Int. J. Math. And Appl., 10(1)(2022), 79–84
ISSN: 2347-1557
Available Online: http://ijmaa.in/

International Journal of Mathematics And its Applications

Regular Multi-Strings Token Petri Net

D. K. Shirley Gloria[1,*] and D. K. Sheena Christy[2]

1 Department of Mathematics, Dr. Ambedkar Government Arts College, Vyasarpadi, Chennai – 600 039, Tamilnadu, India.

2 Department of Mathematics, SRM Institute of Science and Technology, Kattankulathur – 603 203, Tamilnadu, India.

Abstract: Every regular language can be caused by Multi-Strings Token Petri Net.

MSC: 03D05.

Keywords: Multi-Strings Token Petri Net(MSTPN), Regular Language(RL), Regular Grammar(RG).

1. Introduction

The notion of Petri net has begun in Carl Adam Petri's dissertation which was presented in the year 1962 [2]. String-Token Petri Net has its origin in [1]. Context-Free String-Token Petri Net and Parallel Context-Free String Token Petri Net can be found in [3] and [4]. Context-Free Multi-Strings Token Petri Net can be found in [5].

Here, we introduce Regular Multi-Strings Token Petri Net.

2. Basic Definitions

Definition 2.1. *Regular Grammar(RG) and Regular Language(RL) are defined in [6].*

Definition 2.2. *Evolution rules are found in [3].*

Definition 2.3. *A MSTPN is defined in [5].*

Definition 2.4. *Many systems' activity may be characterised in terms of the system's states and modifications. A state or marking in an MSTPN is modified as per the underlying transition (firing) rules in order to imitate the dynamic response of the structure.*

(i). When each input location p of a transition t includes a collection of strings containing left side terms of the transition rules, the transition is said to be activated. For example, $t_i : A \to aB$ / $A \to xA$ / $B \to b$ / $A \to a$, then input location p_j of t_i should contain strings $\{A, B\}$. Suppose input location p_k of t_m consists of $\{aAb, dB, xaB, ab\}$ then ab can be retained for the immediate firing of transition other than insertion rule, when insertion rule is applied, ab will be carried out to the next immediate location.

* E-mail: shirleygloria1976@gmail.com

(ii). An active transition fires.

(iii). Suppose $t : X \rightarrow aY \ / \ A \rightarrow xA \ / \ B \rightarrow b$ *and a collection of strings of input location* p *of* t *is* $\{bX, B, mBYa\}$*, then* t *is activated as the leftmost non-terminal in each of the string is* $\{bX, B, mBYa\}$ *appears in the left side expression of* t*. So, when* t *fires,* $\{bX, B, mBya\}$ *will be removed from the input location of* t *and* $\{baY, b, mbYa\}$ *will be deposited in the output location of* t*.*

(iv). Identity rule like $ab \rightarrow ab$ *can be used in the MSTPN as far as all strings in the corresponding location become terminals.*

Example 2.5. *A Multi-Strings Token Petri Net* $N_1 = (P, T, V, F, R(t), M_0)$ *causing the RL* $L(N_1)$ *is demonstrated in Figure 1, where* $L(N_1) = \{a^n / n \geq 0\}$.

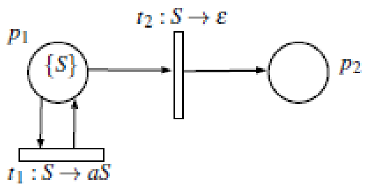

Figure 1. **MSTPN** N_1 **of Example 2.1**

In N_1, $P = \{p_1, p_2\}$, $T = \{t_1, t_2\}$, $V = \{S, a\}$, $F = \{p_1 \rightarrow t_1, t_1 \rightarrow p_1, p_1 \rightarrow t_2, t_2 \rightarrow p_2\}$, $R(t) = \{S \rightarrow aS, S \rightarrow \varepsilon\}$, $M_0 = (\{S\}, \varepsilon)$.

Example 2.6. A Multi-Strings Token Petri Net $N_2 = (P, T, V, F, R(t), M_0)$ causing the RL $L(N_2)$ is demonstrated in Figure 2, where $L(N_2) = \{a^n b^m / n, m \geq 0\}$.

