Malaya Journal of Matematik, Vol. 8, No. 2, 445-449, 2020
https://doi.org/10.26637/MJM0802/0019

Regular string-token Petri nets

D. K. Shirley Gloria[1]*, S. Devi[2] and K. Nirmala[3]

Abstract
String-Token Petri Net can generate Regular Language has been proved here. Also, it has been proved that Regular Language originated by String-Token Petri Net are closed with respect to union and concatenation.

Keywords
String-Token Petri Net (σ), Regular Language (μ), Regular Grammar (ρ), production rule (PR).

AMS Subject Classification
68Q45.

[1] *Department of Mathematics, Dr. Ambedkar Government Arts College, Chennai -600039, India.*
[2] *Department of Mathematics, Government Arts And Science College, Chennai - 600131, India*
[3] *Department of Mathematics, SRM Institute of Science and Technology, Kattankulathur -603203, India.*
***Corresponding author**: [1] shirleygloria1976@gmail.com; [2] devi.snowdrop@gmail.com; [3]gk.nimi@gmail.com
Article History: Received **16** November **2019**; Accepted **22** March **2020**

Contents

1. Introduction

Carl Adam Petri's dissertation which was submitted in the year 1962 had brought the notion of Petri Net's [7].

One of the modifications made on Petri net is colored Petri nets. Later on, in the year 2004, σ was introduced ([5] and [1]). Using σ, its properties were studied ([2] and [3]). One of its application was derived [4] .

In this paper, we introduce a new class of σ, called regular σ. Its properties like union, concatenation are obtained.

2. Basic definitions, examples and a theorem of σ

Definition 2.1. *ρ definition can be seen from [6].*

Definition 2.2. *For every μ, $\exists$ a ρ $\ni L = L(G)$. Converse is also true [6].*

Definition 2.3. *Definition of evolution regulations can be seen from [3] and [2].*

Definition 2.4. *σ definition is given in [3].*

Definition 2.5. *Behaviour of transitions of σ can also be seen from [3].*

Example 2.6. *A σ originating μ '$L(N_1)$' is exhibited in figure 1 where $L(N_1) = \{(ab)^n a : n \geq 0\}$ is a μ.*

In figure 1, $N_1 = (P_1, T_1, V_1, F_1, R_1(t), M_1)$ where $P_1 = \{p_1, p_2\}$, $T_1 = \{t_1, t_2\}$, $V_1 = \{S_1, a, b\}$, $R_1(t) = \{t_1 : S_1 \to abS_1, t_2 : S_1 \to a\}$, $M_1 = (S_1, \in)$. After a sequence of firing of transitions of N_1, we obtain a language $L(N_1) = \{(ab)^n a : n \geq 0\}$.

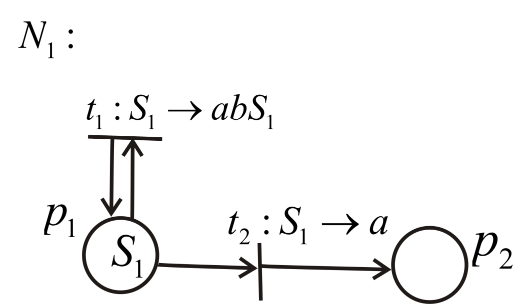

Figure 1

Example 2.7. *A σ originating the μ '$L(N_2)$' is exhibited in figure 2 where $L(N_2) = \{a(ab)^n : n \geq 1\}$ is a μ.*

In figure 2, $N_2 = (P_2, T_2, V_2, F_2, R_2(t), M_2)$ where $P_2 = \{p_3, p_4, p_5\}$, with $T_2 = \{t_3, t_4, t_5\}$, $V_2 = \{S_2, S_3, a, b\}$, $R_2(t) = \{t_3 : S_2 \to S_3ab, t_4 : S_3 \to S_3ab, t_5 : S_3 \to a\}$, $M_2 = (S_2, \in, \in)$. After a sequence of firing of transitions of N_2, we obtain a language $L_2 = \{a(ab)^n : n \geq 1\}$.

