Journal of Analysis and Applications
Vol. 20 (2022), No.2, pp.135-150
ISSN: 0972-5954

URL : www.sasip.net

Construction of α-language from the language of a QDPDA of order "n"

A. Jain, S. Jain and G.C. Petalcorin, Jr.

Abstract. In [10], A. Jain et al. introduced the notion of quasi-deterministic pushdown automata (QDPDA) of order "n'' as a generalization of already known deterministic pushdown automata (DPDA). The authors also introduced there a new family of language viz. α-language of order n as a subclass of context-free language and have shown that given an α-language of order n, there exists an equivalent QDPDA of the same order that accepts exactly the given α-language of order n. In continuation of that work, in this paper, we show that given the language of a QDPDA of order n, there exists an equivalent α-language of order "n''.

AMS Subject Classification (2020): 68T99, 68Q45

Keywords: Nondeterministic pushdown automata, deterministic pushdown automata, context-free grammar

1. Introduction

We know that the nondeterministic pushdown automata (NPDA) is the machine counterpart of context-free languages while deterministic pushdown automata (DPDA) can be associated with a subset of context-free languages, viz. deterministic context-free languages.

Motivated by the idea to generalize DPDA, the authors in [10] introduced the notion of "quasi-deterministic pushdown automata (QDPDA)" of order n that behaves like DPDA for $n = 1$. The authors also introduced the notion of α-grammar and α-language of order n as a subclass of context-free grammar and context-free language respectively and have shown that given an α-language of order n, there exists an equivalent QDPDA of the same order that accepts exactly the given α-language of order n.

In this paper, we construct an α-language of order n from the language of a given QDPDA of order n. The order of the constructed α-language initially depends on both the number of states as well as the order "n" of the QDPDA. But after removing useless productions and useless variables in the newly constructed α-grammar, we get a minimal equivalent α-grammar that generates α-language of order n.

2. Preliminaries

We first informally define quasi-deterministic pushdown automata (QDPDA) of order n as follows:

Definition 2.1 [10]. A "quasi-deterministic pushdown automata(QDPDA) of order $n(n \geq 1)$" is a PDA that can make atmost "n" transitions corresponding to a given input symbol (real or virtual input λ) and stack top symbol from a given state with the condition that when a λ-move is possible from a give state for some stack top symbol, then no input consuming alternative is possible for the same state and stack top symbol configuration.

We now formally define a "quasi-deterministic pushdown automata (QDPDA) of order n" as follows:

Definition 2.2 [10]. A "quasi-deterministic pushdown automata(QDPDA) of order n" is defined by the sep-tuple

$$M(n) = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F),$$

where

1. Q is a finite state of internal states of the control unit,

2. Σ is the input alphabet.

3. Γ is a finite set of symbols called the stack alphabet,

4. $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to$ finite subset of $Q \times \Gamma^*$ of order at most n is the transition function subject to the restriction that for every $q \in Q$ and $B \in \Gamma$, if $\delta(q, \lambda, B)$ is nonempty, then $\delta(q, a, B)$ must be empty for every $a \in \Sigma$.

5. $q_0 \in Q$ is the initial state of the control unit,

6. $Z_0 \in \Gamma$ is the stack start symbol,

7. $F \subseteq Q$ is the set of final states.

Observations [10].

(i) QDPDA (1) $\subseteq QDPDA(2) \subseteq QDPDA(3) \subseteq \cdots\cdots$.

(ii) QDPDA (1) = DPDA.

Theorem 2.3 [10]. *Every QDPDA of order n is equivalent to a QDPDA $M(n) = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ of order n such that if*

$$\delta(q, a, A) = (r, \gamma), \text{ then } |\gamma| \leq 2, \gamma \in V^*, a \in \Sigma, A \in \Gamma.$$

Remark 2.4 [10]. Theorem 2.3 assumes that a QDPDA will never push more than two stack symbols per move. The single move of a QDPDA either increases or decreases the stack length by a single symbol or keeps the stack length intact.

Definition 2.5 [10]. A context-free grammar $G = (V, T, S, P)$ is said to be a "α-grammar of order "n" if all production in P are of the form

$$A \to ax$$

where $a \in T \cup \{\lambda\}$, $x \in V^*$ and any pair (A, a) occurs atmost n times in P with the restriction that if the pair (A, λ) occurs in the production, then no pair of the form (A, b) where $b \in T$ occurs in the production rule. We denote the α-grammar G of order n by $G_\alpha(n)$.