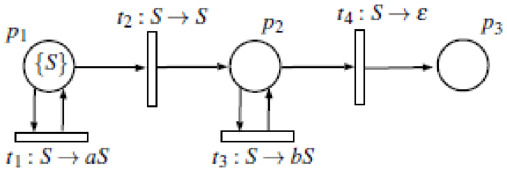

Figure 2. **MSTPN** N_2 **of Example 2.2**

In N_2, $P = \{p_1, p_2, p_3\}$, $T = \{t_1, t_2, t_3, t_4\}$, $V = \{S, a, b\}$, $F = \{p_1 \rightarrow t_1, t_1 \rightarrow p_1, p_1 \rightarrow t_2, t_2 \rightarrow p_2, p_2 \rightarrow t_3, t_3 \rightarrow p_2, p_2 \rightarrow t_4, t_4 \rightarrow p_3\}$, $R(t) = \{S \rightarrow aS, S \rightarrow S, S \rightarrow bS, S \rightarrow \varepsilon\}$, $M_0 = (\{S\}, \varepsilon, \varepsilon)$.

3. Theorem on MSTPN

Theorem 3.1. *For every RL, MSTPN* $'N'$ *can be found such that* $L = L(N)$*.*

Proof. Consider a RL $'L'$ which is caused by RG, $G = (M, \Sigma, P, S)$ with rules P of the form $S \rightarrow aS$, $A \rightarrow aB$, $B \rightarrow bB$, $B \rightarrow b$, $S \rightarrow a$, $A \rightarrow \varepsilon$, where $S, A, B \in M;\ a, b \in \Sigma$.

Build a MSTPN $N = (P_1, T, V, F, R(t), M_0)$ as described below: $V = M \cup \Sigma$ is a finite collection of alphabets, T is a finite collection of transitions and each transition $t_i \in T$ are labelled by rules in P. Segregate the rules of P as

(i). T – rules

(ii). NT rules.

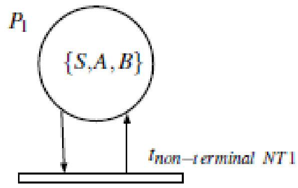

Figure 3(a).

The rules of the form $S \to aS$, $B \to bB$ are called as NT rules since it can be replaced indefinite time for the non-terminal on the left side of the rule appear on the right side of the rule also. Call the remaining rules as T – rules. Few of the T – rules, like $B \to b$, $A \to \varepsilon$, $S \to a$ will terminate where $a, b \in \Sigma$.

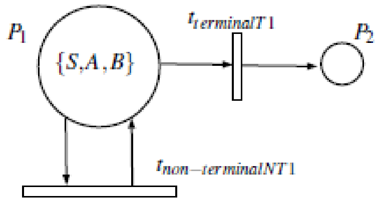

Figure 3(b).

Among all these non-terminals production rules, collect all NT – rules. From production rules of P, we have rules like $S \to aS, B \to bB$. This type of rules are called as non-terminal NT – rules. From other rules, rules like $A \to aB$ we call it as non-terminal T rules. Other rules like $A \to \varepsilon$, $B \to b$, $S \to a$, we call them as terminal T-rules. Let $t_{non-terminal\ NT1}$ be the label of these non-terminal NT production rules. This transition $t_{non-terminal\ NT1}$ can be fired indefinite time (see Figure 3(a)).

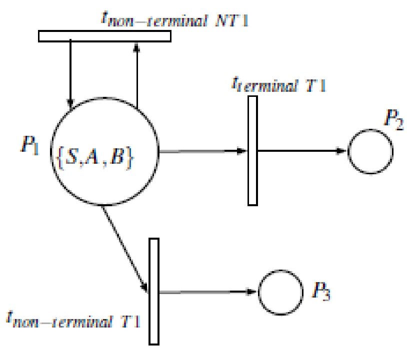

Figure 3(c).

If non-terminal NT rules are not there in P, then omit this construction. Now, collect all terminal T-rules. From P_1, through a transition $t_{terminal\ T1}$, connect P_1 to P_2. Label $t_{terminal\ T1}$ with all these terminal T-rules (see Figure 3(b)).

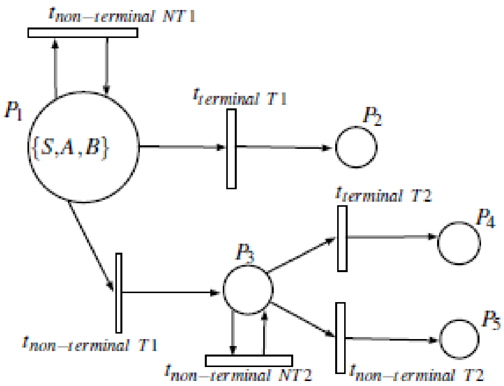

Figure 3(d).