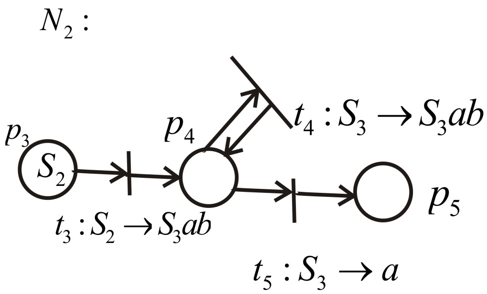

Figure 2

Theorem 2.8. *If L is a μ, then $\exists$ a σ 'N' $\ni L = L(N)$.*

Proof. Let L be a μ originated by a ρ, $G = (V,T,S,P)$ with PR's of the form $A \to xB$, $A \to x$,$A \to yD$, $D \to zD$, $B \to x$, $D \to y$ where $x,y,z \in T^*$ and $A,B,D \in V$.

Erect a σ, $N = (P_1,T_1,V_1,F_1,R_1(t),M_1)$ as follows: Let $V_1 = V \cup T$ be a finite set of alphabets. Let T_1 be a finite set of transitions and each $t_i \in T_1$ be a tag of the PR's of P.

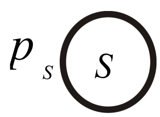

Figure 3

Categorize the PR's of P as

1. T-regulations (calling it as terminating regulations).
2. NT-regulations (calling it as non- terminating regulations).

Name the PR's of the form $A \to xB$, $A \to yD$, $D \to zD$ as NT-regulations and all other regulations as T-regulations. Among NT-regulations, regulations like $D \to zD$ will have a loop structure as it can be applied any number of times. Name these kind of NT-regulations as LNT-regulations. Name other regulations which will not give rise to loop structure as $WLNT$-regulations. Here,regulations like $A \to xB$,$A \to yD$ will not generate loops. All T-regulations will lead to terminate (T-regulations like $B \to x$, $D \to y$).

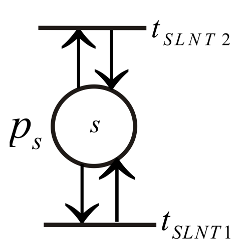

Figure 4

Since S is the beginning character of G, erect a place with S as a token in it and name this place as p_S (see fig 3). Now, group all S - PR's of P (Here,$A \to xB$,$A \to yD$,$A \to x$ are known as A - PR's of P). Among all these S -PR's, group all NT-regulations. These NT-regulations of S- PR's are known as SNT-regulations and other regulations of S- PR's as ST-regulations. Let t_{SLNT1} be the tag of the first $SLNT$-regulation. It's input and output place be p_S. t_{SLNT1} can fire any number of times (see figure 4).(If S- PR's of P have no LNT-regulations, then avoid this erection). If there is another $SLNT$-regulation, then keeping p_S as the input and output place, attach t_{SLNT2} (see figure 4) (if there is no second $SLNT$-regulation, then avoid this erection). Suppose there are more than two $SLNT$-regulations, a similar way of erection can be done (that is, for all $SLNT$-regulations, keep p_S as the input and output place for the corresponding transitions). Presume for simpleness, there are only two $SLNT$-regulations.

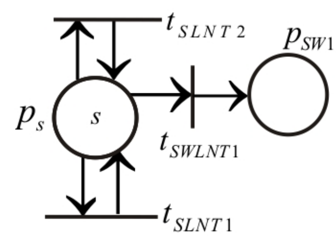

Figure 5

Let t_{SWLNT1} be the tag of the first $SWLNT$-regulation. Then its input place be p_S and let p_{SW1} be the output place of t_{SWLNT1} (see fig 5). Similarly, let t_{SWLNT2} be the tag of the second $SWLNT$-regulation. Then its input place be p_S and let p_{SW2} be the output place of t_{SWLNT2} (see fig 6). Likewise, for the remaining $SWLNT$-regulation, erect places like p_{SW3}, p_{SW4}, ... with input transitions t_{SWLNT3}, t_{SWLNT4}, ...If there are no second, third,... $SWLNT$-regulations, avoid this erection. Presume for simpleness, there are only two $SWLNT$-regulations. Now, group all ST-regulations. Let t_{ST1} be the tag

