1, [(\lambda, a), here), i \geq 1$$

$$r_2 : (a^{n-1}, (\lambda, a), out)\}$$

$$R_2 = \{r_1 : (b, [(\lambda, b), here)\}$$

$$\sigma_2 = r_1^*$$

The language generated by the external contextual KPsystem with programmed grammar is

$$L_{pg}(KCP) = \{a^+ U b^+\}$$

Clearly this language canot be generated by the external contextual grammars without choice (by Marcus). Hence

$$PEC - EC \neq \phi$$

□

Theorem 6.2. $PIC - MICC \neq \phi$

Proof. Consider a contextual KP System with programmed grammar

$$KCP_{pg} = (A, \mu, C_1, C_2, u, i_0)$$

where

$$A = \{a, b, \}$$

$$\mu = [_1[_2]_2]_1$$

$$C_1 = (t_1, \omega_1)$$

$$C_2 = (t_2, \omega_2)$$

$$\omega_1 = \{a\}$$

$$\omega_2 = \{b\}$$

$$t_1 = (R_1, \sigma_1)$$

$$t_2 = (R_2, \sigma_2)$$

$$U = \{[a, (\lambda, a)], [(ab, (a, b)]\}$$

$$R_1 = \{r_1 : (a^i, [(\lambda, a), here), i \geq 1$$

$$r_2 : (a^{n-1}, (\lambda, a), out)\}$$

$$\sigma_1 = r_1^* r_2$$

$$R_2 = \{r_1 : (ab, [(a, b), here)\}$$

$$\sigma_2 = r_1^*$$

$$i_0 = c_2$$

The language generated by the internal contextual mode by programmed grammar is

$$L_{pg}(KCP) = \{a^+ U\{a^n b^n / n \geq 1\}$$

Clearly this language is not in MICC (by lemma 6.1 Marcus). Hence

$$PIC - MICC \neq \phi$$

□

7 Conclusion

In this paper we used the contextual way of processing string objects with matrix grammar and programmed grammar.The power of matrix internal contextual grammar is significantly greater than that of usual grammars.There are Some essential differences between the marix and the programmed control of derivations. In a programmed grammar every initial part of derivation produces a string in the generated language while the derivation in a matrix must be completed..

References

[1] Ajitha P S and Desanamabika K S,Contextual KP Systems with minimal/maximal use of selectors,International Jounrnal Of Computer Trends and Techonologies,Vol 28,issue 1, October 2015

[2] Ajitha P S and Desanamabika K S,Contextual KP Systems,International Jounrnal Of Mathematics Trends and Techonologies,Vol 17,issue 1 January 2015

[3] Gh. Paun, Marcus Contextual Grammars ,Kluwer Academic publishers ,1997

[4] Gheorghe Păun, *Membrane Computing An Introduction*, Springer–Verlag, Berlin, 2002.

[5] Madhu Mutyam, Kamala Krithivasan, Contextual P Systems, Fundamenta Informaticae, 2002

[6] Marian Gheorghe, Florentin Ipate, Ciprian Dragomir, Laurentiu Mierla, LuisValencia-Cabrera, Manuel Garcia-Quismondo,and Mario J, Kernel P System - Version 1, International Confrerence On Membrane Computing, 2013