Clearly,

$$G_\alpha(1) \subseteq G_\alpha(2) \subseteq G_\alpha(3) \subseteq \cdots\cdots$$

Definition 2.6 [10]. The language generated by an α-grammar G of order n is defined as the "α-languages of order n" and is denoted as $\alpha_G(n)$. In other words, if $G = (V, T, S, P)$ is an α-grammar of order n then

$$\alpha_G(n) = \{w \mid w \in T^* \text{ and } S \Rightarrow_G^* w\}.$$

3. Construction of α-language from the language of a QDPDA of order n

In this section, we prove the main algorithmic result pertaining to the construction of α-language (or α-grammar) from the language of a QDPDA of order n. The result is further illustrated with the help of an example.

Theorem 3.1. *If L is $L(M(n))$ for some QDPDA $M(n) = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \phi)$ of order n, then L is an α-language of order n.*

Proof. The Construction of the associated α-grammar is based on the fact that contents of the stack are reflected in the variable part of the sentential form while the processed input is the terminal prefix of the sentential form. Without any loss of generality, we may assume that

(1) The QDPDA $M(n)$ accepts an input string iff the stack is empty after processing the string.

(2) With $a \in \Sigma \cup \{\lambda\}$ and $A \in \Gamma$, all transitions must have the form

$$\delta(q, a, A) = \{(q_1, \gamma_1), (q_2, \gamma_2), \cdots, (q_k, \gamma_k)\},$$

where $1 \leq k \leq n$ and $\gamma_j \in \Gamma^*$ with $|\gamma_j| \leq 2$ for all $1 \leq j \leq k$. That is, each move either increases or decreases the stack length by a single symbol or keeps the stack length intact.

Now, let $G = (V, \Sigma, P, S)$ be an α-grammar where

$V=$ set of objects of the form $[q, A, p]$, $q, p \in Q$ and $A \in \Gamma$ plus the new symbol S;

P is the set of productions given by

(i) $S \to [q_0, Z_0, q]$ for all $q \in Q$,

(ii) If $\delta(q, a, A)$ contains (q_1, λ), then the production rule is

$$[q, A, q_1] \to a;$$

If $\delta(q, a, A)$ contains (q_1, B_1), then the production rule is

$$[q, A, q_2] \to a(q_1, B_1, q_2);$$

If $\delta(q, a, A)$ contains $[q_1, B_1 B_2]$, then the producion rule is

$$[q, A, q_3] \to a[q_1, B_1, q_2][q_2, B_2, q_3]; B_1, B_2 \in \Gamma,$$

where q_2, q_3 assume all possible values in Q.

Clealry, G is an α-grammar of order $t \leq |Q|^2 n$. But after removing the useless variables and useless productions, we get the minimal equivalent α-grammar of order exactly equal to n.

Note that the above production rules are defined so that

$$[q, A, p] \Rightarrow_G^* w \text{ for } w \in \Sigma^* \text{ iff } w \text{ causes}$$

$M(n)$ to erase A from its stack by some sequence of moves begining in state q and ending in state p. There might be some states r that can not be reached from q while erasing A. In that case, the resulting variables

$[q, A, r]$ are useless symbols and do not affect the language generated by the α-grammar. We shall remove such type of useless variables during minimization of the newly constructed α-grammar G.

Further, the variables that appear in any step of a leftmost derivation in G correspond to the symbols on the stack of $M(n)$ at a time when $M(n)$ has seen as much of the input as the grammar has alreday generated.

Now, in order to show

$$L(G) = L(M(n)),$$

we prove by induction on the number of steps in a derivation of G or number of moves of $M(n)$ that

$$[q, A, p] \Rightarrow_G^* w \text{ for } w \in \Sigma^* \text{ iff } (q, w, A) \vdash_M^* (p, \lambda, \lambda). \tag{1}$$

Firstly, we show by induction on i that if

$$(q, w, A) \vdash_M^i (p, \lambda, \lambda),$$

then

$$[q, A, p] \Rightarrow_G^* w.$$

If $i = 1$ then w is either λ or a single real input symbol and $(p, \lambda) \in \delta(q, w, A)$.