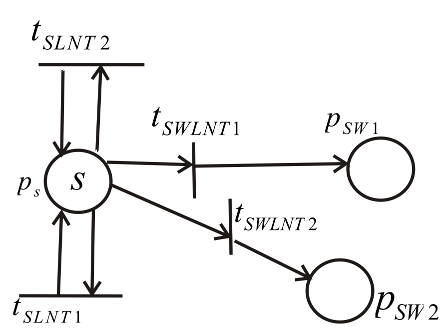

Figure 6

of first ST-regulation. Its input place be p_S and denote its output place as p_{ST1}. If there is a second ST-regulation, let t_{ST2} be the tag of second ST-regulation. Its input place be p_S and name its output place as p_{ST2} (See fig.**??**). Similar erection can be done for any number of ST-regulations. For simpleness , We presume that there are only two ST-regulations. If there is no ST-regulations, then avoid this erection. Since S is a beginning non-terminal in G, there will be atleast one terminal regulation in G. Suppose there are two ST-regulations, then

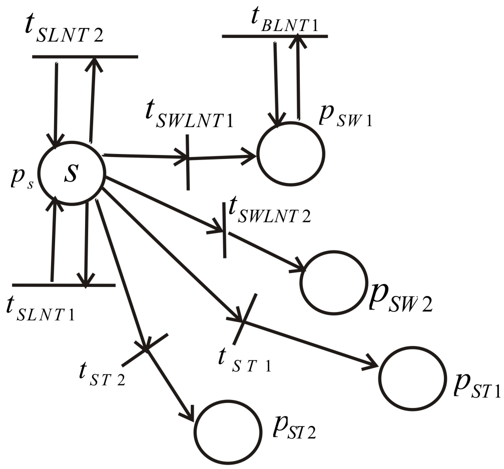

Figure 7. BLNT regulation

p_{ST1}, p_{ST2} would have terminal strings, so name them as final places.

Now, find the leftmost non-terminal that appears in the string on p_{SW1}. Let it be B. Now, group all B- PR's of P. Since G is a ρ, there will be atleast one B- PR. Among all these B-PR's, group all NT-regulations. If there are LNT-regulations in B- PR's, erect t_{BLNT1},t_{BLNT2},.. with p_{SW1} as its input and output place. For simpleness, presume that there is only one $BLNT$-regulation. (If there is no $BLNT$-regulation, avoid this erection) (see fig.7).

If there are $WLNT$-regulations in B- PR's, erect transitions t_{BWLNT1}, t_{BWLNT2}, t_{BWLNT3},... with p_{SW1} as its input place and p_{BW1},p_{BW2},p_{BW3},... as their respective output places (Similar to $SWLNT$-regulations). For simpleness, presume that there is only one $BWLNT$-regulation (If there is no $BWLNT$-regulation, avoid this erection) (see fig.8). Now, group all

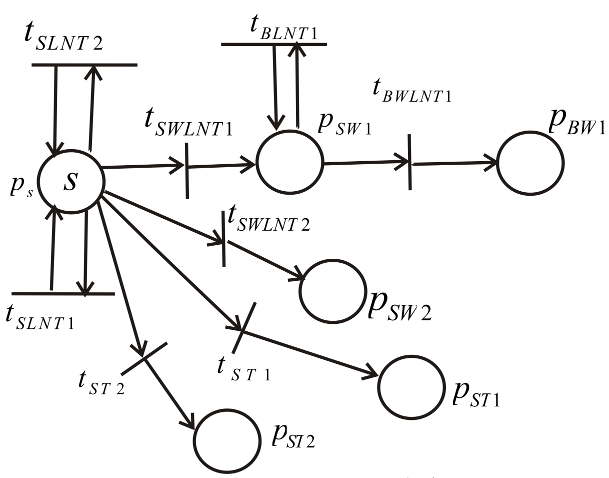

Figure 8. BWLNT regulation

BT-regulations. ie, T-regulations in B- PR's. Let t_{BT1},t_{BT2},... be the tags of the BT-regulations. Their input place would be p_{SW1} and denote its output places as p_{BT1},p_{BT2},... For simpleness, presume that there is only one BT-regulation (If there is