After $t_{terminal\ T1}$ fires, some of the words of G will be deposited in P_2. As S is the beginning symbol of G, at least one terminal T – rule will be there in P. Now, collect all non-terminal T- rules. From P_1, through a transition $t_{non-terminal\ T1}$, connect P_1 to P_3. Label $t_{non-terminal\ T1}$ with all these non-terminal T – rules (see Figure 3(c)). After $t_{non-terminal\ T1}$ fires, some strings of G will be deposited in P_3. Among these strings, look at the left most non-terminal and apply that corresponding rules of P. From P_3, through transition $t_{non-terminal\ NT2}$, connect P_3 to P_3. From P_3, there will be two paths. Connect P_3 to P_4, through a transition $t_{terminal\ T2}$. Also, connect P_3 to P_5, through a transition $t_{non-terminal\ T2}$. If these transitions are in need, it will be built. Otherwise it would terminate (see Figure 3(d)).

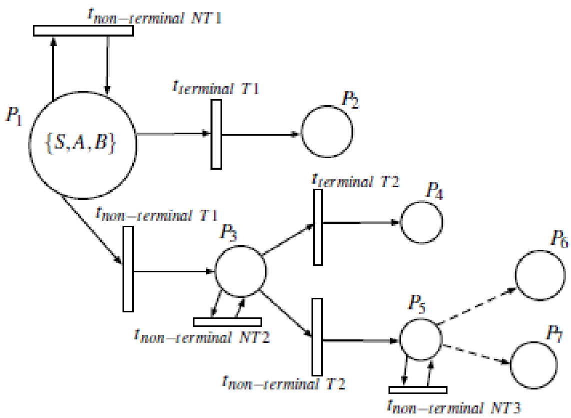

Figure 3(e).

In a similar way, it will be proceeded. In one stage, there will be no-more non-terminals on some locations. In that case, we will obtain all words of G(see Figure 3(e)).

Now, P_1 is the collection of all locations built so far and T is the collection of all transitions built so far. A MSTPN can be built in a similar way, for any given RL. Thus, for any given RL, it is possible to build a corresponding MSTPN $'N'$. Therefore, when L is a RL, $L = L(N)$. □

Example 3.2. *Consider Example 2.5. Let us see the construction of* N_1*. RG for* $L(N_1)$ *is* $G = (\{S\}, \{a, \varepsilon\}, S, P)$*, where* $P = \{S \to aS, S \to \varepsilon\}$.

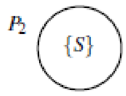

Figure 4(a).

Among these production rules of P, $S \to \varepsilon$ is a terminal T rule. Rules like $S \to aS$ is called as non-terminal NT rules. There are no non-terminal T rules.

Let $N_1 = (P_1, T, V, F, R(t), M_0)$ be the MSTPN, where $V = \{S\} \cup \{a, \varepsilon\} = \{S, a, \varepsilon\}$. As S is the beginning symbol, build a location P_1 with S as token (see Figure 4(a)). Now, collect non-terminal NT rules. Connect P_1 with P_1 through $t_{non-terminal\ NT1}$ rule(see Figure 4(b)).

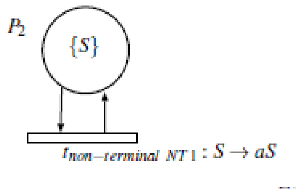

Figure 4(b).

Now, collect all terminal T-rules. Connect P_1 to P_2 with the transition $t_{terminal\ T1}$ rule. Here, $S \to \varepsilon$ is the only terminal T-rule.

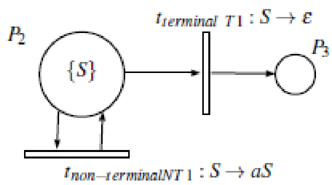

Figure 4(c).

On P_1, only words of G will be deposited. So, no more construction is needed. Figure 4(c) is the same as Figure 1. Collect all the locations constructed so far and call it as P_1. Here, $P_1 = \{P_2, P_3\}$. Also, collect all the transitions constructed so far and call it as T. Here, $T = \{t_{non-terminal\ NT1}, t_{terminal\ T1}\}$. Initial marking is $M_0 = (\{S\}, \varepsilon)$. Thus, N_1 is constructed and $L(N_1) = \{a^n / n \geq 0\}$.

4. Conclusion

Thus, every Multi-Strings Token Petri Net causes a Regular Language.

References

[1] Beulah Immanuel, 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, Spain, (2004).

[2] James Peterson, *Petrinet theory and modeling of systems*, Prentice Hall, USA, (1997).

[3] S. Devi and D. K. Shirley Gloria, *Context-Free Languages Of String Token Petri Nets*, International Journal of Pure and Applied Mathematics, 113(2017), 96–104.

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

[5] D. K. Shirley Gloria and D. K. Sheena Christy, *Multi-Strings Token Petri Nets*, International Conference on Instrumentation, MEMS and Biosensing Technology, (Submitted).

[6] Peter linz, *An Introduction to Formal Languages and Automata*, 5th Edition, Jones & Bartlett Learning, LLC, (2012).