Thus

$$[q, A, p] \to w \text{ is a production of } G.$$

Now suppose $i > 1$. Let $w = au$ where $u \in \Sigma^*$ and

$$(q, au, A) \vdash_M (q_1, u, B_1 B_2 \cdots B_k) \overset{i-1}{\vdash_M} (p, \lambda, \lambda), k = 0, 1, 2 \text{ with } B_0 = \lambda. \tag{2}$$

The string u can be written as

$$u = u_1 u_2 \cdots u_k,$$

where $u_j \in T^*(1 \leq j \leq k)$ has the effect of popping B_j from the stack possibly after a long sequence of moves.

In general , $B_j(1 \leq j \leq k, 0 \leq k \leq 2)$ reamins unchanged on the stack while $u_1, u_2, \cdots, u_{j-1}$ is processed. Also, there exist states $q_2, q_3, \cdots, q_{k+1}$ with $q_{k+1} = p$ such that for all $j = 1$ to k,

$$(q_j, u_j, B_j) \vdash_M^* (q_{j+1}, \lambda, \lambda)$$

in fewer than i moves.

We apply induction hypothesis and get

$$[q_j, B_j, q_{j+1}] \Rightarrow_G^* u_j \text{ for } 1 \leq j \leq k.$$

Recalling the first move in (2) viz.

$$(q, au, A) \vdash_M (q_1, u, B_1 B_2 \cdots B_k),$$

we know by the construction of production rules in G that

$$[q, A, p] \Rightarrow a[q_1, B_1, q_2][q_2, B_2, q_3] \cdots [q_k, B_k, q_{k+1}],$$

with $q_{k+1} = p$.

Thus

$$[q, A, p] \Rightarrow_G^* au_1 u_2 \cdots u_k = w.$$

Conversly supoose that

$$[q, A, p] \Rightarrow_G^* w.$$

We show by induction on i that

$$(q, w, A) \vdash_M^* (p, \lambda, \lambda).$$

For $i = 1, w$ is either λ or a symbol in Σ and $[q, A, p] \rightarrow w$ must be a production of G. Thus in this case, we have

$$(q, w, A) \vdash_M (p, \lambda, \lambda).$$

Now assume $i > 1$.

Let

$$[q, A, p] \Rightarrow a[q_1, B_1, q_2][q_2, B_2, q_3] \cdots [q_k, B_k, q_{k+1}] \Rightarrow^{i-1} w, \tag{3}$$

where $0 \leq k \leq 2$, $q_{k+1} = p$ and $B_0 = \lambda$.

Then we write

$$w = au_1u_2 \cdots u_k,$$

where $u_j \in T^*$ for all $1 \leq j \leq k$ and

$$[q_j, B_j, q_{j+1}] \Rightarrow_G^* u_j \text{ for all } 1 \leq j \leq k,$$

with each derivation taking fewer than i steps.

By the induction hypothesis, we get

$$(q_j, u_j, B_j) \vdash^* (q_{j+1}, \lambda, \lambda) \text{ for all } 1 \leq j \leq k. \tag{4}$$

The sequence of ID's in (4) clearly shows

$$(q_j, u_j, B_jB_{j+1} \cdots B_k) \vdash^* (q_{j+1}, \lambda, B_{j+1} \cdots B_k). \tag{5}$$

From the first step in the derivation of w from $[q, A, p]$ given in (3) viz.

$$[q, A, p] \Rightarrow a[q_1, B_1, q_2][q_2, B_2, q_3] \cdots [q_k, B_k, q_{k+1}],$$

where $q_{k+1} = p$ we know that

$$(q, w, A) = (q, au_1u_2 \cdots u_k, A) \vdash (q_1, u_1u_2 \cdots u_k, B_1B_2 \cdots B_k), \tag{6}$$

where $0 \leq k \leq 2$ is a legal move of $M(n)$.

From (5) and (6), we get

$$(q, w, A) \vdash^* (p, \lambda, \lambda).$$

Finally, on substituting $q = q_0$ and $A = Z_0$ in (1), we obtain

$$[q_o, Z_0, p] \Rightarrow^* w \text{ iff } (q_0, w, Z_0) \vdash^* (p, \lambda, \lambda).$$

The above identity together with production rule (i) of G gives

$$S \overset{*}{\Rightarrow} w \text{ iff } [q_o, w, Z_0] \vdash^* (p, \lambda, \lambda),$$

for some state $p \in Q$.