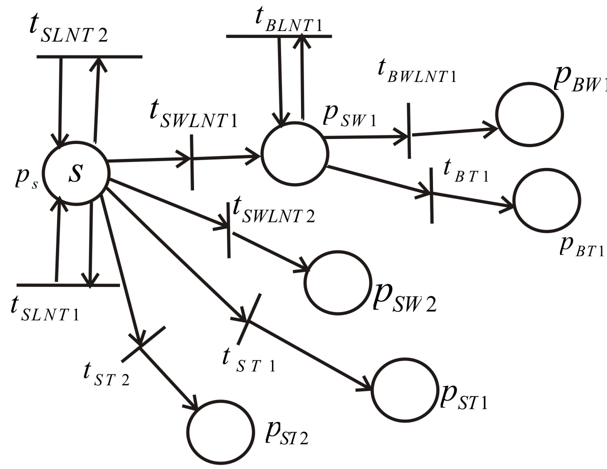

Figure 9. BT regulation

no BT-regulation, avoid this erection) (see fig 9). Similarly, find the leftmost non-terminal that appears in the string on p_{BW1},p_{BW2},... and then do the NT-regulation erection and T-regulation erection as we did above. Here, p_{ST1},p_{ST2},p_{BT1},... are final places as it would have terminal strings. Group the places we erected so far and denote it as P_1. Also, group the transitions we erected so far and denote it as T_1.

Hence, for any given μ, we can erect a corresponding σ. ie, If L is a μ then $L = L(N)$.

□

Example 2.9. *Consider example 2.6. Let us see the erection of N_1. RG for $L(N_1)$ is $G = (\{S_1\}, \{a,b\}, S_1, P)$ where $P = \{S_1 \to abS_1, S_1 \to a\}$. Among the PR's of P, $S_1 \to a$ is a T-regulation and $S_1 \to abS_1$ is a NT-regulation. Let $V_1 = \{S_1\} \cup \{a,b\} = \{S_1, a, b\}$. Since S_1 is the beginning character, erect a place p_S with S_1 as the token (see figure 10).*

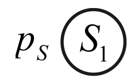

Figure 10

Now, group NT-regulation of S_1. We have only one NT-regulation. $S_1 \to abS_1$ is the NT-regulation of S_1. It is LNT-regulation, since it originates a loop. So, name it as S_1LNT-regulation. Let t_{S_1LNT1} be the tag of S_1LNT-regulation. Its input and output place is p_S (see figure 11). Now, group T-regulation of S_1. There is only one T-regulation of S_1, namely $S_1 \to a$. Name it as S_1T-regulation and let t_{S_1T1} be the tag of

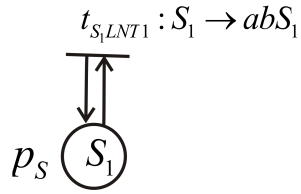

Figure 11

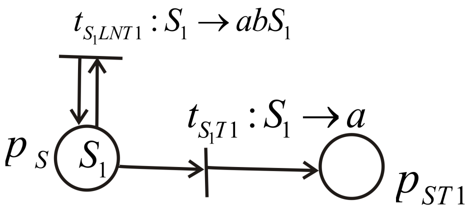

Figure 12

this S_1T-regulation. Its input place is p_S and let p_{ST1} be its output place (see figure 12). No more PR of P is left. Now, group all the places erected so far and name it as P_1. Also, group all the transition erected so far and name it as T_1.

Thus, N_1 is erected.

3. Closure Properties

In this section, results on closure properties like union, concatenation are derived.

Theorem 3.1. *The clan of regular σ language is closed under union.*

Proof. Let $N_1 = (P_1, T_1, V_1, F_1, R_1(t), M_1)$ be a σ originates a μ 'L_1' and $N_2 = (P_2, T_2, V_2, F_2, R_2(t), M_2)$ be a σ originates a μ 'L_2'. Now, a σ, $N = (P_1 \cup P_2 \cup \{p\}, T_1 \cup T_1 \cup \{t_\alpha, t_\beta\}, V_1 \cup V_2 \cup \{S\}$, $F_1 \cup F_2 \cup \{$ arcs from p to t_α, p to t_β, t_α to initial state of N_1, t_β to initial state of $N_2\}$, $R_1(t) \cup R_2(t) \cup \{t_\alpha : S \to S_1, t_\beta : S \to S_2\}, M_3)$ can be erected to originate $L_1 \cup L_2$. In this erection, remove the initial tokens from L_1 and L_2, put S on the place p and connect p and N_1 by t_α, also connect p and N_2 by t_β. It is exhibited in figure 13.