Thus

$$w \in L(G) \Longleftrightarrow w \in L(M(n)).$$

Now the required minimal α-grammar of orer n can be obtained from the α-grammar G by removing useless variables and useless productions in G. Hence the proof. $\square$

Example 3.2. Consider the QDPDA $M(2)$ of order 2 given by

$$M(2) = (\{q_0, q_1, q_2\}, \{a, b\}, \{X, Z_0, A, B\}, \delta, q_0, Z_0, \{q_2\}),$$

where

$$\begin{aligned}
\delta(q_0, \lambda, Z_0) &= \{(q_1, XZ_0)\}, \\
\delta(q_1, a, X) &= \{(q_1, XA)\}, (q_1, \lambda)\} \\
\delta(q_1, b, A) &= \{(q_1, B)\}, \\
\delta(q_1, b, B) &= \{(q_1, \lambda)\}, \\
\delta(q_1, \lambda, Z_0) &= \{(q_2, \lambda)\}.
\end{aligned}$$

The language accepted by the above QDPDA $M(2)$ is

$$L(M(2)) = \{a^n b^{2n-2} | n \geq 1\}.$$

We obtain the equivalent α-grammar by the algorithm discussed in Theorem 3.1. We note that with q_0 as the initial state and q_2 as the final state, the QDPDA satisfies conditions (1) and (2) assumed in the proof of Theorem 3.1.

We write the production rules of the α-grammar $G = (V, \Sigma, P, S)$ where

$$
\begin{aligned}
V = \{&[q_0, X, q_0], [q_0, X, q_1], [q_0, X, q_2], [q_0, Z_0, q_0], [q_0, Z_0, q_1],\\
&[q_0, Z_0, q_2], [q_0, A, q_0], [q_0, A, q_1], [q_0, A, q_2], [q_0, B, q_0],\\
&[q_0, B, q_1], [q_0, B, q_2], [q_1, X, q_0], [q_1, X, q_1], [q_1, X, q_2], [q_1, Z_0, q_0], [q_1, Z_0, q_1],\\
&[q_1, Z_0, q_2], [q_1, A, q_0], [q_1, A, q_1], [q_1, A, q_2], [q_1, B, q_0],\\
&[q_1, B, q_1], [q_1, B, q_2], [q_2, X, q_0], [q_2, X, q_1], [q_2, X, q_2], [q_2, Z_0, q_0], [q_2, Z_0, q_1],\\
&[q_2, Z_0, q_2], [q_2, A, q_0], [q_2, A, q_1], [q_2, A, q_2], [q_2, B, q_0],\\
&[q_2, B, q_1], [q_2, B, q_2], S\},
\end{aligned}
$$

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

and the production rules in P are given by

$$
\begin{aligned}
S &\rightarrow [q_0, Z_0, q_0]|[q_0, Z_0, q_1]|[q_0, Z_0, q_2];\\
[q_0, Z_0, q_0] &\rightarrow \lambda[q_1, X, q_0][q_0, Z_0, q_0]|\lambda[q_1, X, q_1]\\
&\quad [q_1, Z_0, q_0]|\lambda[q_1, X, q_2][q_2, Z_0, q_0];\\
[q_0, Z_0, q_1] &\rightarrow \lambda[[q_1, X, q_0][q_0, Z_0, q_1]|\lambda[q_1, X, q_1]\\
&\quad [q_1, Z_0, q_1]||\lambda[q_1, X, q_2][q_2, Z_0, q_1],\\
[q_0, Z_0, q_2] &\rightarrow \lambda[[q_1, X, q_0][q_0, Z_0, q_2]|\lambda[q_1, X, q_1]\\
&\quad [q_1, Z_0, q_2]||\lambda[q_1, X, q_2][q_2, Z_0, q_2];\\
[q_1, X_1, q_0] &\rightarrow a[q_1, X, q_0][q_0, A, q_0]|a[q_1, X, q_1]\\
&\quad [q_1, A, q_0]||a[q_1, X, q_2][q_2, A, q_0];\\
[q_1, X, q_1] &\rightarrow a[q_1, X, q_0][q_0, A, q_1]|a[q_1, X, q_1]\\
&\quad [q_1, A, q_1]|a[q_1, X, q_2][q_2, A, q_1];
\end{aligned}
$$

$$\begin{aligned}
[q_1, X, q_2] &\rightarrow a[q_1, X, q_0][q_0, A, q_2]|a[q_1, X, q_1]\\
&\quad [q_1, A, q_2]|a[q_1, X, q_2][q_2, A, q_2];\\
[q_1, X, q_1] &\rightarrow a;\\
[q_1, A, q_0] &\rightarrow b[q_1, B, q_0];\\
[q_1, A, q_1] &\rightarrow b[q_1, B, q_1];\\
[q_1, A, q_2] &\rightarrow b[q_1, B, q_2];\\
[q_1, B, q_1] &\rightarrow b;\\
[q_1, Z_0, q_2] &\rightarrow \lambda.
\end{aligned}$$

The constructed α-grammar G is of order 3.Now, we obtain the equivalent minimal α-grammar G' of order 2 by removing useless variables and useless productions in G as follows:

A variable that does not occur on the leftside of any production must be useless, so we eliminate the productions involving useless variables and have the following minimal α-grammar G':

$$\begin{aligned}
S &\rightarrow [q_0, Z_0, q_2],\\
[q_0, Z_0, q_2] &\rightarrow \lambda[q_1, X_1, q_1][q_1, Z_0, q_2],\\
[q_1, X, q_1] &\rightarrow a[q_1, X, q_1][q_1, A, q_1],\\
[q_1, X, q_1] &\rightarrow a,\\
[q_1, A, q_1] &\rightarrow b[q_1, B, q_1],\\
[q_1, B, q_1] &\rightarrow b,\\
[q_1, Z_0, q_2] &\rightarrow \lambda.
\end{aligned}$$

Renaming the variables $[q_0, Z_0, q_2]$ as L, $[q_1, X, q_1]$ as M, $[q_1, Z_0, q_2]$ as N, $[q_1, A, q_1]$ as R, $[q_1, B, q_1]$ as P, we write the above minimal α-grammar G' in a user friendly form as

$$\begin{aligned}
S &\rightarrow L,\\
L &\rightarrow MN,\\
M &\rightarrow aMR|a,\\
R &\rightarrow bP,\\
P &\rightarrow b,\\
N &\rightarrow \lambda.
\end{aligned}$$

Thus $G' = (V', \Sigma, P', S)$ is the required minimal α-grammar of order 2 where

$$\begin{aligned}
V' &= \{S, L, M, N, R, P\},\\
\Sigma &= \{a, b\},
\end{aligned}$$

and production rules in P' are as given above. Consider the string $w = a^n b^{2n-2}$ for $n = 1, 2, 3$.

(i) For $n = 1,\ w = a$.

The corresponding derivation of w form the above α-grammar G' is

$$S \Rightarrow L \Rightarrow MN \Rightarrow aN \Rightarrow a.$$

(ii) For $n = 1, w = a^2b^2$. The derivation of w is given by

$$S \Rightarrow L \Rightarrow MN \Rightarrow aMRN \Rightarrow aaRN \Rightarrow aabPN \Rightarrow aabbN \Rightarrow aabb.$$

(iii) For $n = 3, w = a^3b^4$. The derivation of w is given by

$$\begin{aligned}
S &\Rightarrow L\\
&\Rightarrow MN\\
&\Rightarrow aMRN\\
&\Rightarrow aaMRNRN\\
&\Rightarrow aaaRNRN
\end{aligned}$$

$$
\begin{aligned}
&\Rightarrow aaabPNR \\
&\Rightarrow aaabbNRN \\
&\Rightarrow aaabbRN \\
&\Rightarrow aaabbbbPN \\
&\Rightarrow aaabbbbN \\
&\Rightarrow aaabbbb.
\end{aligned}
$$

(i) Again the string $w = a$ is accepted by the QDPDA with successive configurations

$$
\begin{aligned}
(q_0, a, Z_0) &\vdash (q_1, a, XZ_0) \\
&\vdash (q_1, \lambda, Z_0) \\
&\vdash (q_2, \lambda, \lambda).
\end{aligned}
$$

(ii) The string $w = aabb$ is accepted by the QDPDA with successive configurations

$$
\begin{aligned}
(q_0, aabb, Z_0) &\vdash (q_1, aabb, XZ_0) \vdash (q_1, abb, XAZ_0) \\
&\vdash (q_1, bb, AZ_0) \vdash (q_1, b, BZ_0) \vdash (q_1, \lambda, Z_0) \\
&\vdash (q_2, \lambda, \lambda).
\end{aligned}
$$