The same can be extended to any number of μ. That is, if $L_1, L_2, L_3, ..., L_n$ are μ's, then $L_1 \cup L_2 \cup ... \cup L_n$ is also a μ (erection of σ for $L_1 \cup L_2 \cup ... \cup L_n$ is similar to that of N in figure 13).

Thus, it is yielded that the clan of regular σ Languages is closed under union. □

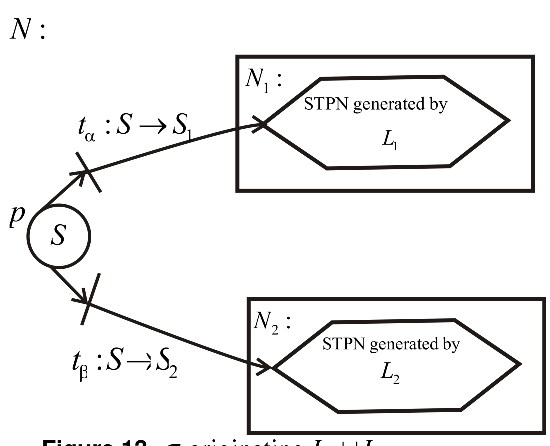

Figure 13. σ originating $L_1 \cup L_2$

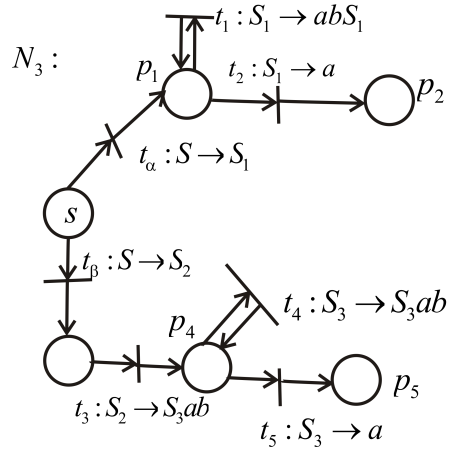

Figure 14

Example 3.2. *A σ originating the μ 'L_1' is exhibited in figure 1 where $L_1 = \{(ab)^n a : n \geq 0\}$. A σ originating μ 'L_2' is exhibited in figure 2 where $L_2 = \{a(ab)^n : n \geq 1\}$. From figure 14, it can be seen that a σ 'N_3' originates $L_1 \cup L_2$.*

Hence, it is concluded that $L(N_3) = L(N_1) \cup L(N_2)$. Now, it has been verified that μ '$L(N_3)$' which is originated by σ, N_1 and N_2 is closed under union. That is, σ 'N_3' originates $L_1 \cup L_2$.

Theorem 3.3. *The clan of regular σ Languages is closed under concatenation.*

Proof. Let $N_1 = (P_1, T_1, V_1, F_1, R_1(t), M_1)$ be a σ originates a μ 'L_1' and $N_2 = (P_2, T_2, V_2, F_2, R_2(t), M_2)$ be a σ originates a μ 'L_2'. Now, a σ, $N = (P_1 \cup P_2, T_1 \cup T_2 \cup \{t_\alpha\}, V_1 \cup V_2 \cup \{\in\}$, $F_1 \cup F_2 \cup \{$ a place with terminal string of N_1 to t_α, t_α to the initial place of $N_2\}$, $R_1(t) \cup R_2(t) \cup \{t_\alpha : \in \to S_2\}, M_4)$ can be erected to generate $L_2.L_1$. It is exhibited in figure 15.