(iii) The string $w = aaabbbb$ is accepted by the QDPDA with successive configurations

$$
\begin{aligned}
(q_0, a^3b^4, Z_0) &\vdash (q_1, a^3b^4, XZ_0) \vdash (q_1, a^2b^4, XAZ_0) \\
&\vdash (q_1, ab^4, XAAZ_0) \vdash (q_1, b^4, AAZ_0) \\
&\vdash (q_1, b^3, BAZ_0) \vdash (q_1, b^2, AZ_0) \vdash (q_1, b, BZ_0) \\
&\vdash (q_1, \lambda, Z_0) \vdash (q_2, \lambda, \lambda).
\end{aligned}
$$

6. Conclusion

In this paper, we have provided an algorithmic method to construct an α-grammar from the language of a given QDPDA of order n. The initial order of the constructed α-grammar is shown to depend both on the number of states as well as the order "n" of the given QDPDA. But after removing useless variables and uselss productions, we get the minimal equivalent α-grammar of order exactly equal to the order of the language of the given QDPDA.

The constructive algorithmic method is further illustrated with the help of an example.

References

[1] A.V. Aho and J.D. Ullman, The Theory of Parsing, Translation and Computing, Vol. 1, Englewood Cliffs. N.J.: Prentice Hall, 1972.

[2] C. Allauzen, B. Byrne, A.D. Gispert, G. Iglesias and M. Riley, *Pushdown Automata in Statistical Machine Translation*, Association for Computational Linguistics, 40 (2014), doi:10.1162/COLI 00197.

[3] C. Allauzen and R. Michael Riley, Pushdown Transducers, 2011, http://pdt.openfst.org.

[4] J. Berstel, Transductions and Context-Free Languages, Teubner, 1979.

[5] M.A. Harriosn, Introduction to Formal languages Theory, Addison Wesley, Reading, Mass., 1978.

[6] J.E. Hopcroft, R. Motwani and J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education, Addison-Wesley, Reading, Mass., 2004.

[7] M. Hopkins and G. Langmead, *SCFG decoding without binarization*, In Proceedings of EMNLP, Cambridge, MA, 2010, pp.646-655.

[8] R. Hunter, The Design and Construction iof Compilers, Chichester, John Wiley, New York, 1981.

[9] A. Jain. G.C. Petalcorin and K.-S. Lee, *Semi-deterministic pushdown automata (SDPDA) of order n and β-languages*, Journal of Algebra and Applied Mathematics, 14 (2016), 27-40.

[10] A. Jain, K.P. Shum, G.C. Petalcorin, Jr. and K.-S. Lee: *α-grammar and quasi-deterministic pushdown automta (QDPDA) of order "n"*, J. Algeb. and Applied Mathematics, 18 (2020), 99-114.

[11] P. Linz, An Introduction to Frmal lanuages and Automta, Narosa Publishing House, 2009.

[12] M. Mohri, *Weighted automata algorithms*, In M. Drosde, W. Kuick, and H. Vogler, editors, Handbook of Weighted Automata, Springer, Chapter 6, 2009, pp.213-254.

[13] I. Petre and S. Arto, *Algebraic systems and pushdown automata*, In M. Drosde, W. Kuick and H. Vogler, Ed., Handbook of Weighted Automata, Springer, Chapter 7, 2009, pp.257-289.

[14] G.E. Revesz, Introduction to Formal Languages, McGraw-Hill, 1983.

[15] A. Salomaa, *Computations and Automata*, in Encyclopedia of Mathematics and Its Applications, Cambridge University Press, Ca,mbridge, 1985.

[16] W. Kuich and S. Arto, Semirings, automata, languages, Springer Verlag, 1986.

[17] T. Xiao, M. Li, D. Zhang, J. Zhu and M. Zhou, *Better synchronous binarization for machine translation*, In Proceedings of EMNLP, Singapore, 2009, pp.362-370.

[18] H. Zhang, H. Liang, G. Daniel and K. Knight, *Synchronous binarization for machine translation*, In Proceedings of HLT-NAACL, 2006, pp.256-263, New York, NY.

Shanxi Normal University
P.R. China
E-mail: jainarihant@gmx.com

Shanxi Normal University
P.R. China
E-mail: sapnajain@gmx.com

Department of Mathematics and Statistics
College of Science and Mathematics
MSU-Iligan Institute of Technology
Tibanga, Iligan City
Philippines
E-mail: gaudencio.petalcorin@g.msuiit.edu.ph

(*Received: December, 2021; Revised: May, 2022*)