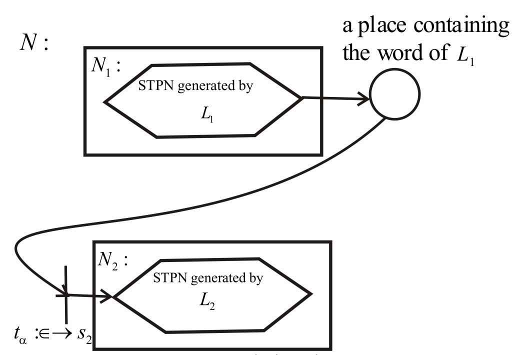

Figure 15. σ originating $L_2.L_1$

From figure 15 , it can be seen that on the place with terminal string (say f_1), strings of L_1 will be deposited after all the sequence of firing of transitions of N_1. So, L_1 can be obtained as a string that is deposited on f_1. Connect f_1 and initial place of N_2 by a new transition $t_\alpha : \in \longrightarrow S_2$. Remove initial token fromN_2. When t_α fires, S_2L_1will be deposited on the initial place of N_2. After all the sequence of firing of transitions of N_2, $L_2.L_1$ will be obtained as a string that is deposited on f_2 (a place with terminal string on N_2). Similarly $L_1.L_2$ is obtained by taking N_2 first and then N_1. Also it can be seen that $L_1.L_2 \neq L_2.L_1$.

The same can be extended to any number of μ. That is, if $L_1, L_2, L_3, ..., L_n$ are μ's, then $L_1.L_2.L_3...L_n$ is also a μ (erection of σ for $L_1.L_2.L_3...L_n$ is similar to that of N in figure 15).

Thus, it is yielded that the clan of regular σ Languages is closed under concatenation.

□

Example 3.4. *A σ originating the μ 'L_1' is exhibited in figure 1 where $L_1 = \{(ab)^n a : n \geq 0\}$. A σ originating μ 'L_2' is exhibited in figure 2 where $L_2 = \{a(ab)^n : n \geq 1\}$. Now,*

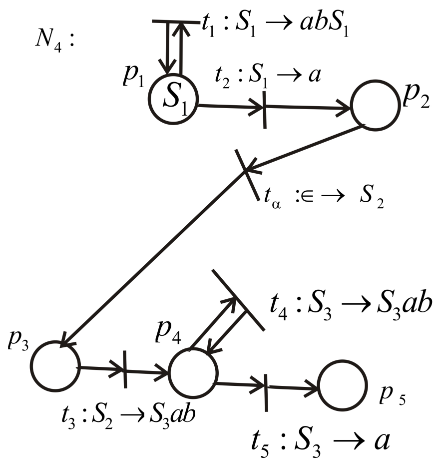

Figure 16

it can be seen from figure 16 that μ which is originated by σ's L_1 and L_2 is closed under concatenation. That is, σ 'N_4' originates $L_2.L_1$. In figure 16, S_2 will be removed from figure 2 and $t_\alpha :\in \longrightarrow S_2$ is taken as a transition between p_2 and p_3.

4. Conclusion

It can be concluded that every μ can be originated by σ. Also regular σ is closed with respect to union and concatenation.

References

[1] D.K. Shirley Gloria, K. Rangarajan, Algorithmic Approach to the Length of the Words of String Token Petri Nets,*CIIT International Journal of Data mining and Knowledge Engineering* 3(6)(2011), 387–390.

[2] S. Devi and D.K. Shirley Gloria, Context-free languages of String Token Petri Nets, *International Journal of Pure and Applied Mathematics,* 113(11)2017, 96–104.

[3] D.K. Shirley Gloria, Beulah Immanuel, K. Rangarajan, Parallel Context-Free String Token Petri Nets, *International Journal of Pure and Applied Mathematics* 59(3)(2010), 275–289.

[4] D.K. Shirley Gloria, Parsing for Languages of String Token Petri Nets, *IEEE Fifth International Conference on Bio-Inspired Computing: Theories arid Applications (BIC-TA 2010)* at Liverpool Hope University, Liverpool, U.K., 2(2010), 1577–1584.

[5] Beulah Immanuel and K. Rangarajan and K.G.Subramanian, String-token Petri nets, In: *Proceedings of the European Conference on Artificial Intelligence, One Day Workshop on Symbolic Networks at Valencia*, In:Spain, 2004.

[6] Peter Linz, *An Introduction to Formal Languages and Automata*, Jones & Bartlett Learning, 2006

[7] James Peterson, *Petrinet Theory and Modeling of Systems*, Prentice Hall, USA, 1997.

⋆⋆⋆⋆⋆⋆⋆⋆⋆
ISSN(P):2319 − 3786
Malaya Journal of Matematik
ISSN(O):2321 − 5666
⋆⋆⋆⋆⋆⋆⋆⋆⋆